8. Базовая архитектура микропроцессорной системы.

Архитектура ПЭВМ – это абстрактное представление ЭВМ, которое отражает ее структурную, схемотехническую и логическую организацию.

Понятие архитектуры является комплексным и включает в себя:

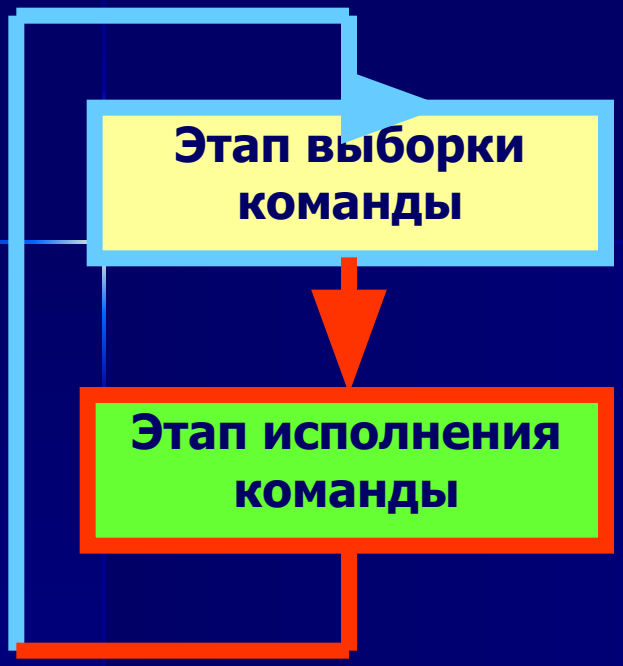
- Структурную схему ПЭВМ,
- Средства и способы доступа к элементам структурной схемы,
- Организацию и разрядность интерфейсов ПЭВМ,
- Набор и доступность регистров,
- Организацию и способы адресации памяти,
- Способы представления и форматы данных,
- Набор машинных команд центрального процессора,
- Форматы машинных команд,
- Порядок обработки нештатных ситуаций (прерываний).

Командный цикл процессора.

Командой называется элементарное действие, которое может выполнить процессор без дальней шей детализации мнемоники машинной команды.

Действия по выбору из памяти и выполнению одной машинной команды процессором называются командным циклом микропроцессора.

Основные этапы выполнения командного цикла микропроцессора.



1. Этап выборки команды:

Цикл 1 – на шину адреса выдается адрес следующей команды, а на шину данных – слово состояния процессора, сопровождаемые соответствующими сигналами SYNC и WR.

Цикл 2 – анализируется наличие сигнала READY, если его нет – переход на ожидание, если есть – чтение команды, сопровождаемое сигналом DBIN.

2 Этап исполнения команды. :

Цикл 1 – код операции команды поступает на дешифратор, инициируется исполнение соответствующей микропрограммы обработки команды.

Цикл 2 – инициируется схема передачи из памяти или из соответствующего регистра первого операнда в буферный регистр БР1.

Цикл 3 – инициируется схема передачи из памяти или из соответствующего регистра второго операнда в буферный регистр БР2.

Цикл 4 – выполняется операция в АЛУ, результат попадает в аккумулятор.

Цикл 5 – результат размещается по адресу нахождения первого операнда.

Система машинных команд микропроцессора.

Разнообразие типов данных, форм их представления и действий, которые необходимы для обработки входной информации и формирования массивов информации для внешних устройств отображения, порождает необходимость использования постоянно расширяемых наборов команд, называемых системой команд микропроцессора.

Система команд должна обладать двумя свойствами – функциональной полнотой и эффективностью.

Функциональная полнота – достаточность системы команд для описания любого алгоритма обработки информации. Известно, что свойством функциональной полноты обладает система, состоящая всего из трех команд (присвоение 0, проверка на 0, присвоение 1), однако построение программ на ее основе неэффективно.

Эффективность системы команд – степень соответствия системы команд назначению ПЭВМ, т.е. классу алгоритмов, для выполнения которых предназначена ПЭВМ. Необходимо отметить, что реализация развитой системы команд требует расширения операционной части процессора, увеличению числа и разрядности дополнительных групп регистров, что приводит к увеличению стоимости и энергоемкости процессоров.

Система команд процессора характеризуется форматами команд, способами адресации и системой операций.

Форматы машинных команд.

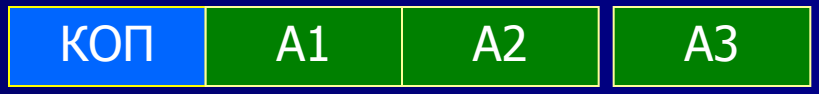
Под форматом команды понимается ее длина, количество, размер, положение, назначение и способ кодирования ее полей.

Состав команды включает в себя следующие виды информации:

- Тип (код) операции, которую следует реализовать в данной команде (КОП),
- Адрес памяти, откуда следует взять первый операнд (А1),
- Адрес памяти, откуда следует взять второй операнд (А2),
- Адрес памяти, куда следует поместить результат (А3).

Для сокращения размера команды в качестве А3 используется обычно последний вычисленный адрес, т.е. А2 или А1.

Формат команды имеет вид:



При использовании регистра-аккумулятора второй адрес не требуется.

Существуют и безадресные команды управления состоянием процессора.

Поле кода операции может содержать указатели на тип формата операнда: байт, слово или двойное слово.

Для уточнения формата данных отдельных команд может использоваться дополнительное поле префикса команды.

Способы адресации операндов в команде.

Способ адресации определяет, каким образом следует использовать информацию, размещенную в поле адреса команды.

Существует пять основных способов адресации операндов:

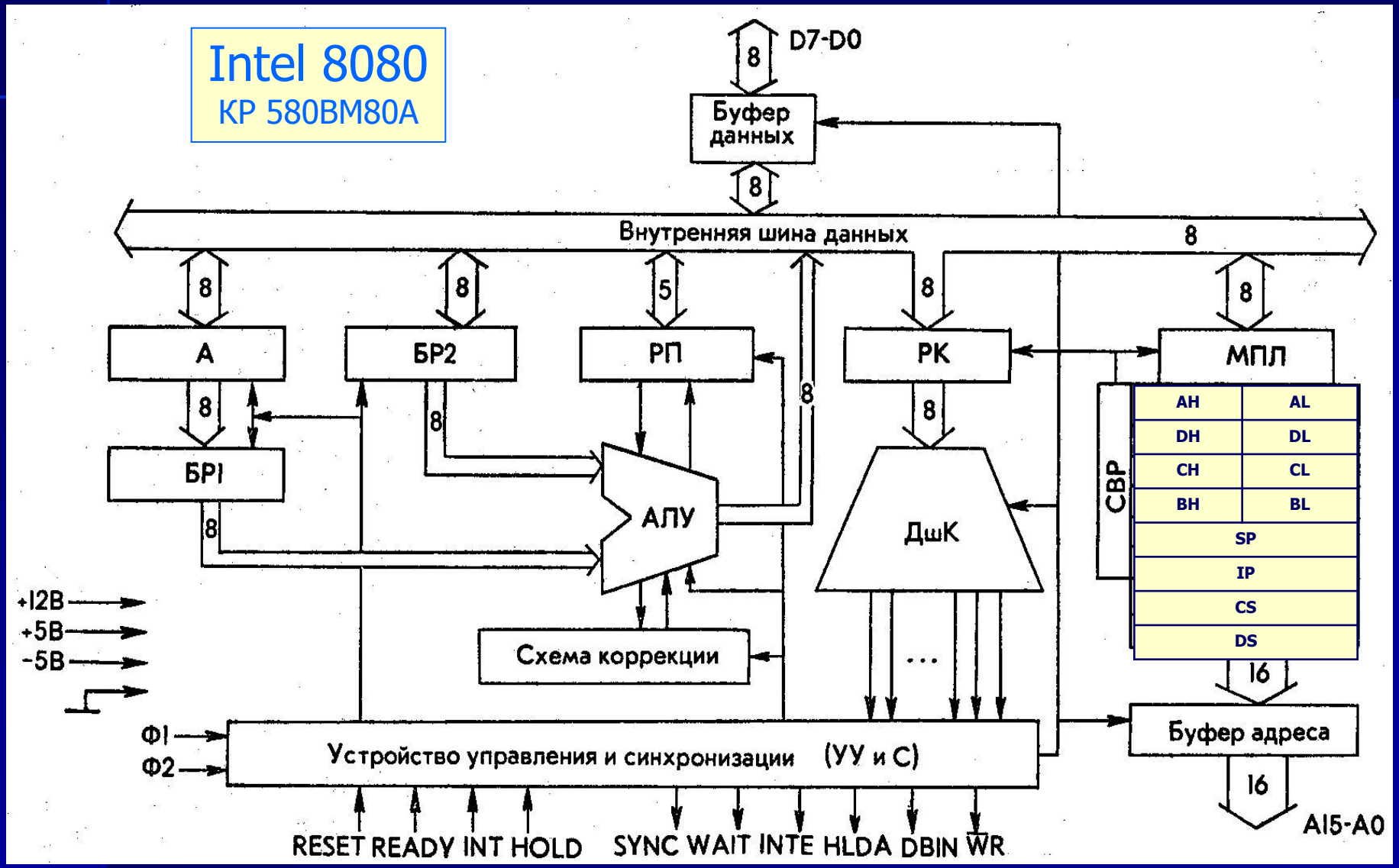
- Непосредственная – в поле адреса команды располагается не адрес, а непосредственно сам операнд (константа),
- Прямая - в поле адреса команды располагается адрес операнда,
- Косвенная - в поле адреса команды располагается не адрес операнда, а адрес ячейки памяти, в которой хранится адрес операнда (адрес адреса операнда),
- Относительная – адрес операнда формируется, как сумма двух слагаемых: базы, хранящемся в специальном регистре или в одном из регистров общего назначения и смещения, извлекаемого из поля адреса команды. Иногда к базе добавляется индекс, увеличивающий или уменьшающий свое значение на единицу после каждого обращения.
- Безадресная – поле адреса в команде отсутствует, а адрес операнда или не имеет смысла, так как сам операнд не нужен)что характерно для системных команд), или адрес его подразумевается по умолчанию (например, аккумулятор).
  Одной из разновидностей безадресного обращения является использование стековой памяти или стека.

Классификация машинных команд по системам операций.

Все операции, выполняемые микропроцессором по кодам операций машинных команд принято подразделять на пять основных классов:

- Арифметико-логические и специальные – команды, по которым процессор выполняет собственно преобразование информации (сложение, вычитание, умножение, деление, логические операции конъюнкции, дизъюнкции, инверсии и т.д.),
- Пересылки и загрузки – передача информации между регистрами процессора и памятью или между уровнями памяти,
- Ввода/вывода – обеспечивают передачу информации между процессором и внешними устройствами через адресное пространство портовых регистров,
- Передачи управления – команды, которые изменяют естественный порядок следования команд программы с изменением содержания счетчика команд. Существует три разновидности команд передачи управления:
  - внутрипрограммные переходы,
  - вызовы подпрограмм,
  - возвраты из подпрограмм.
- Системные – выполняющие управление процессом обработки информации или состоянием внутренних ресурсов процессора.

9. Базовая схема центрального процессора ПЭВМ



Условные обозначения:

А - регистр аккумулятор,
БР1, БР2 – буферные регистры,
РП - регистр признаков (флагов) – слово состояния ЦП,
АЛУ - арифметическое логическое устройство – сумматор,
РК - регистр команд (регистр приема кода операции),
ДшК - дешифратор команд (дешифратор кода операции),
МПЛ - мультиплексор выбора регистров,
СВР - схема выбора и подключения регистров к шине данных,
AH, AL, DH, DL, CH, CL, BH, BL – блок регистров общего назначения (РОН),
SP - 16-разрядный регистр – указатель стека,
IP - программный счетчик (счетчик команд),
CS - 16-разрядный регистр адреса,
DS - 16-разрядный регистр данных.

Внешние сигналы:

RESET - сигнал начальной установки состояния процессора,
READY - сигнал готовности памяти или порта ввода/вывода к обмену данными
INT - сигнал запроса на прерывание (от контроллера прерываний),
HOLD - сигнал запроса на захват шины (от контроллера прямого доступа),
SYNC - сигнал сопровождения выдачи на шину данных слова состояния ЦП,
WAIT - сигнал ожидания готовности READY,
INTE - сигнал разрешения прерывания выполнения программы,
HLDA - сигнал подтверждения захвата шины,
DBIN - сигнал, подтверждающий, что буфер данных включен на чтения,
WR - сигнал, подтверждающий, что буфер данных включен на запись,
Ф1, Ф2 - сигналы генератора синхронизации с различным сдвигом фаз.

Эволюция архитектуры процессоров ПЭВМ семейства x86.

тип	фирма	год	разрядность	частота	память	Расширение операций
i8080	Intel	1976	8 разрядов	4 МГц	640 В	Базовый набор команд
i8086	Intel	1979	16 разрядов	10 МГц	1 МВ	Расширенный базовый набор
i80286	Intel	1982	16 разрядов	16 МГц	16 МВ	Внешний сопроцессор
i80386	Intel	1985	32 разряда	33 МГц	4 ГВ	Внешний сопроцессор
i80486	Intel	1989	32 разряда	40 МГц	4 ГВ	+ Внутренний сопроцессор
Pentium	Intel	1993	32 разряда	100 МГц	4 ГВ	+ Суперскалярность
Pentium II	Intel	1997	32 разряда	400 МГц	4 ГВ	+ MMX
Pentium III	Intel	1999	32 разряда	900 МГц	4 ГВ	+ XMM (SSE)
Pentium IV	Intel	2000	32 разряда	1500 МГц	4 ГВ	+ SSE2
-»- Core 2 Duo	Intel	2007	64 разряда	3000 МГц	4 ГВ	+ SSE3
-”- multi CPU	Intel	2010	128 разрядов	5000 МГц	16 ГВ	+ SSE4

Аналогичные процессора, совместимые по архитектуре с x86, в разное время проектировались и выпускались фирмами IBM, Cyrix, VIA, AMD и другими.

Системная память ПЭВМ

Регистр.

Это ячейка памяти (обычно триггерного типа) емкостью в один байт или одно слово (т.е. 8, 16 или 32 двоичных разряда), используемая для временного хранения данных в ходе выполнения программы.

Схема 8-разрядного регистра – 

Программно-доступные регистры ПЭВМ.

1. 16 (шестнадцать) Общепользовательских регистров, из них:

Восемь 32-битных регистров общего назначения eax (ax, ah, al), ebx (bx, bh, bl), edx (dx, dh, dl), ecx (cx, ch, cl), ebp/bp, esi/si, edi/di, esp/sp.

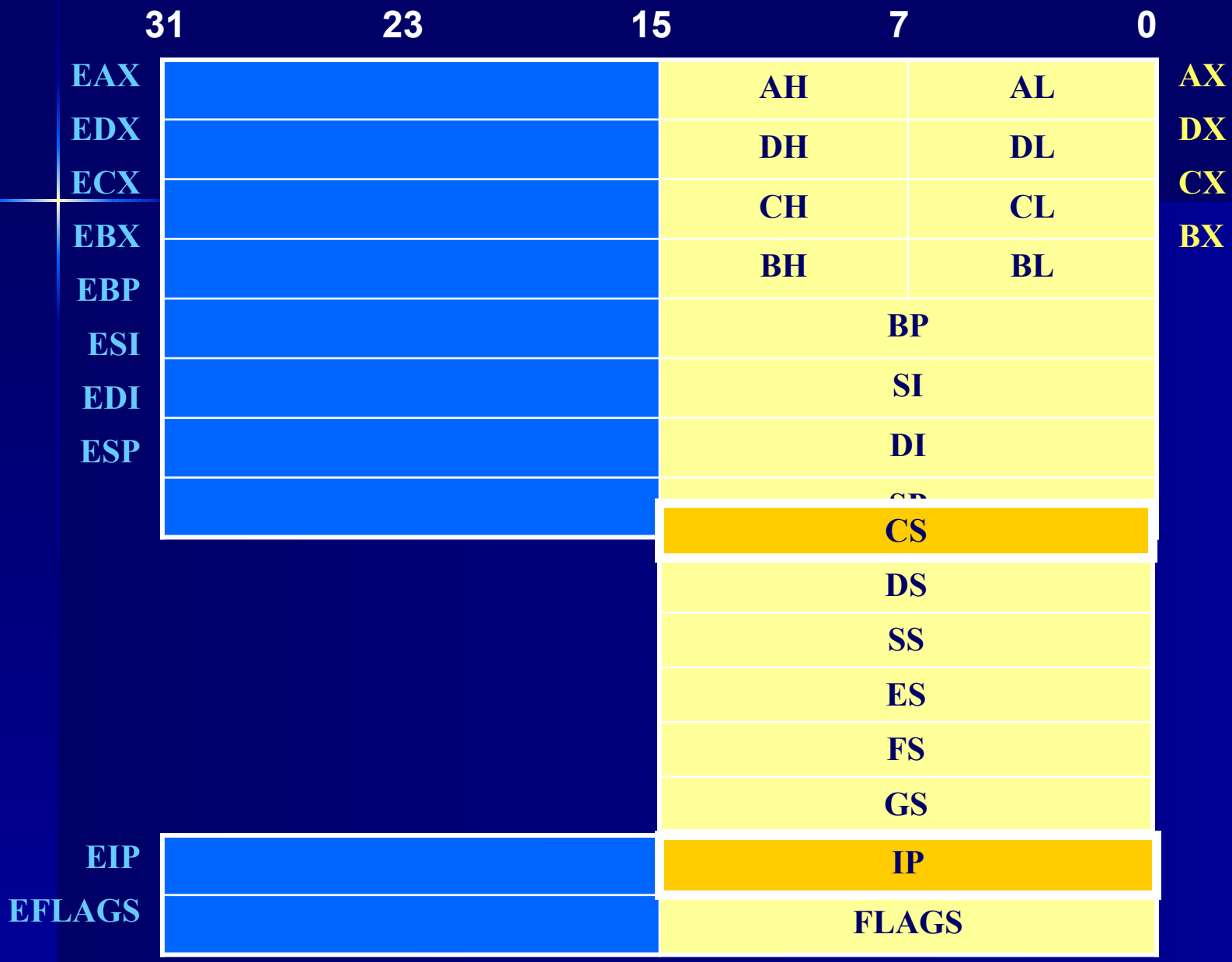
Шесть 16-битных регистров сегментов cs, ds, ss, es, fs, gs.

Два 32-битных регистра состояния и управления – флагов eflags/flags, счетчика (указателя) команд eip/ip.

2. 8 (восемь) 32-битных отладочных регистров - dr0 .. dr7.
3. 8 (восемь) 32-битных регистров тестирования - tr0 .. tr7.
4. 5 (пять) 32-битных управляющих регистра cr0, cr1, cr2, cr3, cr4.
5. 4 (четыре) 48-битных системных адресных регистра gdtr, idtr, tr, ldtr.
6. 24 регистра данных математического сопроцессора, имеющих стековую структуру хранения обрабатываемых вещественных чисел, из них:

- 8 (восемь) 80-битных регистра для размещения вещественных чисел,
- 8 (восемь) 80-битных регистра данных MMX (блок MMX) mmx0 .. mmx7,
- 8 (восемь)128-битных регистра данных XMM (блок XMM) xmm0 .. xmm7 для обработки 3D-графических изображений.

Программно-регистровая модель ПЭВМ.



Регистры общего назначения (РОН).

eax (ax, ah, al) - регистр – аккумулятор для хранения результата операции,
ebx (bx, bh, bl) - базовый регистр для хранения базовой части адреса,
edx (dx, dh, dl) – регистр хранения промежуточного результата (данных),
ecx (cx, ch, cl) - регистр – счетчик цикла (например, в команде LOOP),
ebp/bp – регистр–указатель базовой части адреса стека (база стека),
esi/si - регистр индекса источника (величины постоянного смещения адреса),
edi/di- регистр индекса приемника (величины постоянного смещения адреса),
esp/sp – регистр–указатель адреса последней записи в стеке (вершина стека),

Примечание: все регистры общего назначения кроме фиксированного назначения могут использоваться для адресных вычислений и для хранения результатов большинства арифметических и логических операций.

Сегментные регистры.

cs – регистр параграфа адреса исполняемой команды (сегмент кода),
ds – регистр параграфа адреса расположения области стека (сегмент стека),
ss – регистр параграфа адреса области данных (сегмент данных),
es, fs, gs – дополнительные сегментные регистры, замещаемые по префиксу

Регистры состояния и управления.

eflags/flags – регистр системных флагов результата исполнения команды,
eip/ip – регистр адреса исполняемой команды (эффективная часть адреса).

Содержание регистра флагов ПЭВМ.

Флаги состояния :

Флаг вложенности задачи
уровень привилегированности ввода-вывода
флаг переполнения
флаг переноса
флаг четности
вспомогательный флаг переноса
флаг нуля
флаг знака

EFLAGS **FLAGS**

31	...	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	...	AC	VM	RF	0	NT	IOPL		OF	DF	IF	TF	SF	ZF	0	AF	0	PF	1	CF

Флаг управления :

Флаг направления

Системные флаги :

Флаг трассировки
флаг прерывания
флаг возобновления
флаг виртуальной 8086
флаг контроля выравнивания

10. Схема построения машинных кодов команд центрального процессора.

длина команды в байтах (L_{max} = 15 байт)

0 – 3	1 - 2	0 - 1	0 – 1	0, 1, 2, 4	0, 1, 2, 4
префикс	КОП	mod r/m	SIB	смещение	операнд

	К	О	П				**w**
	К	О	П			**d**	**w**
	К	О	П			**s**	**w**
	К	О	П		**reg**		
	К	О	П	**w**	**reg**		

mod	**reg/коп**	**r/m**

режим адресации

ss	**index**	**base**

режим масштабирования адреса

mod – 00 – абсолютный адрес операнда (байта смещения нет)
01 – имеется один байт смещения адреса операнда
10 – имеется два байта смещения адреса операнда
11 – операнд в регистре

reg – 000 – AL или AX при w=1
001 – CL или CX при w=1
010 – DL или DX при w=1
011 – BL или BX при w=1
100 – AH или SP при w=1
101 – CH или BP при w=1
110 – DH или SI при w=1
111 – BH или DI при w=1

ss – 00 - множитель 1
01 - множитель 2
10 - множитель 4
11 - множитель 8

префиксы

Цепочечных повторений	**3Fh.**
Размер адреса (32/16)	**67h**
Размер операнда (32/16)	**66h**
Замена сегментного рег.	**2Eh…**
Блокировка шины	**0Fh**

Максимум 3 префикса

Таблица кодов операций (КОП) базовых машинных команд (таблица декомпиляции)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	ADD						PUSH	POP	OR						PUSH	переход 2-й байт
	Eb.Gb	Ev.Gv	Gb.Eb	Gv.Ev	AL.Ib	eAX.Iv	ES	ES	Eb.Gb	Ev.Gv	Gb.Eb	Gv.Ev	AL.Ib	eAX.Iv	CS	
1	ADC						PUSH	POP	SBB						PUSH	POP
	Eb.Gb	Ev.Gv	Gb.Eb	Gv.Ev	AL.Ib	eAX.Iv	SS	SS	Eb.Gb	Ev.Gv	Gb.Eb	Gv.Ev	AL.Ib	eAX.Iv	DS	SS
2	AND						SEG	DAA	SUB						SEG	DAS
	Eb.Gb	Ev.Gv	Gb.Eb	Gv.Ev	AL.Ib	eAX.Iv	=ES		Eb.Gb	Ev.Gv	Gb.Eb	Gv.Ev	AL.Ib	eAX.Iv	=CS	
3	XOR						SEG	AAA	CMP						SEG	AAC
	Eb.Gb	Ev.Gv	Gb.Eb	Gv.Ev	AL.Ib	eAX.Iv	=SS		Eb.Gb	Ev.Gv	Gb.Eb	Gv.Ev	AL.Ib	eAX.Iv	=DS	
4	INC регистр общего назначения								DEC регистр общего назначения							
	eAX	eCX	eDX	eBX	eSP	eBP	eSI	eDI	eAX	eCX	eDX	eBX	eSP	eBP	eSI	eDI
5	PUSH регистр общего назначения								POP регистр общего назначения							
	eAX	eCX	eDX	eBX	eSP	eBP	eSI	eDI	eAX	eCX	eDX	eBX	eSP	eBP	eSI	eDI
6	PUSHA	POPA	BOUND	ARPL	SEG	SEG	Размер операнда	Рармер адреса	PUSH	IMUL	PUSH	IMUL	INSB	INSW/D	OUTSB	OUTSW/D
		eCX	Gv,Ma	Ew,Rw	=FS	=GS			Iv	GvEvIv	Ib	GvEvIb	Yb,DX	Yv,DX	DX,Xb	DX,Xv
7	Близкий переход по условию (JB)															
	JO	JNO	JB	JNB	JZ	JNZ	JBE	JNBE	JS	JNS	JP	JNP	JL	JNL	JLE	JNLE
8	Непосред. Grpl		MOVB	Grpl	TEST		XCHG		MOV					LEA	MOV	POP
	Eb,Ib	Eb,Ib	AL,imm8	Ev,Ib	Eb,Gb	Ev,Gv	Eb,Gb	Ev,Gv	Eb.Gb	Ev.Gv	Gb.Eb	Gv.Ev	Ew,Sw	Gv,M	Sw,Ew	Ev
9	NOP	XCHG с регистром слова или двойного слова eAX							CBW	CWD	CALL	WAIT	PUSHHF	POPF	SAHF	LAHF
		eCX	eDX	eBX	eSP	eBP	eSI	eDI			Ap		Fv	Fv		
A	MOV				MOVSB	MOVSW/D	CMPSB	CMPSW/D	TEST		STOSB	STOSW/D	LODSB	LODSW/D	SCASB	SCASW/D
	AL,Ob	eAX,Ov	Ob,AL	Ov,eAX	Xb,Yb	Xv,Yv	Xb,Yb	Xv,Yv	AL,Ib	eAX,Iv	Yb,AL	Yv,eAX	AL,Xb	eAX,Xv	AL,Xb	eAX,Xv
B	MOV с непосредственным байтом в байтовом регистре								MOV непосредственное слово или двойное слово в регистр							
	AL	CL	DL	BL	AH	CH	DH	BH	eAX	eCX	eDX	eBX	eSP	eBP	eSI	eDI
C	Сдвиг Grp2		ближний RET		LES	LDS	MOV		ENTER	LEAVE	Дальний RET		INT	INT	INTO	IRET
	Eb,Ib	Ev,Ib	Iw		Gv,Mp	Gv,Mp	Eb,Ib	Ev,Ib	Iw,Ib		Iw		3	Ib		
D	Сдвиг Grp2				AAM	AAD		XLAT	ESC (Выход на байт множества команд сопроцессора)							
E	LOOPNE	LOOPE	LOOP	JCXZ	IN		OUT		CALL	JMP			IN		OUT	
	Jb	Jb	Jb	Jb	Al,Ib	eAX,Ib	Ib,AL	Ib,eAX	Jv	Jv	AP	Jb	AL,DX	eAX,DX	DX,AL	Dx,eAX
F	LOCK		REPNE	REP REPE	HLT	CMS	Унарный Grp3		CLC	STC	CLI	STI	CLD	STD	INC/DEC Grp4	INC/DEC Grp5

Кодирование метода адресации (условные обозначения в таблице).

Условные обозначения в таблице:

A - Прямая адресация: команда не имеет байта modR/M; адрес операнда содержится непосредственно в команде; регистр базы, регистр индекса, коэффициент масштабирования не используется.

C - Поле reg байта modR/M указывает управляющий регистр.

D - Поле reg байта modR/M указывает регистр отладки.

E - За кодом операции следует байт modR/M, описывающий операнд. Операндом может быть регистр общего назначения или адрес памяти. Если это адрес памяти, он вычисляется исходя из сегментного регистра и следующих величин: регистр базы, индексный регистр, масштабирование, смещение.

F - Регистр флагов.

G - Поле reg байта modR/M указывает регистр общего назначения.

I - Непосредственные данные (константа). Значение операнда кодируется последовательностью байтов непосредственно в команде.

J - Команда содержит относительное смещение, прибавляемое к регистру счетчика команд (EIP).

M - Байт modR/M может ссылаться только на память.

O - Команда не содержит байта modR/M; относительный адрес операнда кодируется как слово или двойное слово (в зависимости от атрибута размера адреса) непосредственно в команде. Регистр базы, регистр индекса, коэффициент масштабирования не используется.

R - Поле mod байта modR/M может указывать только на регистр общего назначения.

S - Поле reg байта modR/M указывает сегментный регистр.

T - Поле reg байта modR/M указывает регистр.

X - Адресация памяти при помощи пары регистров DS:SI; например, MOVS, COMPS, OUTS.

Y - Адресация памяти при помощи пары регистров DS:DI; например, MOVS, COMPS, INS,.

Кодирование типов операндов (условные обозначения в таблице).

Условные обозначения в таблице:

a - Два операнда длиной в слово в памяти или два операнда длиной в двойное слово в памяти, в зависимости от атрибута размера операнда (используется для команды BOUND).

b - Байт (независимо от атрибута размера операнда).

c - Байт или слово, в зависимости от атрибута размера операнда.

d - Двойное слово (независимо от атрибута размера операнда).

p - 32- или 48-разрядный указатель, в зависимости от атрибута размера операнда.

s - 6-разрядный псевдо-дескриптор.

v - Слово или двойное слово, в зависимости от атрибута размера операнда.

w - Слово (независимо от атрибута размера операнда).

Коды регистров в таблице.

Когда в качестве операнда используется закодированный в команде регистр, он определяется по имени, например **AX**, **CL** или **ESI**.

Имя регистра определяет его размер - 32, 16 или 8 бит. В случае, если размер регистра определяется атрибутом размера операнда, используется запись такого формата **eXX**.

Например, запись **eAX** означает, что при атрибуте размера операнда равном 16, используется регистр **AX**, а при атрибуте размера операнда равном 32 - регистр **EAX**.

Для нахождения строки, содержащей нужную операцию необходимо использовать значение старших четырех битов кода операции, а для нахождения столбца - значение младших четырех битов. Если код операции равен 0FH, необходимо обратиться к таблице двухбайтовых кодов операций и использовать значение второго байта кода операции для нахождения соответствующей строки и столбца.

Дополнительные таблицы кодов операций машинных команд.

Существует ряд дополнительных таблиц расширения кодов операций машинных команд, доступных через первый байт КОП:

1. Команды математического сопроцессора.

Эти наборы команд доступны через инструкцию 11011xxx (или D8x) первого байта (см. основную таблицу кодов), которая называется ESC-командой.

Наборы команд математического сопроцессора содержат 8 дополнительных таблиц команд, т.е. позволяют реализовать до 1280 дополнительных машинных команд, для различных модификаций математического сопроцессора архитектуры x86.

В связи с тем, что в настоящее время реализовано только 63 машинные команды для 84-х команд ассемблера, этот набор команд обладает большой избыточностью

2. Команды дополнительных модулей SSE-расширений архитектуры x86.

Эти наборы команд сведены в отдельную таблицу команд, доступную через первый байт КОП 00001111 (или 0f), либо располагаются в таблицах команд математического сопроцессора и доступны через инструкцию 11011xxx (или D8x) первого байта (см. основную таблицу кодов).

3. Команды дополнительных модулей 3DNow!-расширений для процессоров AMD Athlon, совмещенных с архитектурой x86.

Данные наборы имеют в своем составе 3 байта КОП. Причем первые два имеют конструкцию 00001111 00001111 (или 0f 0f), т.е. расположены в собственной таблице, спроектированной фирмой AMD для своих процессоров дополнительно к x86.

4. В настоящее время разработчиками архитектур процессоров ведется работа по расширению наборов команд Ассемблера и соответствующих машинных команд для мультипроцессорных систем.

Примеры формирования кода машинной команды

Команда **MOV** пересылки данных:

Из регистра 1 в регистр 2	1000100w	11 reg1 reg2	88
Из регистра 2 в регистр 1	1000101w	11 reg1 reg2	8B
Регистр – память	1000100w	mod reg r/m	88
Константа – регистр C6	1100011w	11 000 reg	непосредственные данные
Константа – память C6	1100011w	mod 000 r/m	непосредственные данные
Сумматор – память A3	1010001w	полное смещение	

Команда **SAHF** загрузка регистра флагов flags в регистр AH

9E 10011110

Команда **CBW/CWDE** преобразование байта в слово/слова в двойное слово

CBW	98	
CWDE	99	10011000
		10011001

11. Системы команд ассемблера для архитектуры ЦПУ x86.

Классификация базовых (целочисленных) команд ассемблера.

Математический сопроцессор.

Схема центрального процессора содержит обычный сумматор, предназначенный для обработки простых чисел со знаком.

При выполнении операций с вещественными числами на процессор поступают соответствующие команды (начинающиеся с символа «F») с кодом операции, начинающимся с инструкции 11011.

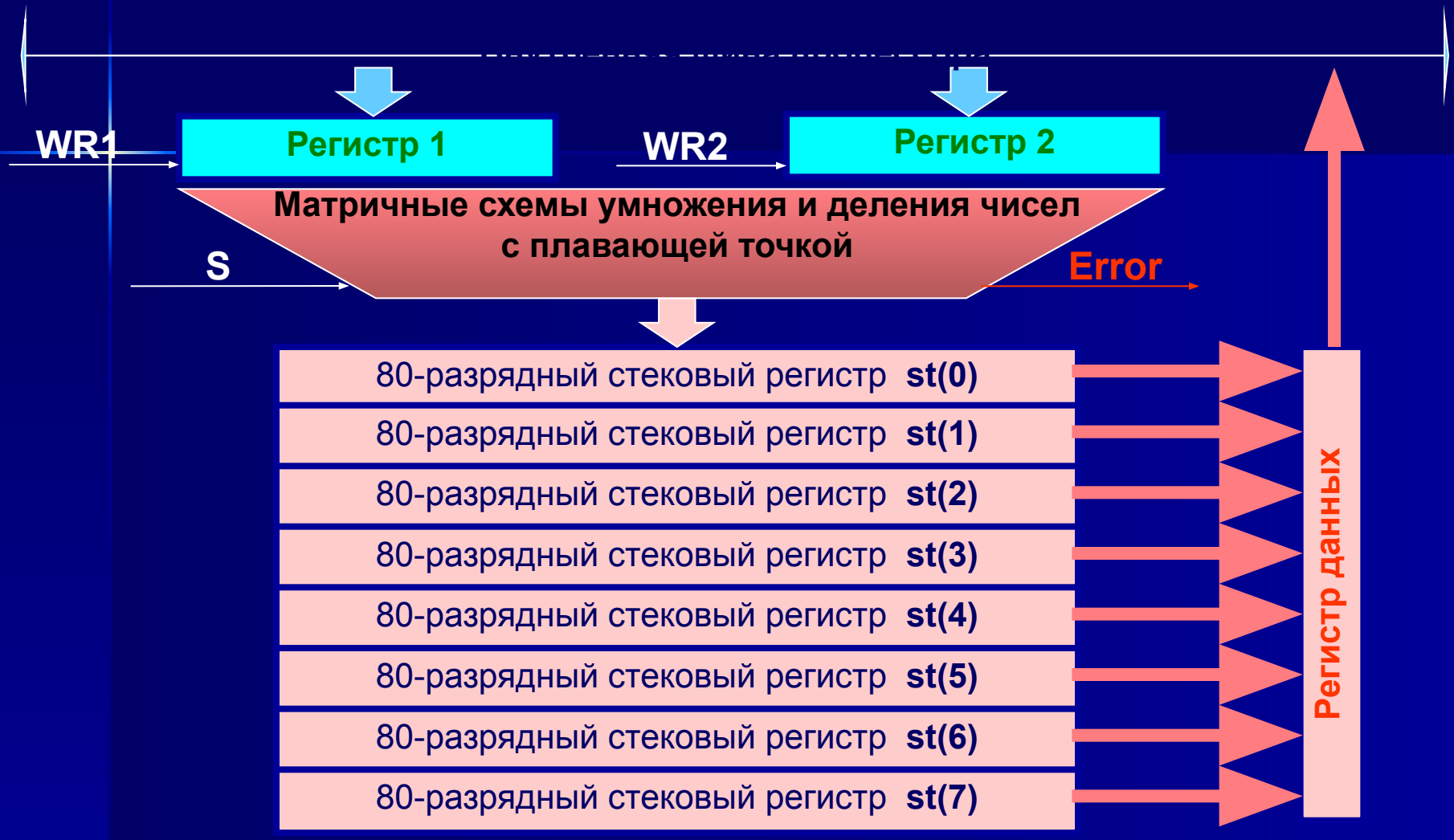
Все обрабатываемые операнды передаются в буферные регистры математического сопроцессора, представляющего собой цифровой автомат с жесткой схемной логикой. Здесь над ними производятся все необходимые для вычислений математические операции. Результат возвращается в аккумулятор центрального процессора.

Если математический сопроцессор отсутствует, то исполнение математических операций над вещественными числами осуществляется соответствующими программами BIOS, эмулирующими работу математического сопроцессора.

Однако, при этом, вещественное число преобразуется в совокупность простых чисел (мантиссу и степень), с которыми работает АЛУ центрального процессора, затем производится обратное преобразование.

Программное эмулирование процесса обработки вещественных чисел математическим сопроцессором приводит к существенному замедлению (на 2 порядка) процесса вычислений.

Принципиальная схема математического сопроцессора.



Классификация команд ассемблера для математического сопроцессора.

Классификация команд ассемблера для MMX-расширения процессора.

Классификация команд ассемблера для XMM-расширения SSE-2.

Классификация команд ассемблера для 3DNow!-расширения для процессоров AMD Athlon.

Основные инструменты для работы с низкоуровневым программным кодом.

1. OllyDbg (www.ollydbg.de) – считается лучшим 32-х битным отладчиком приложений. Имеет встроенный анализатор машинного кода, который распознает и визуально обозначает процедуры, циклы, константы и строки, а также внедренные в код обращения к функциям API NT и их параметры.

2. IDA Pro (www.hex-rays.com/idapro) – интерактивный дизассемблер и отладчик приложений Работает через функции MS Debugging API в NT или библиотеку ptrace в UNIX.

3. Syser Kernel Debugger (www.sysersoft.com) – отладчик ядра ОС и процессов (не работает в Windows 7 и MS Server 2008).

4. SoftICE (в настоящее время не поддерживается) – отладчик для Windows на уровне ядра.

5. Fasm (www.flatassembler.net) - свободно распространяемый многопроходной кросс-ассемблер. Обладает небольшими размерами и очень высокой скоростью компиляции. Поддерживает как объектные форматы, так и некоторые форматы исполняемых файлов.

Помимо базового набора инструкций процессора и сопроцессора Fasm v.1.68 поддерживает наборы инструкций MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4a и 3DNow!, а также EM64T и AMD64 (включая AMD SVM и Intel SMX).

6. WinHex (www.x-ways.net/winhex) - Один из лучших шестнадцатеричных редакторов машинного кода. Позволяет редактировать файлы размером более 4 Гб с жестких дисков, внешних накопителей, флешек. Имеет собственный механизм работы с файловыми системами FAT, NTFS, Ext2/3, ReiserFS, Reiser4, UFS, CDFS, UDF, а также RAID-массивами и динамическими дисками (независимо от операционной системы). В программу также встроены средства для восстановления данных, безопасного удаления файлов, инструмент для редактирования содержимого оперативной памяти, алгоритмы шифрования и т.д.

Пример использования декомпилятора для создания антиотладочной защиты файла