Програмен модел. Режими на работа на IA-32 микропроцесори

Режими на работа

- IA-32 архитектурата поддържа 3 базови режими на работа:

1. Защитен режим /Protected mode/ — Този режим е естественото състояние на процесора. Сред възможностите на защитения режим е възможността за директно изпълнение на софтуер, написан за процесора 8086, който се изпълнява в реален режим на работа /real-address mode/, обаче в защитено, многозадачно обкръжение. Това свойство на процесора се нарича виртуален 8086 режим /**virtual-8086 mode/**, обаче това всъщност не е друг режим на процесора. Виртуалният 8086 режим е атрибут на защитения режим, който може да бъде разрешен за всяка задача.

2. Реален режим /Real-address mode/ — Този режим допълва програмната среда на Intel 8086 процесора с разширения (като възможността да превключва в защитен режим или режим на системно управление). Процесорът се включва в реален режим при включване на захранваането или при НУ/reset/.

Режими на работа

3. Режим на системно управление /System management mode (SMM)/

Този режим обезпечава ОС с прозрачен механизъм за допълнителни платформено-специфични функции като управление на захранването и системна безопасност. Процесорът влиза в SMM режим когато се активира крачето на процесора за външно прекъсване за системно управление /SMM interrupt (SMI#)/ или бъде прието такова SMI от усъвършенствувания контролер за прекъсване /advanced programmable interrupt controller (APIC)/.

Базова операционна среда

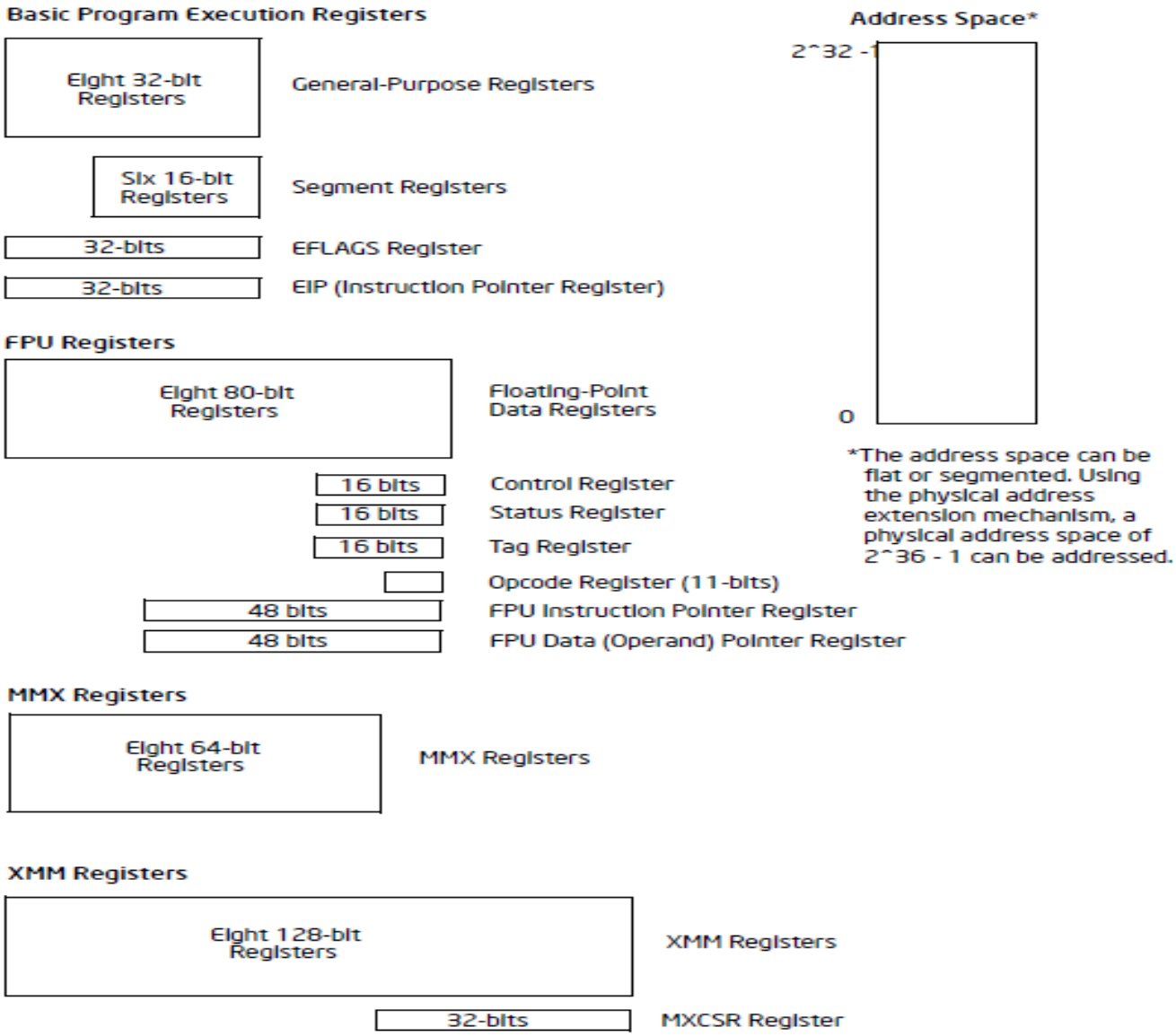
Всяка програма или задача, изпълнявана на IA-32 процесор притежава набор от ресурси за изпълнение на инструкциите и записания код, данни, информация за състоянието. Те са съставени от:

- **Адресно пространство** — Всяка задача или програма, изпълнявана на IA-32 процесор може да адресира линейно адресно пространство до 4 GB (2^{32} B) и физическо адресно пространство до 64GB (2^{36} B).
- **Базови регистри за изпълнение на програмата** — 8 регистри с общо предназначение, 6 сегментни регистри регистър EFLAGS и EIP (instruction pointer) регистър са включени в базовата операционна среда, в която се изпълняват инструкциите с общо предназначение. Тези инструкции извършват базовата целочислена аритметика върху цели числа с размер байт, дума и двойна дума, като управляват пътя на програмата, оперират с битове, стрингове от байтове и адресират паметта.

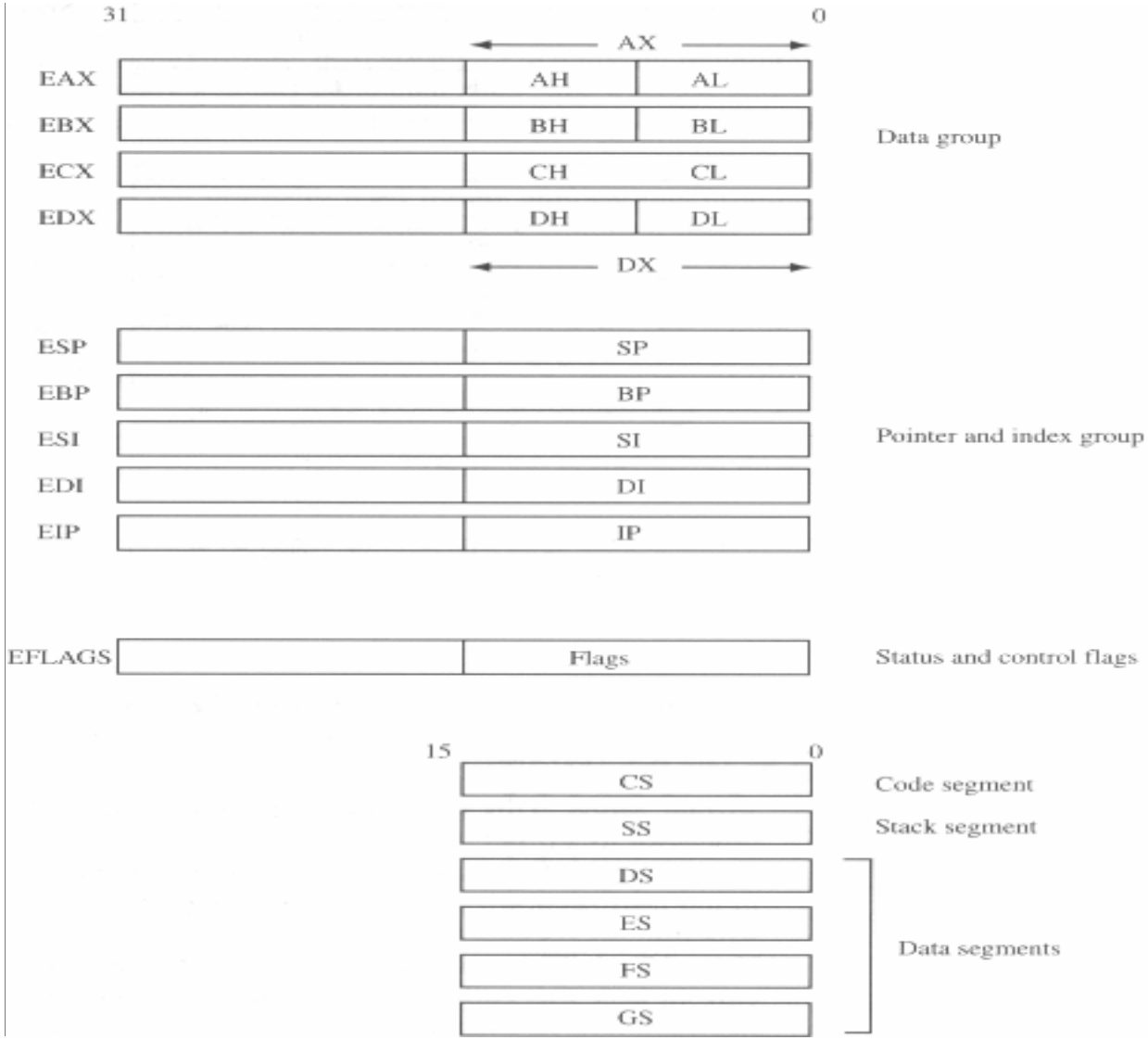
Базова операционна среда

- **Регистри на x87 FPU /устройство за работа с числа с плаваща запетая/** — 8 бр. 80b регистри за данни, 3 бр. 16b регистри – регистър за управляващата дума, регистър на състоянието, маркерен регистър /tag register/. Освен това x87 FPU има програмен брояч, указател на операнда(данните) регистър на операционния код /opcode register/. Тези регистри създават операционната среда за действия с единична точност, двойна точност и с двойна удължена точност , числа с плаваща запетая, цели числа с дължина четворни думи и двоично кодирани десетични (BCD) числа.
- **MMX регистри** — 8 бр. MMX регистри поддържат изпълнение на единична инструкция, multiple-data (SIMD) операции над 64-b пакетирани байтове, думи, и двойни думи, които представят цели числа.
- **XMM регистри** — 8 бр. XMM регистри за данни и MXCSR регистър поддържат изпълнението на SIMD операции върху 128-b пакетирани числа с плаваща точка с единична точност и с с двойна точност, като и върху цели числа представени като 128-b пакетирани байтове, думи, двойни думи и четворни думи.
- **Стек** — За поддръжка на извиквания на процедури и подпрограми и за прехвърляне на параметри между процедурите и подпрограмите в операционната среда се включва стека и управлението на ресурсите му. Стекът е разположен в паметта.

Програмен модел IA-32



Базов програмен модел



Базов програмен модел

IA-32 архитектурата осигурява 16 базови програмно изпълнителни регистъра за ползване в обща система или при приложно програмиране. Тези регистри могат да бъдат разделени на:

- **Регистри с общо предназначение**. Тези 8 регистъра са достъпни за запис на операнди и указатели.
- **Сегментни регистри**. Тези регистри съхраняват до 6 сегментни селектора.
- **EFLAGS (регистър за състояние и управление на програмата)**. EFLAGS регистърът отчита състоянието на изпълняваната програма и позволява ограничен (на приложно ниво) контрол на процесора.
- **Програмен брояч - EIP регистър**. EIP съдържа 32-b указател към следващата инструкция, която ще се изпълни.

Регистри с общо предназначение

General-Purpose Registers

31 16	15 8	7 0	16-bit	32-bit
	AH	AL	AX	EAX
	BH	BL	BX	EBX
	CH	CL	CX	ECX
	DH	DL	DX	EDX
	BP			EBP
	SI			ESI
	DI			EDI
	SP			ESP

Регистри с общо предназначение

32-b регистри с общо предназначение EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP служат за съхранение на следните обекти:

- Операнди за логически и аритметични операции
- Операнди за адресни пресмятания
- Указатели на паметта

Регистърът ESP съхранява стековия указател и не бива да не се ползва за други цели.

Много инструкции са предназначени за съхранение на операнди в специфични регистри. Например стринговите инструкции използват съдържанието на ECX, ESI и EDI регистри като операнди.

Регистри с общо предназначение

Тези регистри се ползват и за специални цели:

- **EAX** — Акумулатор за операнди и даннови резултати
- **EBX** — Указател за данни в DS сегмента
- **ECX** — Брояч за стрингови операции и за цикли
- **EDX** — I/O указател
- **ESI** — Указател за данни в сегмент, сочен от DS регистъра; указател за стринга-източник при стрингови операции
- **EDI** — Указател за данни /получател/ в сегмент, сочен от ES регистър; Указател на получателя при стрингови операции
- **ESP** — Стеков указател (в SS сегмента)
- **EBP** — Указател за данни в стека (в SS сегмент)

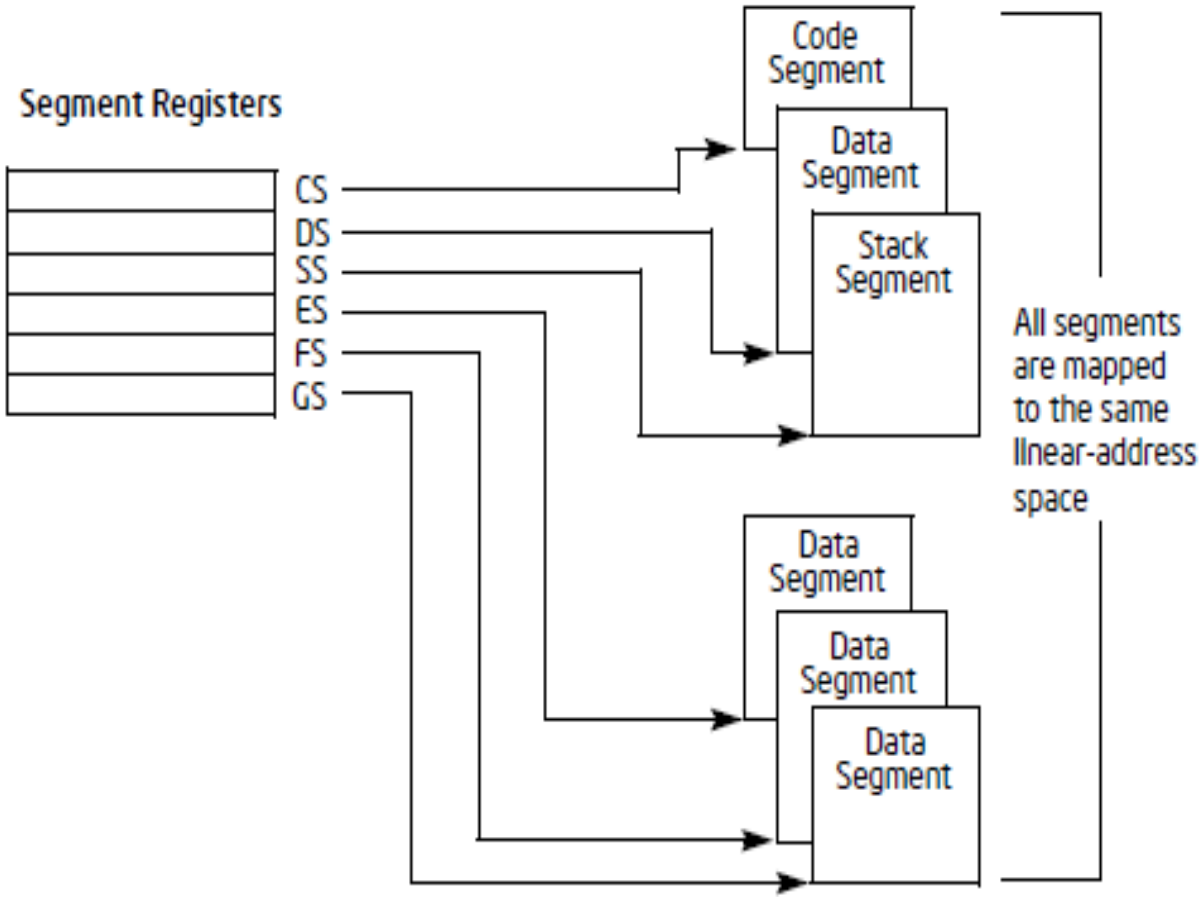
Регистри с общо предназначение

Register Type	Without REX
Byte Registers	AL, BL, CL, DL, AH, BH, CH, DH
Word Registers	AX, BX, CX, DX, DI, SI, BP, SP
Doubleword Registers	EAX, EBX, ECX, EDX, EDI, ESI, EBP, ESP
Quadword Registers	N.A.

Сегментни регистри

- Сегментните регистри (CS, DS, SS, ES, FS, GS) съхраняват 16-b сегментни селектори.
- Сегментният селектор е специален указател, който идентифицира сегмента в паметта. За достъп до отделен сегмент в паметта, сегментният селектор трябва да бъде разположен в подходящ сегментен регистър.
- Когато се ползва сегментен модел на паметта, всеки сегментен регистър е обикновено зареден с различен сегментен селектор така, че всеки сегментен регистър сочи към различен сегмент в линейното адресно пространство.

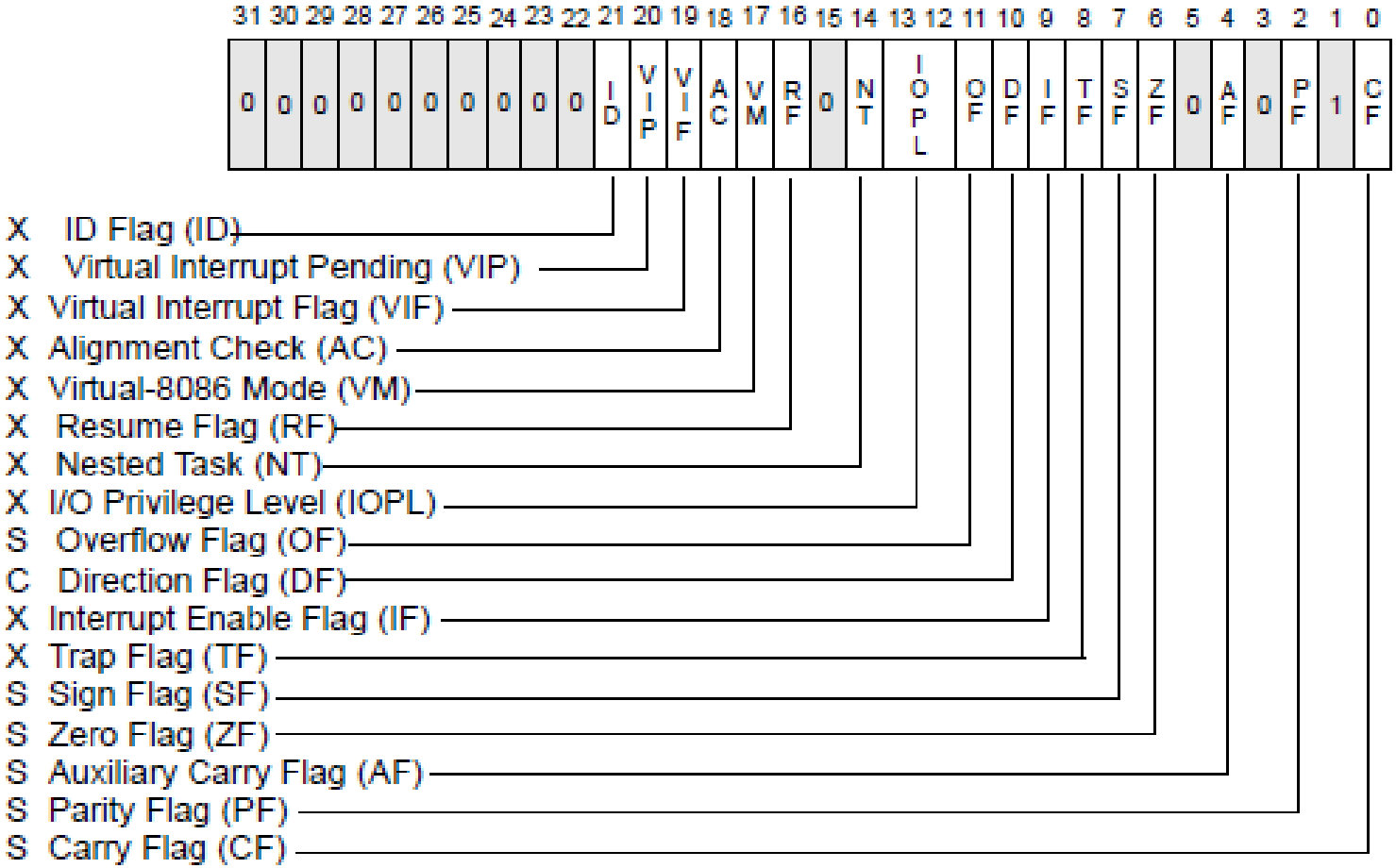
Сегментни регистри



Сегментни регистри

- Всеки от сегментните регистри се асоцииира с един от 3 типа на информация в паметта: код, данни, стек. Например CS регистърът съдържа сегментен селектор за **кодовия сегмент, където са записани или се изпълняват инструкциите.** Процесорът извлича инструкциите от кодовия сегмент, използвайки логически адрес, който се състои от сегментния селектор, записан в CS регистъра и съдържанието на EIP регистъра. EIP регистърът съдържа отместването в кодовия сегмент на следващата инструкция, която ще се изпълни.
- DS, ES, FS, GS регистри сочат към 4 **сегмента за данни.** Достъпа до 4 даннови сегмета позволява ефективен и сигурен достъп до различни типове даннови структури.
- SS регистърът съдържа сегментния селектор за **стековия сегмент**, където е записан процедурния стек за програма, задача или манипулатор, които се изпълняват в момента. Всички стекови операции ползват SS регистъра за намиране на стековия сегмент.

EFLAGS регистър



S Indicates a Status Flag
C Indicates a Control Flag
X Indicates a System Flag

EFLAGS регистър

32-b EFLAGS регистър съдържа:

- Група флагове за състоянието,
- Управляващ флаг
- Група системни флагове.

При инициализация на процесора (при активиране на краче RESET или INIT), състоянието на EFLAGS регистъра става 00000002H.

Флагове на състоянието

- Флаговете на състоянието (битове 0, 2, 4, 6, 7 и 11) от EFLAGS регистъра индицират резултата от аритметичната инструкция, като ADD, SUB, MUL, DIV. Функциите на флаговете за състояние са:
- **CF (b0) Carry flag** — Установява се ако аритметичната операция генерира пренос/заем от най-старшия бит на резултата, иначе се нулира.
- **PF (b2) Parity flag** — Установява се ако най-младшият байт на резултата съдържа четно число или четен брой единици, иначе се нулира.
- **AF (b4) Adjust flag** — Установява се ако аритметичната операция генерира пренос или заем от b3 на резултата, иначе се нулира. Този флаг се ползва при двоично кодирани десетични числа (BCD аритметика).
- **ZF (b6) Zero flag** — Установява се ако резултатът е 0; нулира се в противен случай.
- **SF (b7) Sign flag** — В него се копира най-старшия бит на резултата, който е знака на целите числа със знак. (0 – показва неотрицателно, а 1 – отрицателно число.)
- **OF (b11) Overflow flag** — Установява се ако целия резултат е твърде голямо положително число или твърде малко отрицателно число (при невалиден знаков бит) за да бъде верен резултат, иначе се нулира. Този флаг индицира състояние препълване при аритметика на числа със знак, представени в допълнителен код.

DF флаг

- DF флаг. direction flag (DF, разположен в b 10 на EFLAGS регистъра) управлява стринговите операции (MOVS, CMPS, SCAS, LODS, STOS). Установяването на DF флага предизвиква при стринговите инструкции авто-декремент (обработка на стринга от големите към малките адреси). Нулирането на DF флага flag предизвиква при стринговите инструкции авто-инкремент (обработка на стринга от малките към големите адреси).

Системни флагове

Системните флагове и IOPL полето в EFLAGS регистъра управлява операциите на ОС и изпълнимите операции.

Те не трябва да се модифицират от потребителските програми. Функциите на системните флагове са следните:

TF (b8) Trap flag — Установява се за разрешаване на режим на постъпково изпълнение за настройка; нулира се за забрана на постъпков режим.

IF (b9) Interrupt enable flag — Управлява отговора на процесора на маскируемите заявки за прекъсване. Установява се отговор на маскируемите прекъсвания, нулира се за забрана на маскируемите прекъсвания.

IOPL (b12 и b13) I/O privilege level поле — Индицира текущото ниво на I/O привикегия на изпълнявана програма (CPL) . Текущото ниво ниво на привилегия трябва да бъде по-малко или равно на нивото на I/O привилегия за достъп до I/O адресно пространство. Това поле може да се модифицира само чрез POPF и IRET инструкции, при опериране на CPL=0.

Системни флагове

NT (b14) Nested task flag — Управлява верижното изпълнение на прекъсваната и извикваната задачи. Установява се когато текущата задача е свързана с предишната изпълнявана задача. Нулира се когато текущата задача не е свързана с друга задача.

RF (b16) Resume flag — Управлява отговора на процесора на debug изключение.

VM (b17) Virtual-8086 mode флаг — Установява се за разрешаване на виртуален-8086 режим; нулира се при връщане към защитен режим без семантика за виртуален-8086 режим.

AC (b18) Alignment check флаг — Този флаг се установява и AM бита в CR0 регистъра за разрешаване на проверка за изравняване на адресите на операндите и се нулира AC флага и/или AM бита за забрана на проверката за изравняване.

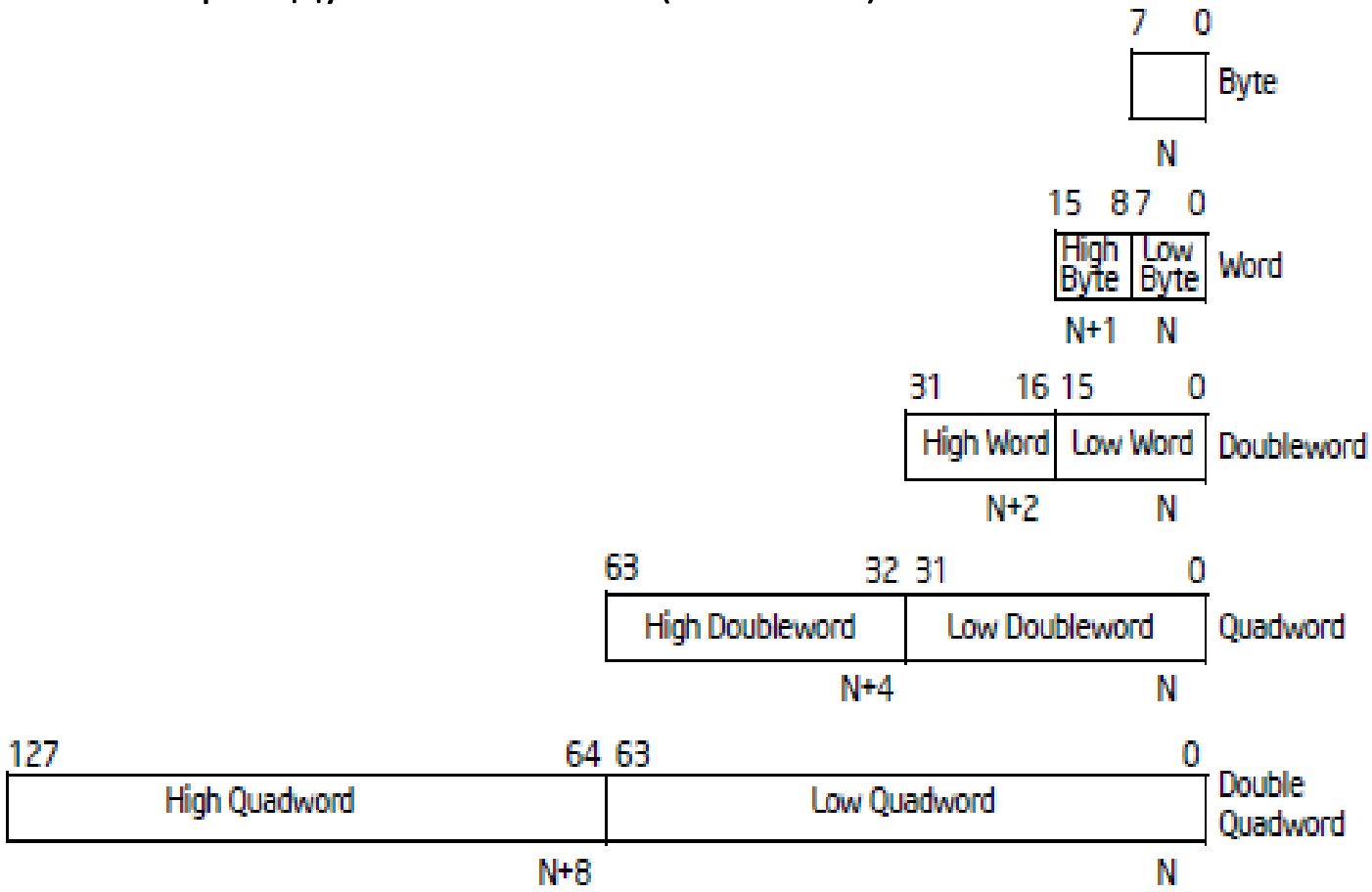
VIF (b19) Virtual interrupt флаг — Виртуален образ на IF флага. Ползва се съвместно с VIP флаг.

VIP (b20) Virtual interrupt pending флаг — Установява се за индикация на очакване на прекъсване, Нулира се когато няма чакащо прекъсване. (Софтуерът го вдига и сваля, процесорът само го чете.) Ползва се с VIP флаг.

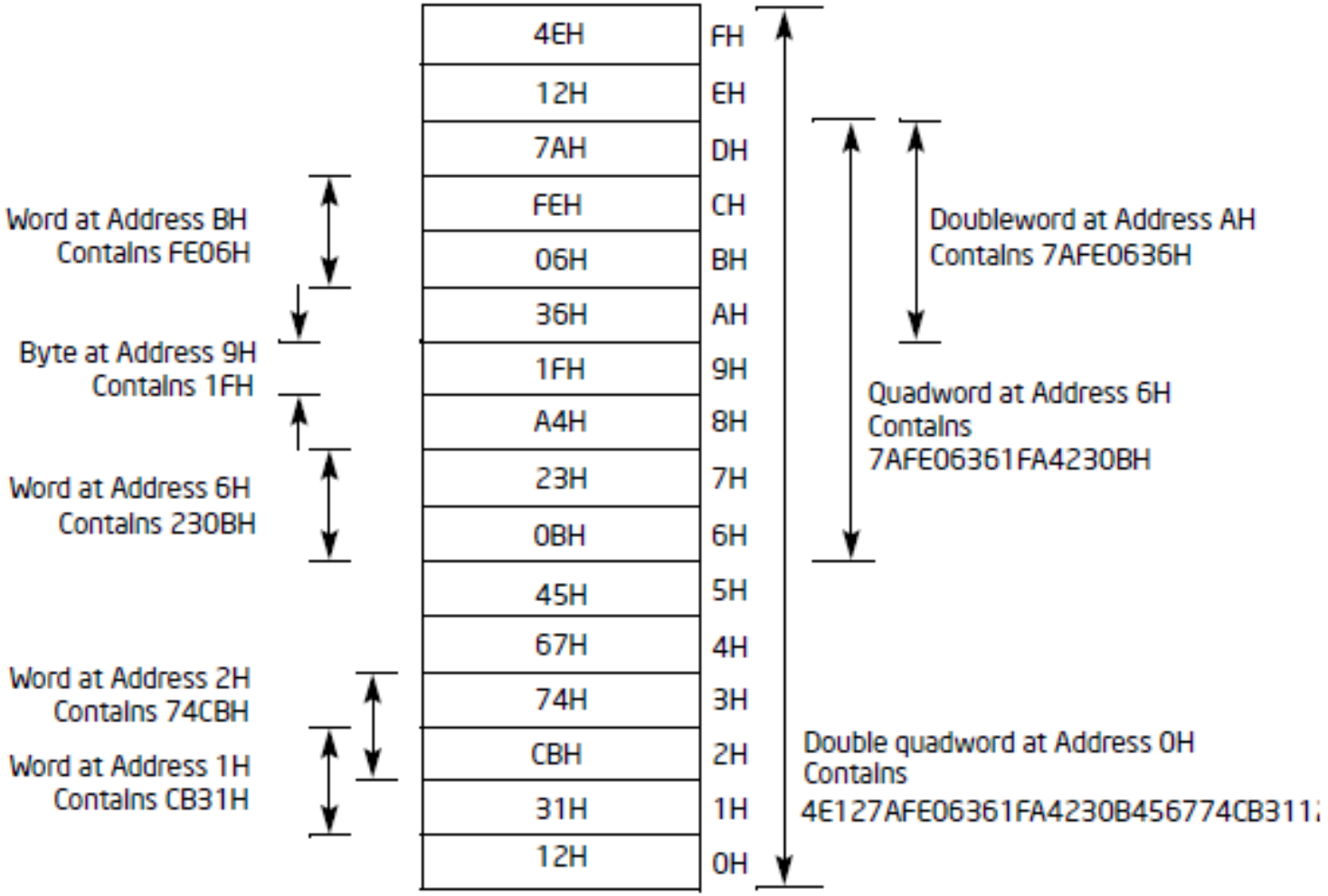
ID (b21) Identification флаг — Възможност на програмата да установява или нулира този флаг показва поддръжка на CPUID инструкциия.

Фундаментални типове данни

Фундаменталните типове данни са байтове, думи, двойни думи, четворни думи, двойни четворни думи. Байтът е 8 бита, думата е 2 байта (16 бита), двойната дума е 4 байта (32 бита), а четворната дума е 8 байта (64 бита), а двойната четворна дума е 16 байта (128 бита).



Фундаментални типове данни

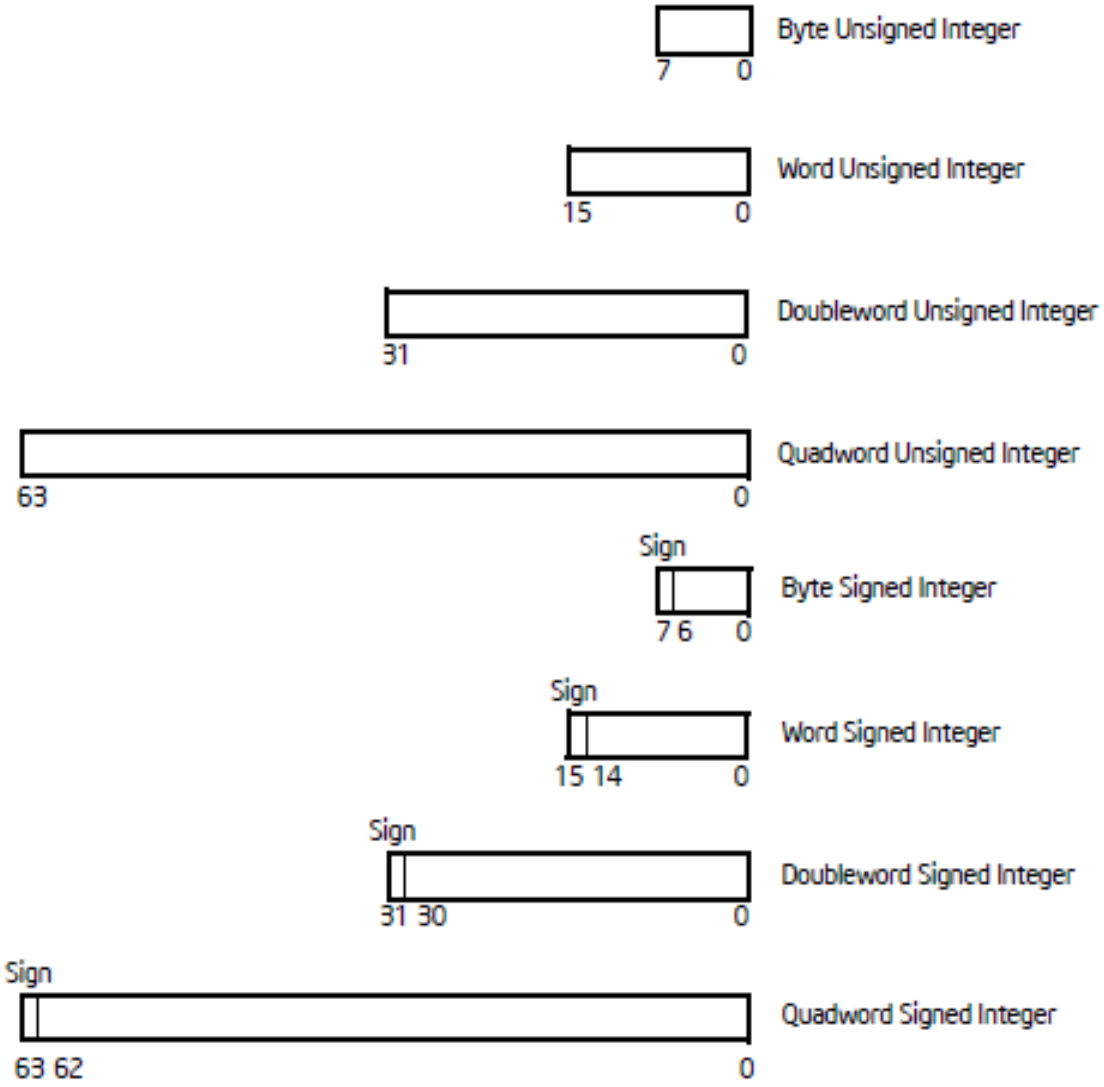


Младшият байт (бит 0-7) на всеки даннов тип заема най-ниските адреси от паметта и представлява също младшия байт на операнда.

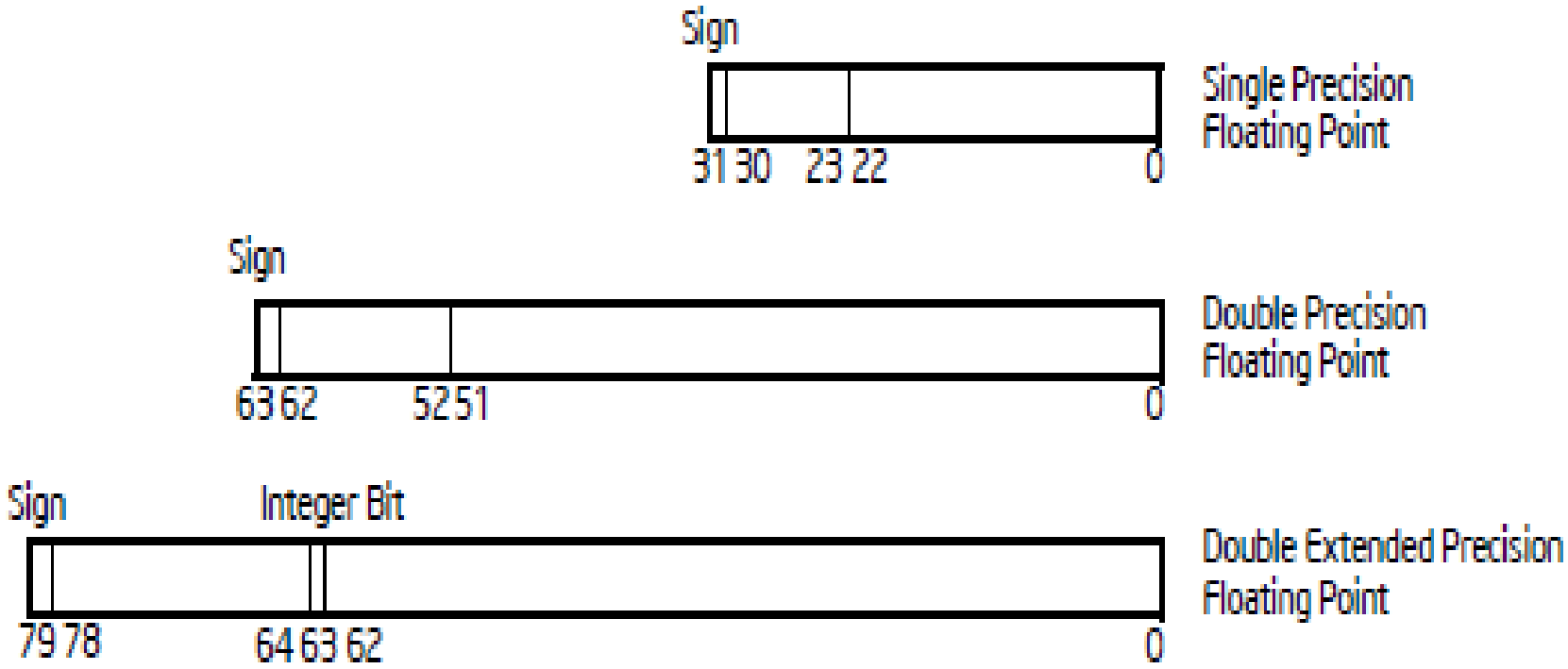
Числови типове данни

Макар че байтовете, думите и двойните думи са фундаментални типове данни, някои инструкции поддържат допълнителни интерпретации на тези даннови типове за да се разреши извършването на операции над числови типове данни (цели числа без знак и със знак и числа с плаваща запетая).

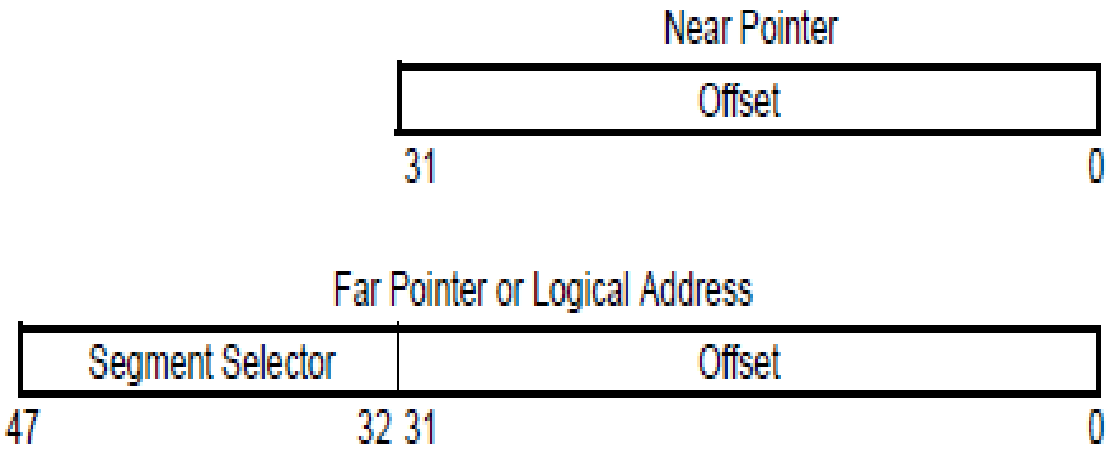
Цели числа


Byte Unsigned Integer
7
0
Word Unsigned Integer
15
0
Doubleword Unsigned Integer
31
0
Quadword Unsigned Integer
63
0
Sign
Byte Signed Integer
7 6
0
Sign
Word Signed Integer
15 14
0
Sign
Doubleword Signed Integer
31 30
0
Sign
Quadword Signed Integer
63 62
0

Числа с плаваща запетая



Указатели


Near Pointer
Offset
31
0
Far Pointer or Logical Address
Segment Selector
Offset
47
32 31
0

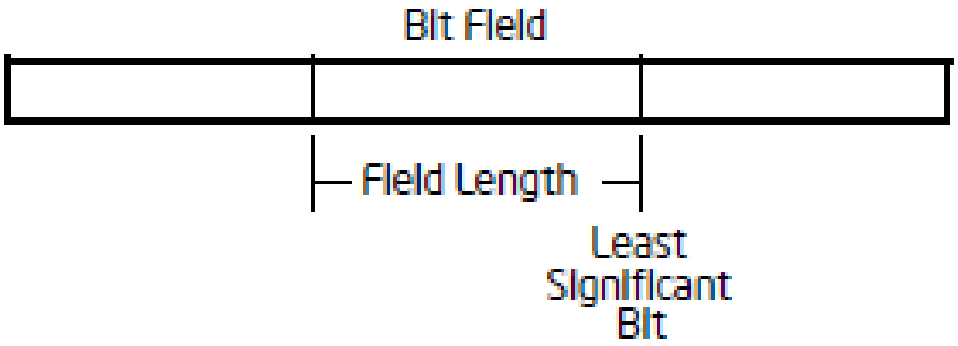
Указатели

Указателите са адресите на местоположението в паметта.

- Архитектурата дефинира 2 типа указатели:
- **Близък /near pointer.** Близкият указател е 32-b (или 16-b) offset /отместване (наричано също ефективен адрес) вътре в сегмента. Близкият указател се ползва за всички обръщения към паметта в плоския модел на паметта и за обръщения в сегментния модел на паметта, където сегментът е зададен неявно.
- **Далечен /far pointer**. Далечният указател е логически адрес, състоящ се от 16 битов сегментен селектор и 32 бита (или 16 бита) отместване /offset/. Далечният указател се ползва за обръщения към паметта при сегментен модел на паметта, където сегментът за достъп се задава явно.

Битови полета

Битовите полета са непрекъсната последователност от битове. Тя може да започва от произволна позиция в произволен байт на паметта и може да съдържа до 32 бита.



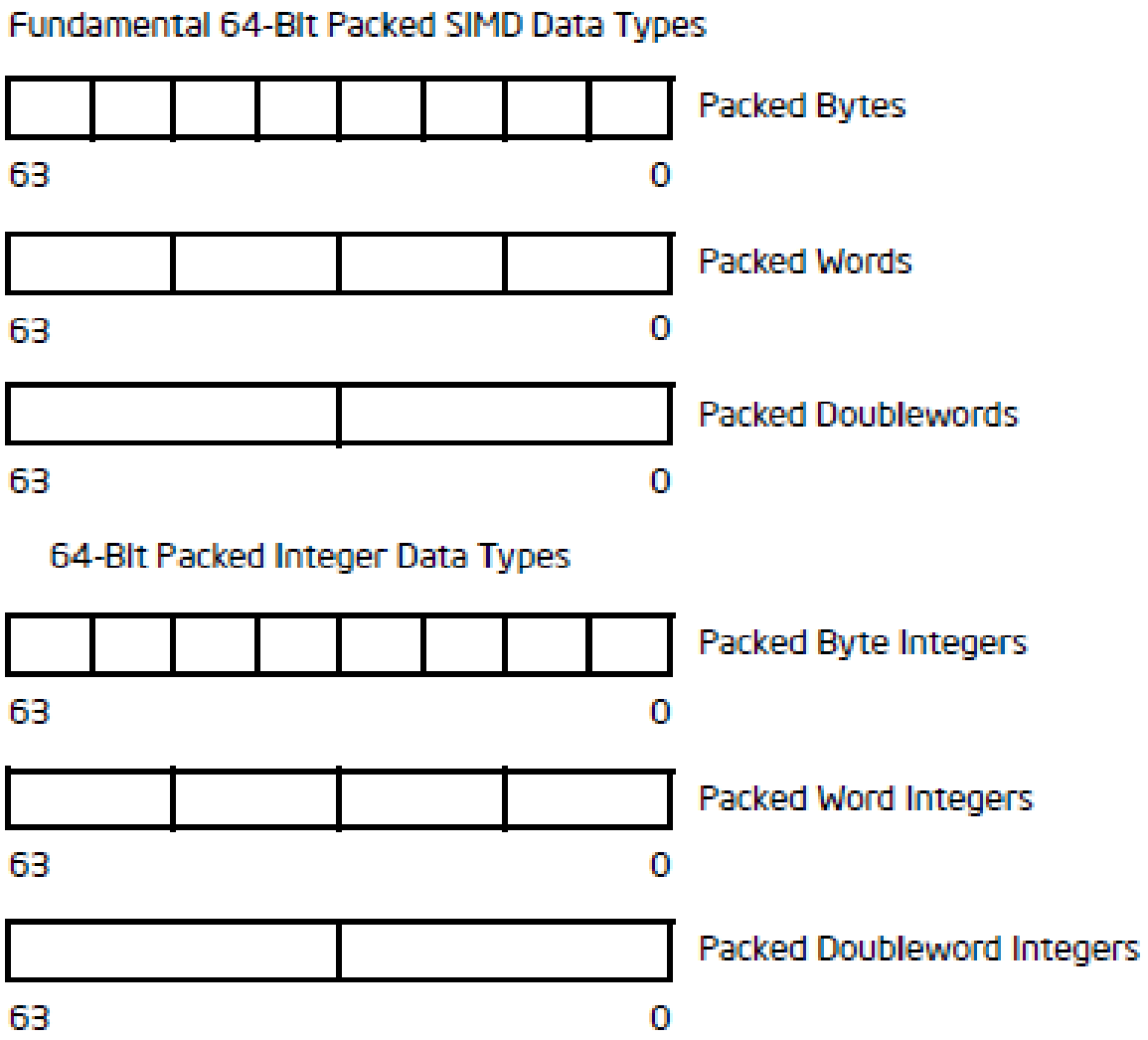
Стрингове

Стринговете са непрекъсната последователност от битове, байтове, думи или двойни думи. Битовият стринг започва от произволна позиция в произволен байт и може да съдържа до $2^{32} – 1$ бита. Байтовият стринг може да съдържа байтове, думи или двойни думи и може да има дължина от 0 до $2^{32} – 1$ байта (4 GB).

Пакетирани SIMD типове данни

Intel 64 и IA-32 архитектурите дефинират и оперират върху множество от 64-b и 128-b пакетирани даннови типове за използване в SIMD операции. Тези даннови типове се състоят от фундаментални типове данни (пакетирани байтове, думи, двойни думи и четворни думи) и числови интерпретации на фундаменталните типове за използването им в операции с цели числа и с числа с плаваща запетая..

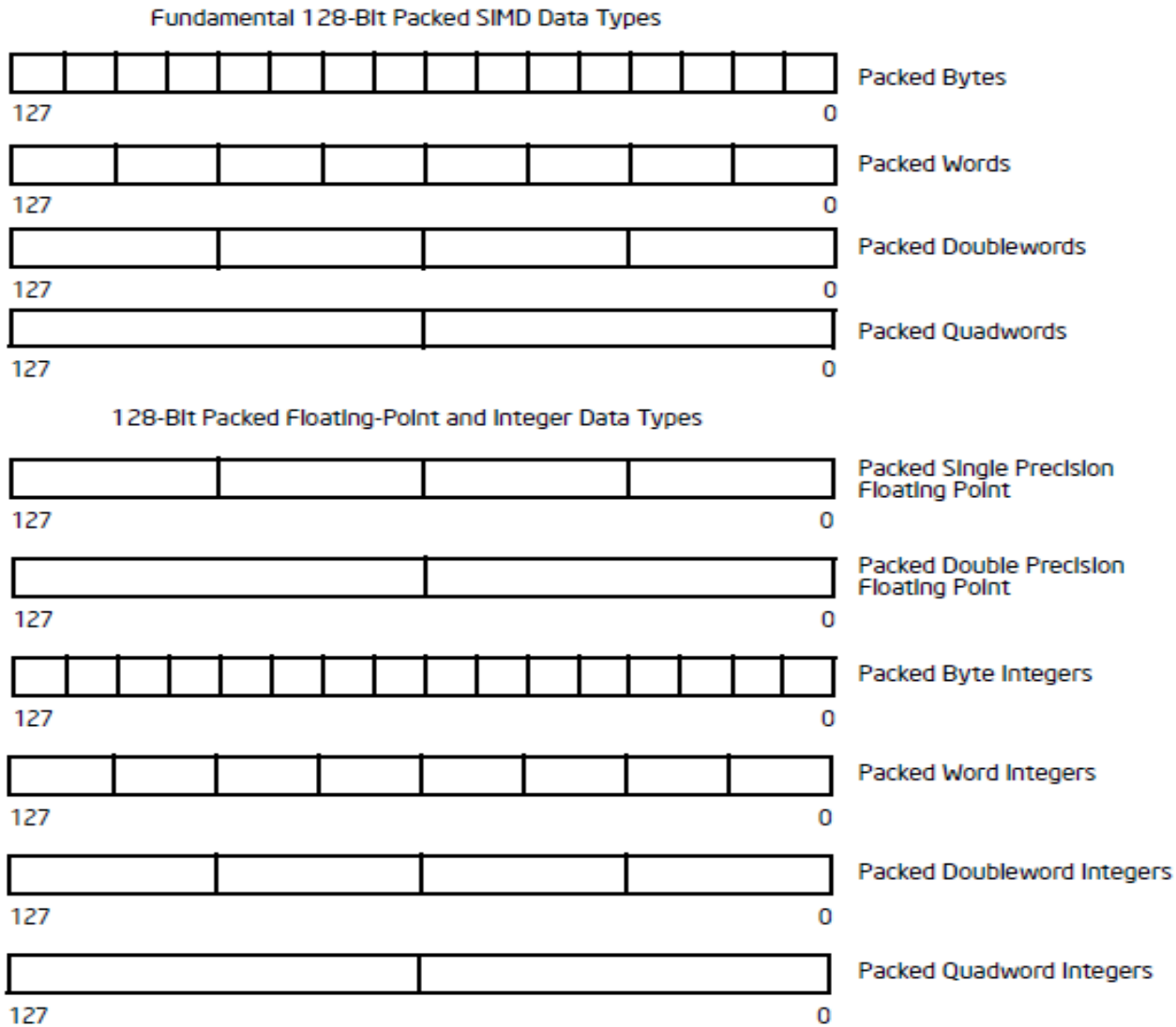
64-b пакетиран SIMD тип данни

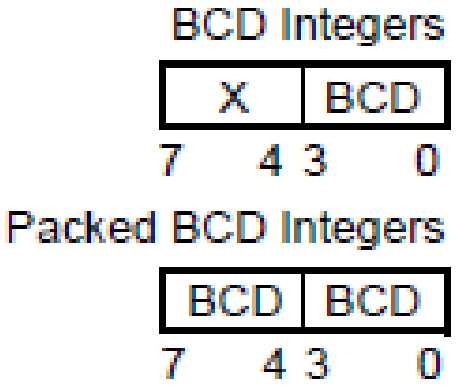


128-b пакетирані SIMD тип данни

128-b пакетиран SIMD тип данни е въведен в SSE разширенията и е ползван с SSE2, SSE3 и SSSE3 разширенията. Те се разполагат в 128-b XMM регистри и в паметта. Фундаменталните 128-b пакетиран тип данни са пакетирани байтове, пкетирани думи, пакетирани двойни думи и пакетирани четворни думи. Когато се извършва SIMD операция над фундаменталните типове данни те се разползгат в XMM регистрите. Тези типове данни се интерпретират като съдържащи пакетирани или скаларни числа с плаваща запетая с единична или с двойна точност или като цяло число в пакетирани байт, дума, двойна дума, или четворна дума.

Фундаментални SIMD типове данни 128b



BCD и пакетирани BCD цели числа

BCD целите числа са цели без знакови 4-b цели числа с валидна стойност 0-9. IA-32 архитектурата дефинира операции над BCD цели числа, разположени в един или повече регистри с общо предназначение и в един или повече регистри на x87 FPU.