

Основни конструкции на асемблерния език

1. Азбука на асемблера

Допустими символи - всички латински букви / малки и големи / , цифрите и специалните символи : + - / * < > = () _ @ & ; ; " ' []

Разликата между големите и малки букви важи при символните константи. Асемблерната програма се нарича изходен код на програмата. Основната конструкция се нарича идентификатор /име/ .

За да отделим две съседни имена в асемблерната конструкция се използват разделители / интервала е най-често използван/. Ако асемблера срещне недопустим символ, той го възприема като празен интервал.

Освен разделителя се използват специални символи, като задават края на идентификатора /ограничители/ .

ADD AX, [BX+3]

Всеки един изходен модул представлява последователност от **оператори**. Всеки един оператор заема един ред / може и повече, като продължението на реда се означава със символа & /.

Операторите се класифицират три групи:

- командни оператори;
- оператори за разпределяне на данни;
- директиви на асемблера.

1.1. Команден оператор

Всеки команден оператор съдържа така наречената мнемоничен. код на инструкцията и един или два операнда.

(*) - два операнда /акумулатор и клетка от паметта/

Форматът е следния:

{етикет:} мнемоника {операнди} {;коментар}

Етикет - име завършващо с `:`. Използва се от програмния брояч за да се движи по асемблерната програма.

Код на операцията - име, което се явява част от речника на асемблерния език и идентифицира типа на използваната команда. Командите обикновено са три буквени съкращения.

Операнди - може да липсва, а може да има 1/2 в зависимост от инструкцията, която се използва. Има случаи на повече от два операнда; макрокоманди - при които изброяваме повече от два операнда, като ги разделяме със запетая.

Коментар - отделя се с `;` / не се транслира /

1.2. Директиви на асемблера

{име} директива {операнди} {; коментар}

име - различава се от етикета, не завършва с `:` . Някои от директивите изискват задължително наличие на име /SEGMENT, ENDS, PROC и др./ . При други е задължително да няма име /NAME, ASSUME, ORG, PUBLIC и др./ . Трета група -

могат да имат и могат да нямат. Към тях спадат операторите за разпределение на данни.

Директива - съдържа ключова дума за асемблера (DW, DB, DD), която задава последващите действия на асемблера в процеса на асемблиране. Най-честият случай на използване е при разпределението на паметта. Има свързани директиви - трябва да се употребяват съвместно.

SEGMENT	PROC	CODEMACRO
ENDS	ENDP	ENDM

2. Елементи на операторите

- *ключови думи* - имена, които за асемблера имат точно определен смисъл ADD, MOV, директивите.

- *идентификатори* - определяща се от програмиста последователност от символи, като първият символ трябва да бъде буква. Допускат се @ _ , ?

- *регистри на микропроцесора* - тип BYTE, WORD.

- *променливи* - всяка една променлива се състои от три атрибута: сегмент, отместване в сегмента и тип.

```
SEGMENT  
ENDS
```

Отместването се задава с директивата OFFSET

Разстоянието между мястото на сегмента и мястото където е дефинирана дадената променлива не може да бъде по > от FFFF / от размера на сегмента 64 K /.

- *етикет* - име, което най-често се използва като операнд при команди за предаване на управлението / условен и безусловен преход /. Разстоянието на което се предава управлението се задава като двубайтово отместване. Това двубайтово отместване, когато е вътре в сегмента се нарича **NEAR**, командата за преход модифицира само програмния брояч. Когато преходът е извън границите на сегмента - прехода се дефинира като **FAR**, като при инструкцията от този тип се модифицира програмния брояч и код сегмента (PC и CS).

- *числа* - може да се представи число чрез име. За целта има директива EQU. Имената зададени чрез тази директива се интерпретират от асемблера, като явно указани от програмиста числа.

```
CR    EQU ODH  
MOV    AX, CR
```

- *числови константи* - число, което се изчислява от асемблера по време на асемблирането от даден аритметичен израз. Числовите константи, могат да бъдат записани в (2) (8) (16) (10), като това от кой тип е числото зависи от най-младшия разряд.

H	B	Q	D
16	2	8	10

- *символни вериги* - последователност от символи, които са заградени с " ' ". Максималната дължина на една такава символна верига е равна на 255 символа. Използват се за съобщения /грешка получава при определено събитие/.

```
MESSAGE DB 'RANGE ERROR'
```

Ако езика за програмиране е Асемблер съставянето на програмата минава през следните етапи:

- написване на програмата - използваме обикновен текстови редактор;
- транслиране чрез транслатор / откриват се евентуалните синтактични грешки/;
- пускане на програмата през "симулатор" - имитира работа на съответния микропроцесор - по този начин търсим логическа грешка;
- изпълнение на програмата върху хардуера т.е. програмираме EPROM слагаме върху хардуера / етап на стиковане между хардуера и софтуера/.