DSM51ASS (Asembler 8051) wersja 1.01

Podstawowe cechy asemblera DSM51ASS:
- jest makroasemblerem jednoprzebiegowym,
- komunikaty wypisuje w języku polskim (DOS strona kodowa 852),
- pozwala asemblować tylko pojedyńczy plik wejściowy (nie ma fazy linkowania),
- asemblowany program może liczyć maksymalnie około 20,000 linii listingu (zależnie od dostępnej pamięci),
- zezwala na stosowanie rozbudowanych wyrażeń arytmetycznych o postaci zbliżonej do wyrażeń języka C,
- wszystkie wyrażenia arytmetyczne liczy na liczbach 32 bitowych ze znakiem,
- wartości wszystkich symboli pamięta w postaci liczb 16 bitowych bez znaku.

Wywołanie programu: DSM51ASS <nazwa>

gdzie <nazwa> jest nazwą pliku zawierającego kod źródłowy programu. Jeśli w nazwie nie zawarto rozszerzenia to domyślnie przyjmowane jest rozszrzenie .asm. W wyniku działania programu powstają następujące zbiory:
<nazwa>.hex - zbiór zawierający kod wynikowy w formacie Intel HEX.
<nazwa>.lst - listing programu

Format linii programu: [<etykieta>] [<rozkaz>] [<operandy>] [;<komentarz>]

Znaczenie poszczególnych pól linii programu jest następujące:
<etykieta> - symbol umieszczony na samym początku linii (pierwszy znak etykiety musi być pierwszym znakiem w linii). Etykieta musi zaczynać się od litery lub znaku podkreślenia '_', i może zawierać dowolną kombinację liter, cyfr i podkreśleń. Jeśli etykieta jest zakończona dwukropkiem to nadawana jest jej wartość określająca jej pozycję w kodzie źródłowym (adres rozkazu z tej linii programu). Etykiety (symbole) stosowane z dyrektywami nadającymi im wartość nie są zakończone dwukropkiem.
<rozkaz> - mnemonik kodu maszynowego procesora, dyrektywa asemblera lub makro.
<operandy> - informacje wymagane przez mnemonik, dyrektywę asemblera lub makro. Poszczególne operandy są oddzielane przecinkami.
<komentarz> - wszystkie znaki występujące po średniku są traktowane jako komentarz i ignorowane przez asembler.

Poszczególne pola linii programu muszą być oddzielone między sobą co najmniej jednym znakiem spacji (lub tabulacji). W programie mogą występować puste linie lub linie zawierające wyłącznie komentarz.

Stałe liczbowe.

Stała liczbowa musi zaczynać się od cyfry.
DSM51ASS akceptuje następujące typy stałych liczbowych:

Typ	Składnia	Przykład
Dziesiętny	<cyfry>	125
Szesnastkowy	<cyfra><cyfry szesnastkowe>H	0FFFFH
Ósemkowy	<cyfry ósemkowe>O	7777O
Binarny	<cyfry binarne>B	10101B
Znakowy	'<znak>'	'A'

Dyrektywy asemblera.

W poszczególnych liniach programu oprócz mnemoników oznaczających poszczególne rozkazy procesora mogą wystąpić dyrektywy asemblera. Umożliwiają one wstawianie danych w treść programu, przypisywanie wartości symbolom, sterowanie przebiegiem asemblacji i budowanie makr (zestawów poleceń wywoływanych pojedyńczą nazwą).
Asembler DSM51ASS akceptuje następujące dyrektywy:

Dyrektywy danych.

DB - wstawienie w kod wartości numerycznych i tekstowych Składnia: [<etykieta>] DB <parametry>

Wpisuje w treść programu wartości parametrów. Poszczególne parametry oddzielane są przecinkami. Parametry mogą być wyrażeniami arytmetycznymi lub ciągami znaków ujętymi w znaki ' lub ". Wartości kolejnych parametrów będących wyrażeniami arytmetycznymi są wpisywane w kolejne bajty treści programu. Parametry będące ciągami znaków są wpisywane w całości do treści programu. W ciągu znaków ujętym w znaki " może wystąpić znak ' i odwrotnie.

DW - wstawienie w kod dwubajtowych wartości numerycznych Składnia: [<etykieta>] DW <parametry>

Wpisuje w treść programu wartości parametrów. Poszczególne parametry oddzielane są przecinkami. Parametry są wyrażeniami arytmetycznymi. Wartość każdego parametru jest wpisywana w dwa kolejne bajty treści programu (najpierw starszy).

EQU - definiowanie stałej. Składnia: <symbol> EQU <wyrażenie>

Symbolowi <symbol> przypisywana jest wartość wyrażenia. Typ symbolu ustalany jest na podstawie wyrażenia. Każda wartość zdefiniowana dyrektywą EQU jest stałą i nie może być zmieniana w trakcie asemblacji.

BIT - definiowanie stałej typu bit. Składnia: <symbol> BIT <wyrażenie>

Symbolowi <symbol> przypisywana jest wartość wyrażenia. Kontroluje typ wyrażenia. Zdefiniowany symbol może być używany wyłącznie jako adres bitu. Wartość symbolu nie może być zmieniana w trakcie asemblacji.

REG - definiowanie stałej typu rejestr. Składnia: <symbol> REG <wyrażenie>

Symbolowi <symbol> przypisywana jest wartość wyrażenia. Kontroluje typ wyrażenia. Zdefiniowany symbol może być używany wyłącznie jako adres rejestru wewnętrznego mikrokontrolera lub komórki wewnętrznej pamięci RAM. Wartość symbolu nie może być zmieniana w trakcie asemblacji.

SET - definiowanie zmiennej. Składnia: <symbol> SET <wyrażenie>

Symbolowi <symbol> przypisywana jest wartość wyrażenia. Typ symbolu ustalany jest na podstawie wyrażenia. Wartości zdefiniowane dyrektywą SET mogą być dowolnie wiele razy modyfikowane przez ponowne użycie dyrektywy SET. Zmiana typu symbolu w trakcie kolejnego przypisania powoduje wygenerowanie ostrzeżenia.

Dyrektywy sterujące.

IF - początek bloku warunkowej asemblacji. Składnia: IF <wyrażenie>

Jeżeli wartość wyrażenia jest różna od '0' (prawda) to kod występujący za tą dyrektywą jest asemblowany. Jeżeli wartość wyrażenia jest równa '0' (fałsz) i istnieje dyrektywa ELSE to kod występujący po ELSE jest asemblowany. Dyrektywa ENDIF zamyka blok kodu asemblowanego warunkowo. Dopuszczalne jest zagłębianie dyrektyw IF do 16 poziomów.

ELSE - początek alternatywnego bloku warunkowej asemblacji. Składnia: ELSE

Używana w połączeniu z dyrektywą IF. Jeśli wyrażenie testowane w dyrektywie IF ma wartość '0' to alternatywny kod zaznaczony przez ELSE jest asemblowany.

ENDIF - koniec bloku warunkowej asemblacji. Składnia: ENDIF

Zakończenie bloku warunkowej asemblacji rozpoczętego dyrektywą IF.

ORG - ustawienie adresu dla następnego bloku kodu. Składnia: ORG <wyrażenie>

Ustawienie adresu dla następującego po tej dyrektywie bloku kodu. Adres dla następnej instrukcji procesora jest ustalany poprzez wyliczenie wartości wyrażenia. Możliwe jest jedynie zwiększanie aktualnego adresu kodu. Próba zmniejszenia adresu jest sygnalizowana jako błąd. Standardowo kod programu jest umieszczany rozpoczynając od adresu 0.

END - koniec programu. Składnia: END

Zaznaczenie końca programu. Linie występujące w pliku źródłowym po tej dyrektywie nie są asemblowane. Użycie tej dyrektywy w programie nie jest konieczne. Przy jej braku końcem programu jest koniec pliku.