Директиви Асемблера

**Директива AREA**

Створює сегмент даних або команд(коду) виходячи з заданих атрибутів до сегменту.

Синтаксис:

**AREA** [Section\_name], {type,..}, {attr,..}, {align}

Параметри:

**Section\_name** - ім’я сегменту. Може називатися будь якою назвою. Якщо імя починається не з літери алфавіту - воно повинно бути заключене у вертикальні лінії. Наприклад:  AREA |.krnv|. Імя RESET - зарезервоване. Сегмент з цим імям буде розташовано на початку памяті.

**type -** Тип сегменту. Може бути або CODE, або DATA.

*CODE* - сегмент коду. Розміщуються лише команди. Після закінчення команд програми ініціалізується командами DCD.

*DATA* - сегмент даних.  При старті або рестарті ініціалізується нулями. Розміщуються лише дані, розміщення команд недопустиме.

**attr** - атрибути сегмента.

*NOINIT* - сегмент даних що при рестарті МК не буде ініціалізуватися, а зберігати свої значення. Може бути використаний лише для сегменту даних, тому тип сегменту можна не писати.

*COMMON* - спільний сегмент даних. Можуть розміщуватися як команди, так і дані. При старті або рестарті окрім команд  память заповнюється нулями. Цей атрибут може бути використаний лише для сегменту даних.

*READ* - сегмент лише для читання.

*READWRITE* - сегмент що використовується і для читання і для запису. (READ - CODE, READWRITE - DATA)

*ALIGN* - вирівнювання сегмента по байтам. Може бути ціле число від 0 до 31. За замовчуванням для сегменту ALIGN дорівнює двум.

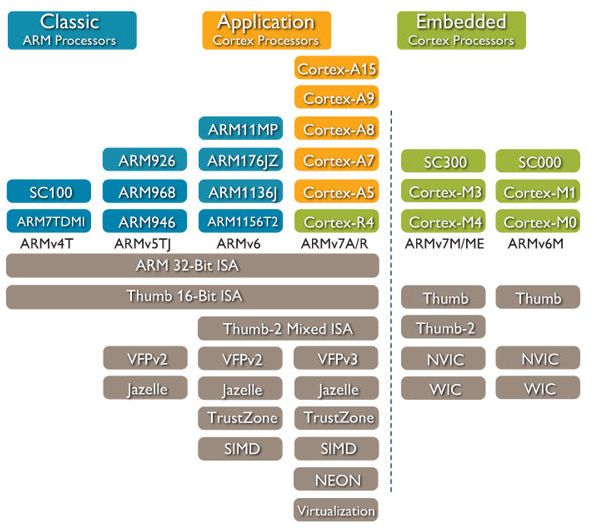
**Не використовуйте ALIGN = 0 для системи команд THUMB!!**

Інші директиви

***PREVERVE8*** - зберігати 8-байтне зміщення SP. Якщо SP буде зміщенно не на 8 (а на 4, 12, 20 і так далі) то компілятор генеруватиме warning. Для багатьох архітектур необхідне саме 8-байтне зміщення SP. Зокрема для [Procedure Call Standard for the ARM Architecture [AAPCS]](http://infocenter.arm.com/help/topic/com.arm.doc.ihi0042d/index.html).

***THUMB***, ***THUMBX***, ***ARM*** - директиви, команди після яких інтерпретуються як команди thumb-thumb2/thumbEE/arm відповідно.

Нижче представлено для яких процессорів який набір команд може бути використано.



В нашому курсі використовуються процессори Cortex M3 або M4, тому можна використовувати як і thumb так і thumb2 набір команд. Thumb2 доповнений до thumb 32-байтними командами з набору інструкцій ARM. При використанні директиви THUMB будуть використовуватися команди з THUMB і THUMB2 набору інструкцій, а при використанні інших макросів програма видасть помилку.

{*label*} **SPACE** *expr -* резервує кількість байт вказану в expr і заповнює цю память нулями.

{*label*} **FILL** *expr*{,*value*{,*valuesize*}} -  резервує кількість байт вказану в expr і заповнює цю память нулями. Додатково може бути вказано:

value: Якими значеннями заповняти память. За замовчуванням - 0х0.

valuesize: Розмір заповнюючого значення value. Може бути 1,2 або 4 байти

{label} **DCD** expression{,expression}

Директива DCD виділяє одне або кілька слів пам'яті, вирівняні на кордонах чотири байти, і визначає початкове вміст середовища виконання пам'яті.

& є синонімом DCD.

**DCDU** та сама директива, але не вирівнює память по кордоні у чотири байти.

Директива **EXPORT** оголошує символ, який може бути використаний лінковщиком для дозволу посилань символів в окремих об'єктах і файлах бібліотек. GLOBAL є синонімом для EXPORT.