

студент: група: преподавател:

ЕМК MC68HC11. СИСТЕМА ЗА РАЗВИТИЕ EVBplus2

I. ТЕОРЕТИЧНА ПОСТАНОВКА - АПАРАТНИ И ПРОГРАМНИ СРЕДСТВА

➤ ЕДНОЧИПОВ МИКРОКОМПЮТЪР MC68HC11E1

- Архитектура: микропроцесор, шини, регистри, памет, периферия.
- Режими: основни - ЕМК (*single chip*) и разширен (*expanded*); допълнителни - *Test*, *Boot*.
- Карта на паметта на ЕМК MC68HC11E1
- Формат на командите – код на операцията, операнди.

➤ СИСТЕМА ЗА РАЗВИТИЕ EVBplus2

- Предназначение, възможности, режимы на работа.
- Устройство (блокова схема, принципна схема, платка).
- Разпределение (карта) на адресното пространство в EVBplus2.

➤ ИНТЕГРИРАНА ПРОГРАМНА СРЕДА AsmIDE

- Мониторна програма BUFFALO.
- Редактор – въвеждане и редактиране на програма на Асемблер.
- Асемблиране (създаване програма в обектен код), зареждане и изпълнение с развойната платка.



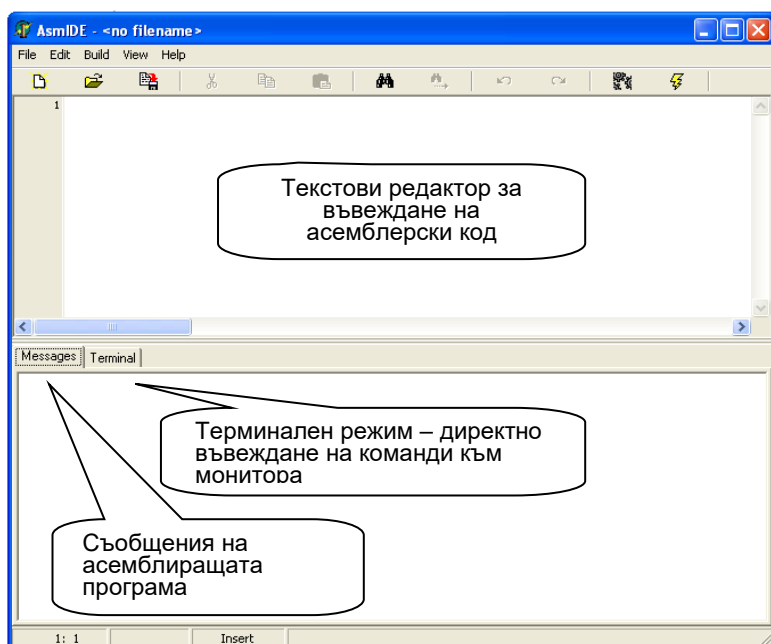
Стартира се приложението **AsmIDE** от неговата графична иконка или от изпълнимия файл в съответната директория: C:\Ep2IDE\AsmIDE.exe. Върху екрана се появява прозорец, разделен на две:

❖ *Terminal* – директно изпълнение на команди от мониторната програма (в това упражнение се работи само в този режим).

За да изберете този режим, натиснете *Terminal* и влезте в долния прозорец. Натиснете ENTER за да се появи напомнящият знак ➤, което показва, че мониторът контролира системата. Ако напомнящият знак не се появи, натиснете бутон RESET (SW2) от развойната платка,

при което се появява: *EVBplus.com: BUFFALO 3.43 - Bit User Fast Friendly Aid to Logical Operation*

Режимите, изпълнявани в горния екран, са свързани с асемблиращата програма. Те ще бъдат използвани и разгледани подробно в следващото упражнение.



✓ Основни команди на монитор BUFFALO

RM (register modify) - четене и запис в регистрите. При влизане в монитора, регистрите на изпълняваната програма се съхраняват, а след излизане се възстановяват. Командата RM променя съдържанието на съхранените регистри, а не на текущите в монитора.

MM (memory modify) – четене/запис в клетка от паметта;

MD (memory dump) – изобразяване съдържанието на област от паметта;

MOVE (move memory) – преместване на област от паметта;

ASM – асемблиране/деасемблиране на команден ред

G – изпълнение на програма; **CALL** – извикване на подпрограма;

BR (breakpoint set) – слагане/махане на точки на прекъсване;

P (proceed) – продължаване на изпълнението на програмата;

T (trace) – изпълняване на определен брой инструкции.

> **RM** - показва съдържанието на регистрите и очаква да бъде променен програмния брояч **P**:

P-FFFF 0000 ENTER - променя съдържанието на програмния брояч

ENTER (без въвеждане) - излиза от командата без промяна

SPACE BAR - изобразява следващите регистри

> **RM X** - показва съдържанието на регистрите и очаква да бъде променен индексния регистър **X**:

P-FFFF Y-FFFF X-FFFF A-FF B-FF C-D0 S-0041 - показва съдържанието на регистрите

X-FFFF 8F00 ENTER - променя съдържанието на индексния регистър **X**

> **MM** <адрес> - показва съдържанието на клетка от паметта, напр.:

>**MM 100** - показва съдържанието на клетка с адрес \$100:

0100 FF 86 ENTER - заменя съдържанието **FF** с **86** и излиза от командата

ENTER (без въвеждане) - излиза от командата без промяна

SHIFT+ - показва съдържанието на следващата клетка на нов ред

SPACE BAR - показва съдържанието на следващата клетка на същия ред и очаква да бъде заменено с ново (може да се повтаря многократно)

, BACKSPACE - показва съдържанието на предишната клетка

=, / - показва съдържанието на същата клетка

ENTER (без въвеждане) - излиза от командата; повторно **ENTER** - влиза отново.

> **MD** <адрес 1> <адрес 2> - показва съдържанието на област от паметта като между двата адреса като 16-ични стойности (и като ASCII символи в дясната част, ако са видими)

> **MD** - показва съдържанието на 9 реда по 16 байта от последния използван адрес.

▪ Въвеждане на програма в обектен (машинен) код с помощта на команда **MM**

Това е възможно най-ниското ниво на програмиране, предлагано от мониторната програма, при което за удобство на човека командите на микрокомпютъра (колона 5 от таблицата от *Reference Guide*) се въвеждат в определена област от паметта директно в шестнадесетичен вид. Мониторът разпознава само собствените си команди в символичен вид; всичко останало предполага че е записано в шестнадесетичен вид!

Пример – въвеждане на програма в шестнадесетичен вид, която променя съдържанието на някой от регистрите и след изпълнението си го показва (използваме регистърът на порт В, защото е индициран със светодиоди върху развойната платка). Програмата се разполага в свободно място на паметта RAM (напр. от адрес \$0100) като кодовете на операциите и операндите се въвеждат на последователни адреси чрез командата **MM**. Въпреки, че всички те могат да бъдат разположени на един ред или всеки един на отделен ред, прието е за четливост да се групират по една инструкция на ред. Символичните означения на инструкциите и обясненията в сив цвят са написани в този текст само за подпомагане на читателя.


0100	86 80	LDAA #80	Зарежда \$80 в акумулатор А
0102	B7 10 04	STAA 1004	Изпраща го в порт В
0105	3F	SWI	Програмно прекъсване (край)

Тук първо се зарежда числото 80 (HEX) в акумулатора А. След това съдържанието му се изпраща в порт В и светодиодите го индицират. Накрая програмата се прекъсва и управлението на микрокомпютъра се предава на мониторната програма, която визуализира съдържанието на регистрите в момента на прекъсването. Програмата се стартира с командата **GO** (кратко **G**). Може да бъде изпълнявана многократно при различни стойности на числото, зареждано в акумулатор А, така че да светват различни светодиоди.

▪ Въвеждане на програма в символичен вид чрез асемблера на монитора – команда **ASM**

Това е много по-удобен начин, при който командите на микрокомпютъра (стр. 487 на *Reference Manual*) се въвеждат в техния символичен вид (кол.1 от табл. на стр.13 на *Reference Guide*), а вграденият асемблер за 68HC11E1 ги преобразува веднага след натискане на **ENTER** в шестнадесетичен вид (обектен код).

Пример – въвеждане на същата програма в символичен вид чрез вградения в монитора асемблер от адрес \$D000 в следния ред:

- Влиза се в режим на асемблер като в монитора (при наличие на напомнящ знак ) се въвежда командата **ASM**, последвана от началния адрес на програмата (разделени с интервал) и се натиска ENTER. В примера по-долу е използван различен от горния адрес.
- Напомнящият знак се измества вдясно, което е указание че се намираме в асемблера на монитора и той се опитва да деасемблира (превърне от шестн. в символичен вид) информацията, записана от този адрес
- Въвежда се поредната инструкция на микрокомпютъра (кодовете на операциите в символичен, а стойностите на операндите – в шестнадесетичен вид) и се натиска ENTER. Асемблерът ги преобразува веднага в шестнадесетичен вид.
- Излиза се от режим на асемблер със знака „.” (точка) или **CTRL-D**
- Стартира се програмата с командата **GO (G)**.

D000	xxxxxxx 86 80	>LDAA #80	Зарежда \$80 в акумулатор A
D002	xxxxxxx B7 10 04	>STAA 1004	Изпраща го в порт B
D005	xxxxxxx 3F	>SWI	Програмно прекъсване (край)

Тук символичните означения на инструкциите са реални, а само обясненията в сив цвят са написани в този текст само за подпомагане на читателя (режимът асемблер на монитора няма такава възможност)!

II. ЗАДАЧИ ЗА ИЗПЪЛНЕНИЕ

1. Запознайте се с блоковата схема на ЕМК MC68HC11 - архитектура, състав, шинна организация, режими на работа.

2. Разучете развойната платка EVBplus2: предназначение, възможности, режими на работа, блокова схема, изпълнение, съответствие на елементите върху платката и на принципните схеми.

3. Наблюдавайте с осцилоскоп сигналите на системните шини (E, R/W, A0-A15, D0-D7 и CS), изведени върху изводите на съединителя (куплунга), разположен над LCD дисплея на кита, като синхронизирате осцилоскопа по тактовия сигнал E. Защо не можете да спрете сигналите неподвижни на екрана?

4. Запознайте се с възможностите на мониторната програма BUFFALO, вградена в интегрираната програмна среда AsmIDE като изпълните следните задачи:

4.1. С помощта на командата **RM** прочетете съдържанието на основните регистри, променете ги и проверете резултата.

4.2. Използвайте командата **MM** за да прочетете съдържанието на клетки от паметта RAM, променете ги и проверете резултата.

4.3. Като използвате командата **MM**, въведете името си в свободна област от RAM паметта (например, с начален адрес \$0100) като поредица от ASCII символи. След това разгледайте областта с помощта на командата **MD** и проверете дали правилно се изписва името ви в дясната част на полето.

4.4. Изключете захранването на развойната платка, изчакайте няколко секунди и го включете отново. Проверете дали въведената от вас информация се съхранява в паметта и обяснете защо.

4.5. Въведете програмка в обектен код „86 F0 B7 80 00 20 F9”, която непрекъснато записва една и съща стойност (в примера – F0) в клетка от потребителската RAM (U4), разположена в адресната област \$8000-\$DFFFF (за да видите програмката в символичен вид, деасемблирайте кода чрез командата ASM). Наблюдавайте с осцилоскоп сигнала CS (извод 20 на U4) чрез канал 1 като синхронизирате осцилоскопа по него (защо?), а останалите сигнали (E, R/W, A0-A15 и D0-D7) - чрез канал 2, като местите сондата последователно по шините, изведени на системния куплунг J1. Прочетете адреса и стойността на записваното число (можете ли да го направите само с помощта на един светодиод?)

4.6. Въведете програмка в обектен код „B6 80 00 20 FB”, която непрекъснато чете съдържанието на същата клетка от потребителската RAM. Наблюдавайте по описания по-горе начин адреса на избраната клетка и стойността на прочитаното число.