

МИКРОПРОЦЕССОРНА ТЕХНИКА

ЛЕКЦИЯ #7

ЕМК HC11 – ПРОГРАМЕН МОДЕЛ (РЕГИСТРИ)

Включва наличните регистри в ЦП:

ACC A (акумулатор A) - 8-битов

ACC B (акумулатор B) - 8-битов или общо → **16-битов акумулатор ACC D**

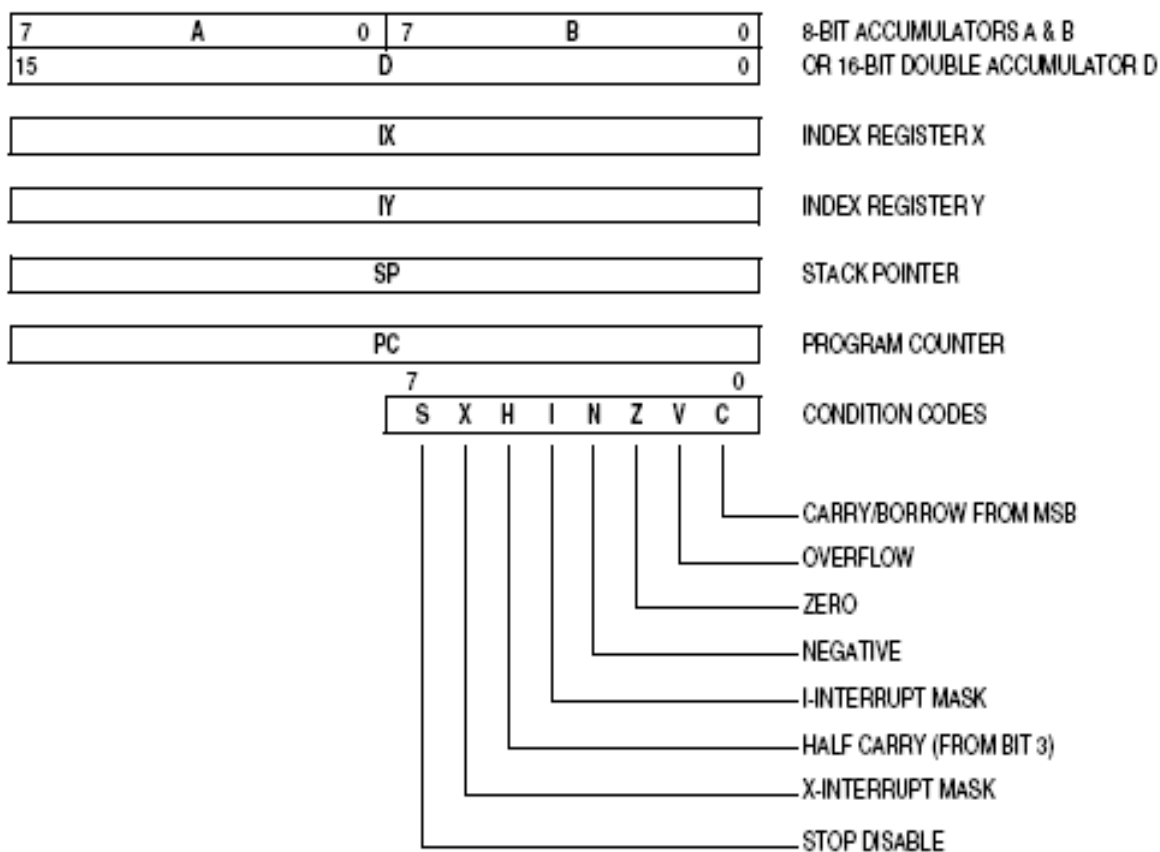
IX (индексен регистър X)

IY (индексен регистър Y)

SP (указател на стека)

PC (програмен брояч)

CCR (регистър на състоянието, 8-битов).



ЕМК HC11 – ПРОГРАМЕН МОДЕЛ (РЕГИСТРИ)

Акумулатори А и В – 8-битови регистри с общо предназначение. Съхраняват временно операндите (данни) и резултата от изпълнение на предишна операция от АЛУ. За някои инструкции образуват 16-битов регистър (акумулатор D).

Равнозначност на А и В с изключение на:

- инструкции **ABX, ABY** добавят съдържанието на акумулатор В към индексни регистри X или Y (без такива за акумулатор А);
- инструкции **TAP, TPA** прехвърлят данни от акумулатор А към регистъра на състоянието и обратно (липсват такива за В);
- инструкция **DAA** (Decimal Adjust accumulator A) – след BCD аритметични операции (не съществува подобна за акумулатор В);
- еднопосочност на операциите събиране, изваждане и сравнение (**ABA, SBA, CBA**).

ЕМК НС11 – ПРОГРАМЕН МОДЕЛ (РЕГИСТРИ)

Индексен регистър IX - при индексна адресация осигурява 16-битова стойност (базов адрес), която да се добави към 8-битовото отместване от инструкцията за образуване на ефективен адрес (ЕА).

Допуска изпълнение на операции INX, DEX, CPX.

Може да се ползва като брояч или за временно съхранение на данни.

Индексен регистър IY - 16-битов. Подобно на IX участва в индексен адресен режим. **Повечето инструкции, ползващи IY, изискват допълнителен байт от кода и цикъл за изпълнение.**

ЕМК HC11 – ПРОГРАМЕН МОДЕЛ (РЕГИСТРИ)

SP (указател на стека). ЦП има автоматичен програмен стек ***Разположен е произволно в адресното пространство и може да има произволен размер (дълбочина на стека).*** Използва се за съхранение текущото съдържание на регистрите, адрес при връщане, общи данни. LIFO тип.

Във всеки момент SP (управлява се от УУ) съдържа 16-битовия адрес на първата свободна клетка от стека. SP се инициализира с някоя от от първите инструкции в приложната програма.

Организация – нарастваща структура от горните към долните адреси.

Със запълване (запис) на стека, SP се намалява с -1, с изпразване (четене) се увеличава с +1.

Ползване на стека и SP – при обработка на прекъсвания, работа с подпрограми, временно съхранение на данни, при рекурсии.

ЕМК HC11 – ПРОГРАМЕН МОДЕЛ (РЕГИСТРИ)

РС (програмен брояч) – 16-битов, съдържа абсолютния адрес на следващата команда за изпълнение. След Reset, РС се инициализира с един от 6-те вектора (в зависимост от режима):

Mode	POR or RESET Pin	Clock Monitor	COP Watchdog
Normal	\$FFFE, F	\$FFFC, D	\$FFFA, B
Test or Boot	\$BFFE, F	\$BFFC, D	\$BFFA, B

CCR (регистър на състоянието). НЕ съдържа данни.

Съдържа 8 бита (индикатори/флагове и маски), от които:

- 5 индикатори на състоянието (C, V, Z, N, H);
- 2 маскови бита при прекъсване (IRQ, XIRQ);
- бит за освобождаване от стоп режим (S).

ЕМК НС11 – ПРОГРАМЕН МОДЕЛ (РЕГИСТРИ)

□ Индикатори на състоянието (C, V, Z, N, H)

Carry/Borrow (C) – пренос. В “1” при реализация на пренос от аритметични операции. Служи още като флаг за грешка при операции умножение и деление. Използва се от операции изместване и ротиране;

Overflow (V) – препълване. Установява се в “1” при аритметично препълване. В противен случай е “0”;

Zero (Z) – нулева стойност. Установява се в “1” ако резултатът от аритметична, логическа операции (обработка на данни) е 0. Използва се при операция сравнение (включени изваждания, ползва се Z флага). **Операции INX, DEX, INY, DEY засягат САМО флага Z.** При тях се определят само състояния = и ≠.

ЕМК НС11 – ПРОГРАМЕН МОДЕЛ (РЕГИСТРИ)

□ Индикатори на състоянието (C, V, Z, N, H)

Negative (N) – отрицателна стойност. Установява се в “1” ако резултатът от аритметична, логическа операция или при обработка на данни е отрицателен (MSB=1, знаков бит). В противен случай е “0”;

Half Carry (H) – полупренос. Установява се в “1” при пренос между битове 3 и 4 от АЛУ при инструкции ADD (A,B,D), ABA, ADC (A,B). В противен случай е “0”. Използва се при операции в BCD код.

ЕМК HC11 – ПРОГРАМЕН МОДЕЛ (РЕГИСТРИ)

❑ Маскови бита I, X при прекъсване (IRQ, XIRQ);

Interrupt Mask (I) – маска за прекъсване (глобална). Забранява всички маскирани прекъсвания. При I = 1 прекъсвания могат да постъпват, но ЦП продължава работата си докато I стане "0". След Reset се установява в 1. Може да се нулира само SW. След обслужване на прекъсването, състоянието на регистрите се възстановява в предишното и бита I = "0";

X Interrupt Mask (X). Забранява прекъсванията от извод XIRQ. След Reset се установява в "1" по подразбиране и може да се нулира само SW. При разпознаване на XIRQ прекъсване, битове X и I се установяват в "1", RTI се изпълнява нормално и регистрите възстановяват стойността си.

X маската за прекъсване се установява в "1" само хардуерно (RESET, XIRQ) и се нулира само програмно (чрез TAP, при съответния бит от A = 0 или с RTI, при бит 6 от CCR = 0). Не е възможно X да се нулира хардуерно.

❑ **STOP Disable (S)**. Бит за освобождаване от режим STOP (режим с ниска консумация). При установяване в "1" предпазва навлизането на ЕМК в режим STOP. В това състояние, при постъпване на инструкция STOP, тя се третира като безоператорна и се изпълнява следващата инструкция. Бит S се установява в "1" след Reset.

ЕМК HC11 – ПРОГРАМЕН МОДЕЛ (РЕГИСТРИ)

Типове данни в ЕМК M68HC11:

1. Данни тип “единичен бит” (побитови данни);
2. 8-bit (байт) / 16-bit (дума) цели числа със/без знак;
3. 16-bit дробни без знак;
4. 16-bit адреси.

1 дума = 2 байта (разположени в два съседни байта, като MSB е в по-долния адрес от адресното пространство).

Команда, КОД (КОП), формат, адресация, предназначение, Методи за адресация (типове адресация) в ЕМК НС11

НС11 използва 8-битов КОД (Код на операцията, Opcode). **Всеки КОД идентифицира определена инструкция и метод на адресация.**

КОД (КОП) – до 256 инструкции.

Допълнителен байт - **PRE BYTE** за увеличаване броя команди (препраща ЦП към съотв. страница от 4 възможни). Предшества КОД в общия формат.

Пълна инструкция – [PRE BYTE], Opcode, без , с 1 или 2 операнда.

Всички методи за адресация (без вътрешна) използват ефективен адрес (ЕА). ЕА- адрес от паметта, от който се вземат аргументите или от който се продължава изпълнението на програмата.

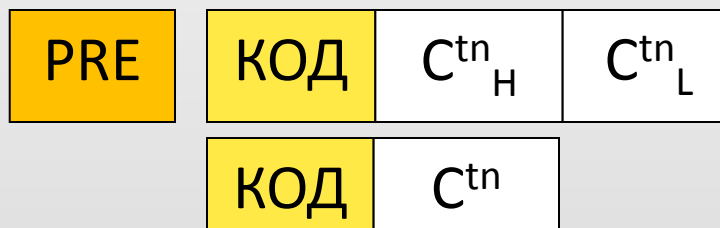
ЕА – задава се в рамките на инструкцията или се изчислява.

Видове (методи) адресации в ЕМК НС11:

- ☐ Непосредствена;
- ☐ Директна;
- ☐ Пълна;
- ☐ Индексна;
- ☐ Относителна;
- ☐ Вътрешна.

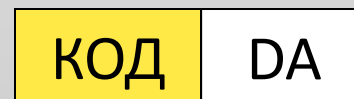
Методи за адресация (типове адресация) в HC11

❑ Непосредствена (Immediate)



Съдържа **директно след КОД аргументите за съответната операция**, които ще се извършва (C^{tn} -константа). В зависимост от дължината на ползваните регистри и наличието на PRE – обща дължина от 2, 3 или 4 байта.

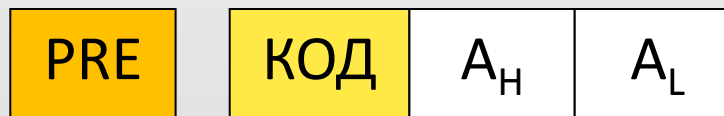
❑ Директна (Direct)



При тази адресация A_L се съдържа в байта след КОД (DA), A_H се предполага \$00. Адреси в диапазона \$00–\$FF са директни. 2-байтова инструкция. Изисква по-малко време за изпълнение. Обикновено тази 256 байта област се резервира за често ползвани данни (от вътрешни регистри, RAM).

Методи за адресация (типове адресация) в HC11

□ Пълна (Extended)



Съдържа директно след КОД адресите на операндите, които ще се обработват. В зависимост от дължината на ползваните регистри и наличието на PRE – обща дължина от 3 или 4 байта.

□ Индексна (Indexed)

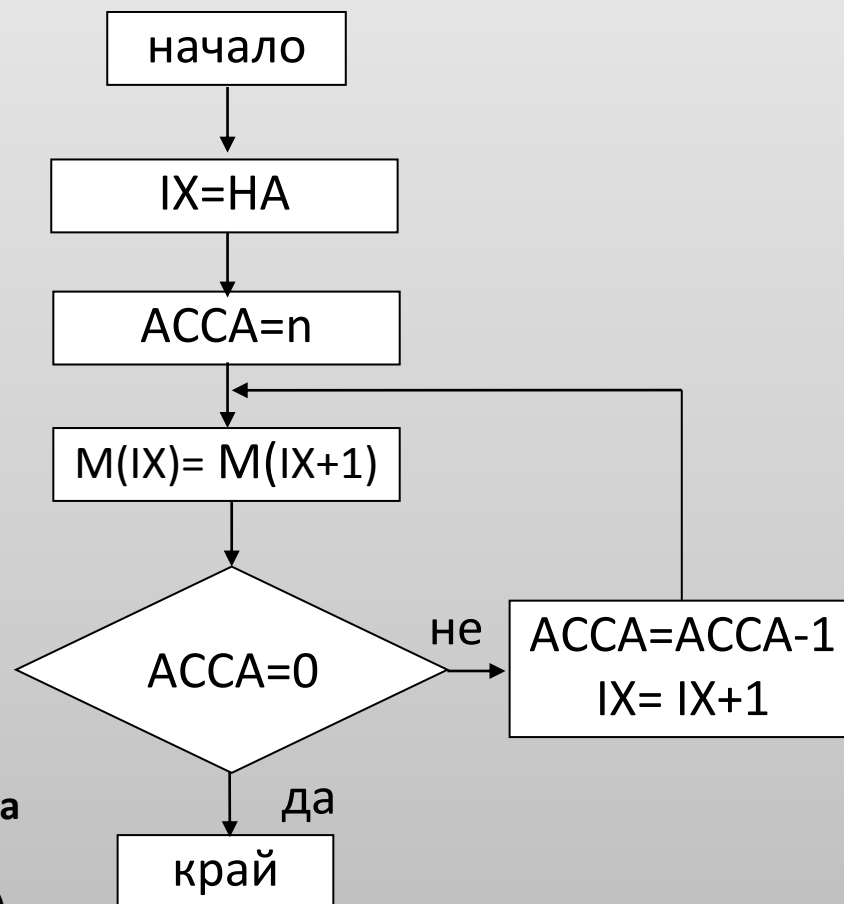


В този адресен режим байтът след КОД съдържа 8-битово отместване (**D– индексно отместване**), което се добавя към съдържанието на индексния регистър (**IX** или **IY**). Резултатът формира ефективния адрес (**EA**). Този режим позволява адресиране на всяка клетка от адресното пространство. Общата дължина 2 или 3 байта.

Методи за адресация (типове адресация) в HC11

□ **Индексна адресация (продължение)** – адресиране на клетки от масив:

$$EA=IX+D; D=00\div FF$$



Методи за адресация (типове адресация) в HC11

☐ **Относителна (Relative)**

КОД	$\pm B$
-----	---------

$\pm B$ – относително отместване /число със знак в ДК/.

Използва се CAMO за преходи под условие!

При разклонение на програми!

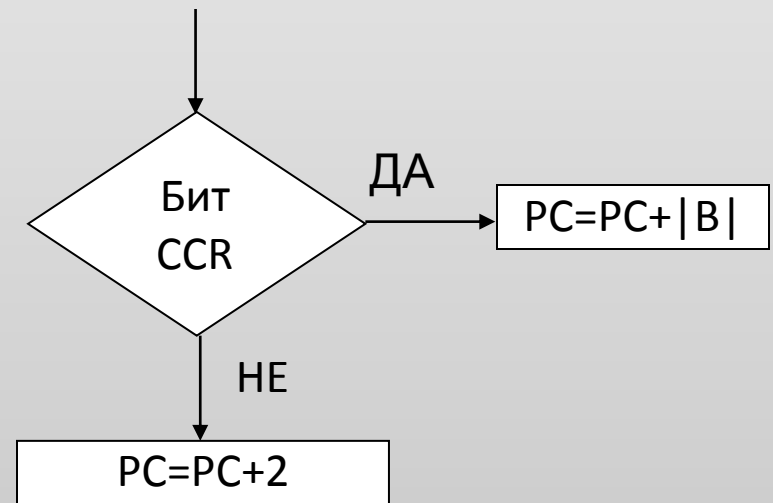
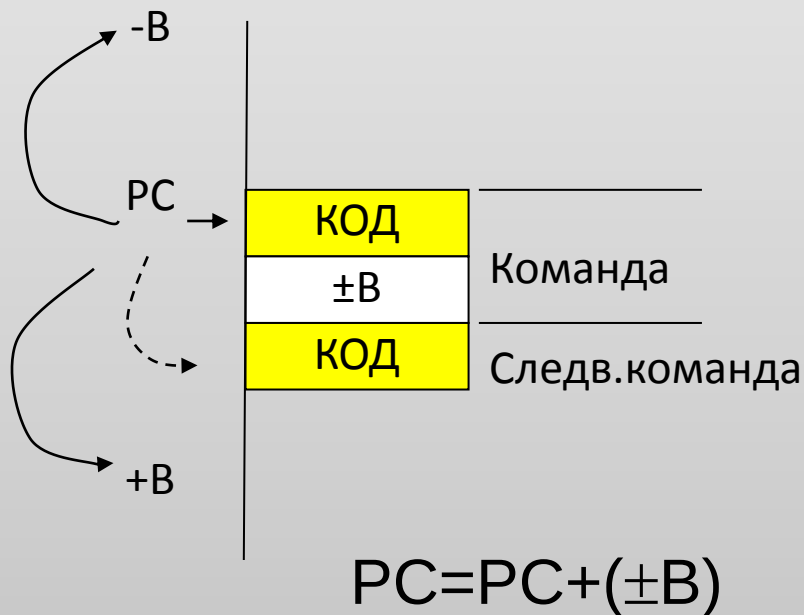
2-байтова инструкция.

Ако условието е “True”, относителното отместване B от инструкцията се добавя към съдържанието на PC за формиране на EA.

В противен случай - при “False” - продължава се към следващата по ред инструкция.

Методи за адресация (типове адресация) в HC11

❑ Относителна адресация (продължение)



B – относително отместване.

$-128 < B < +127$ (256 bt).

При по-голямо $|B| \rightarrow$ команда за безусловен преход.

Методи за адресация (типове адресация) в HC11

❑ **Вътрешна (Inherent)**

КОД

В този адресен режим цялата информация за изпълнение на инструкцията се съдържа в КОД.

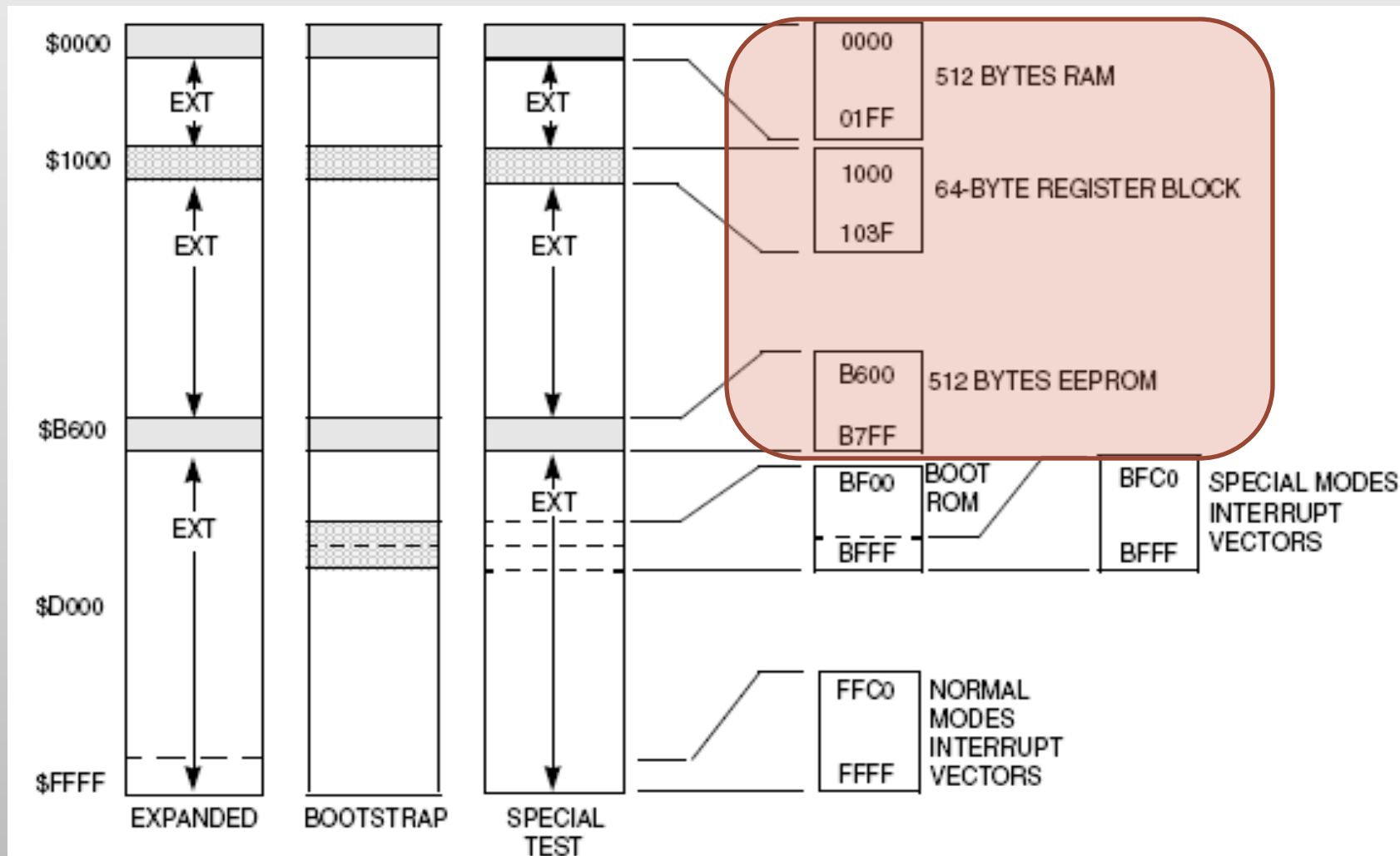
Безоператорна инструкция.

Използва се за работа с вътрешните регистри (акумулатори, индексни регистри, контролни инструкции без аргументи).

1 или 2-байтова инструкция.

ОРГАНИЗАЦИЯ НА ПАМЕТА В ЕМК фамилия HC11E1

Карти на паметта на **ЕМК HC11E1** в различните режими на работа



ОРГАНИЗАЦИЯ НА ПАМЕТА В ЕМК НС11Е

Карти на паметта (разпределение на адресното пространство). Различни за 3-те фамилии схеми от серията Е на ЕМК НС11 (еднакви за режими ЕМК и МП, различни за режими Bootstrap, Test);

❑ **RAM** (\$0000-\$01FF, 512bt) – инициализира се от адрес \$0000 след Reset.

Възможна промяна – от регистър **INIT** (всяка 4К област \$x000);

❑ **регистров блок** (\$1000-\$103F, 64bt) – установява се от адрес \$1000 след Reset,

Възможна промяна – от регистър **INIT** (всяка 4К област \$x000);

Приоритет: **Регистри → RAM → ROM.**

При презастъпване – спазване на приоритета!

ОРГАНИЗАЦИЯ НА ПАМЕТА В ЕМК HC11E

- RAM (SRAM) - за съхранение на инструкции, адреси/данни, променливи, междинни данни.

Може да ползва директна адресация (пести ресурси)!

Съхранение съдържанието на RAM – методи:

1. Stop mode (SW) – E-clock се спира, V_{DD} е включено.
 $P \approx F_{\text{раб}} \text{ (CMOS)} \downarrow$
2. MODB/ V_{STBY} – батерийно захранване (Reset=low при ниско V_{DD}).

ОРГАНИЗАЦИЯ НА ПАМЕТА В ЕМК НС11Е

- Програма за начално установяване (bootloader) – във вътрешна “**bootstrap ROM**” (\$BF00–\$BFFF) – само в режим Bootstrap.
- ROM/ EPROM/OTPROM:
 - >> достъпни в режими ЕМК и Bootstrap (ROMON от CONFIG – без значение).
 - >> в режим МП (ако са налични) – достъпни след Reset–в горните адреси на паметта (ROMON=1).
- **EEPROM:**
 - >> при обем 512 bt. EEPROM →\$B600–\$B7FF) – **НЕ може да се мести** (само за серия Е1)!
 - >> EEPROM се програмира/изтрива SW (след разрешение на т.нар. charge помпа).

Системни регистри – предназначение, видове

CONFIG – системен конфигурационен регистър

Състои се от байт от EEPROM паметта и тригери, конфигуриращи стартовата конфигурация на ЕМК.

Информацията от EEPROM се прехвърля в тригери по време на Reset и тяхното съдържание конфигурира работата на ЕМК.

В нормален режим на работа, промените в CONFIG до следващия Reset не засягат функционалността, дефинирана по-рано.

Регистърът се програмира оптимално в режим Bootstrap (MODA,MODB=0).

Address: \$103F							
	Bit 7	6	5	4	3	2	Bit 0
Read:					NOSEC	NOCOP	ROMON
Write:							EEON
Resets:							
Single chip:	0	0	0	0	U	U	1
Bootstrap:	0	0	0	0	U	U(L)	U
Expanded:	0	0	0	0	1	U	U
Test:	0	0	0	0	1	U(L)	U

Системни регистри – предназначение, видове

NOSEC —бит за забрана на Security

Невалиден, освен при зададена при производство Security маска.

0 -Security разрешена; 1 - Security забранена (липса на маска).

NOCOP — бит за забрана на COP системата

Разрешава/забранява COP системата.

1 - COP забранена; 0-COP разрешена.

ROMON — бит за разрешаване на ROM/EPROM/OTPROM

0 - ROM / EPROM забранена и това адресно пространство може да се адресира като външно;

1 - ROM / EPROM е наличен в адресното пространство.

EEON (EEPROM Enable Bit) бит за разрешаване на EEPROM

0 - EEPROM е изключен от адресното пространство и това адресно пространство може да се адресира като външно;

1 - EEPROM е наличен в адресното пространство.

Системни регистри – предназначение, видове

OPTION регистър

Със специално предназначение за конфигуриране на системата. Установява допълнително вътрешната конфигурация при инициализация. Контролните битове IRQE, DLY, CR[1:0] се записват веднъж и след това от тях само се чете.

Address: \$1039								
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ADPU	CSEL	IRQE ⁽¹⁾	DLY ⁽¹⁾	CME		CR1 ⁽¹⁾	CR0 ⁽¹⁾
Write:								
Reset:	0	0	0	1	0	0	0	0

ADPU – бит за включване на системата за АЦП.

Системни регистри – предназначение, видове

OPTION регистър (продължение)

CSEL (Clock Select Bit) – алтернативен clock източник за EEPROM зарядната помпа (както и за АЦП);

IRQE - дефинира IRQ за задействане само по фронт;

DLY - Enable Oscillator Startup Delay Bit

0 – не включва допълнително закъснение след излизане от режим STOP (само 4 цикъла) ;

1 – закъснение от 4000 цикъла на ТГ (E-clock) се включват при излизане на ЕМК от режим STOP. Стабилизира се кварцовия осцилатор (препоръчва се).

CME - бит за включване на системата Clock Monitor (след Reset-изключена);

Bit 2 - не се ползва (винаги се чете “0”);

CR[1:0] - битове за делене честотата на ТГ за COP системата след предварителното ѝ делене на 2^{15} (prescaler битове).

ЕМК фамилия HC11 - EPROM/OTPROM

Основни характеристики:

- **Обем:** HC711E9 (12K), HC711E20 (20K), HC711E32 (32K);

В изтрито състояние са програмирани "1"!

- Изтриване (при моделите с прозорче) с UV източник за 15-45 мин.
- **Програмиране** – програмират се "0"; Изисква се 12V външен източник към извод X1RQ/V_{PP}. Процедурата по програмиране на EPROM/OTPROM се извършва през регистъра **PPROG** (за повечето схеми от фамилията служи и за програмиране на EEPROM).
- Методи за програмиране и проверка:
 1. Програмиране на конкретен адрес;
 2. Програмиране с предварително заредени данни.

ЕМК фамилия HC11 - EPROM/OTPROM

1. Програмиране на конкретен адрес (валиден при всеки режим);

- Бит ROMON в регистъра CONFIG → "1"
- На извод XIRQ/V_{PPE} → 12V; • Извод IRQ → "1".

Регистър PPROG

Address: \$103B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ODD	EVEN	ELAT ⁽¹⁾	BYTE	ROW	ERASE	EELAT	EPGM
Write:								
Reset:	0	0	0	0	0	0	0	0

LDAB #\$20

STAB \$103B

STAA \$0,X

LDAB #\$21

STAB \$103B

JSR DLYEP

CLR \$103B

; Бит **ELAT=1** от **PPROG** за разрешаване на тригерите
; Записва данни в адрес от EPROM

; Бит **EPGM=1** от **PPROG** за разрешаване
; напрежението за програмиране на EPROM

; Закъснение 2-4ms (процедура) – за записа

; Изключва напр. за програмиране → режим четене

ЕМК фамилия HC11 - EPROM/OTPROM

2. Програмиране с предварително заредени данни

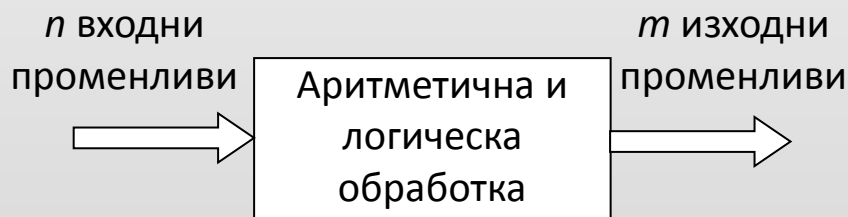
- Програмира се софтуерно в рамките на режими TEST или Bootstrap;
- >> чрез потребителски софтуер (зарежда се през серийния асинхронен интерфейс SCI) ;
- >> с разположена в ROM обслужваща процедура (от стартов адрес \$BF00).

Индексни регистри IX и IY се инициализират, след което получават данни отвън. Начална стойност по подразбиране на първи адрес за запис в EPROM - \$D000 (за промяна – стойност на IY указателя);

- 12V напрежение на входа за програмиране $\overline{XIRQ}/V_{PPE}$.

Плъзгане на програма /"забиване" на SW/

1. SW грешки – Beta версии



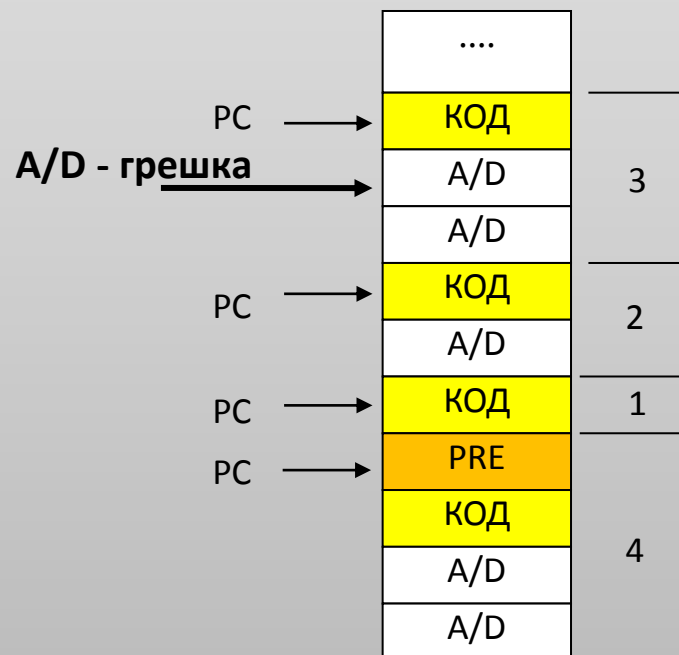
Много комбинации → възможни софтуерни грешки.

Невъзможност на програмиста да тества физически всички възможни комбинации.

Бета-версии – постепенно се отстраняват забелязани от клиентите грешки.

2. HW плъзгане

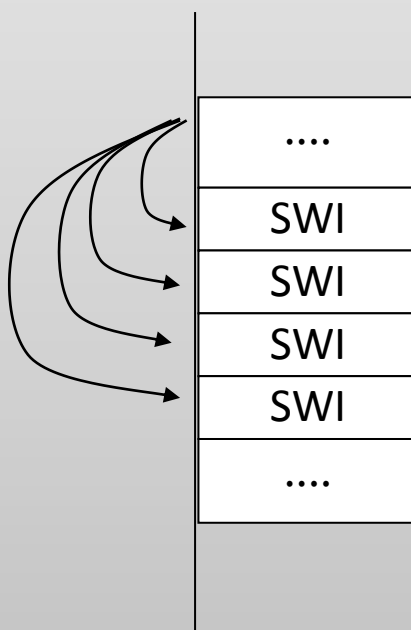
При попадане на Адреси или Данни (A/D) вместо на КОД / PREBYTE



Плъзгане на програма /”забиване” на SW/

HW плъзгане (A/D вместо КОД) - Методи за борба

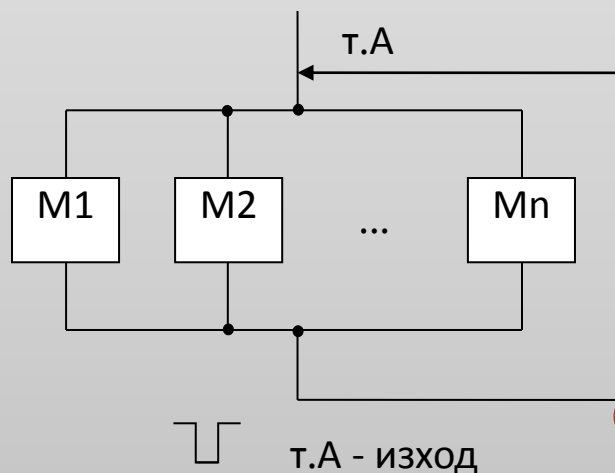
- SW “капани” (изкуствено влагане на 4 команди SWI –“капани”):



Плъзгане на програма /"забиване" на SW/

HW плъзгане (A/D вместо КОД) - Методи за борба:

- **COP система.** NOCOP бит (CONFIG) – вкл./изкл. COP система;
- **Смяна статута на COP системата** – запис в CONFIG, след това Reset;
- **COPRST – регистър** Arm/Reset COP Timer Circuitry (\$103A) – за активиране механизма на COP системата (\$55) или принудително нулиране на таймера (\$AA).



t_{min}

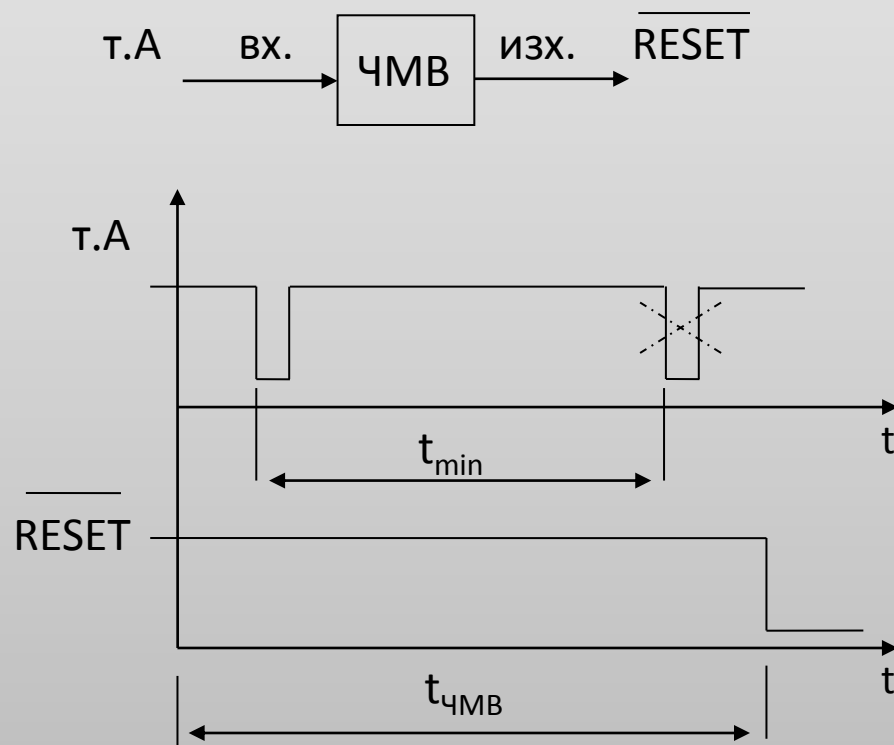
Address	\$103A							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Write:	0	0	0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

CR[1:0]	Divide E/2 ¹⁵ By	XTAL = 4.0 MHz Timeout – 0 ms, + 32.8 ms	XTAL = 8.0 MHz Timeout – 0 ms, + 16.4 ms	XTAL = 12.0 MHz Timeout – 0 ms, + 10.9 ms	XTAL = 16.0 MHz Timeout – 0 ms, + 8.2 ms
0 0	1	32.768 ms	16.384 ms	10.923 ms	8.19 ms
0 1	4	131.072 ms	65.536 ms	43.691 ms	32.8 ms
1 0	16	524.28 ms	262.14 ms	174.76 ms	131 ms
1 1	64	2.098 s	1.049 s	699.05 ms	524 ms
E =		1.0 MHz	2.0 MHz	3.0 MHz	4.0 MHz

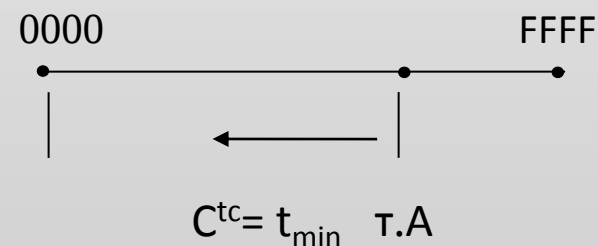
Плъзгане на програма /"забиване" на SWI

COP Watchdog контрол – аналогии:

- Чакащ Мултивибратор (ЧМВ);



- Таймер;



Ако стане 0000 $\rightarrow$ RESET

Бр (- , в режим на изваждане)

C^{tc} – дели чест. поредица