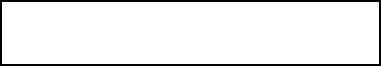
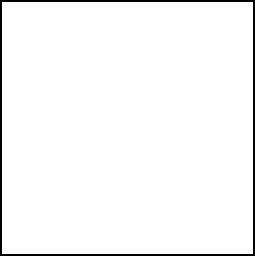
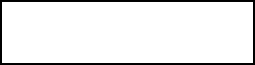
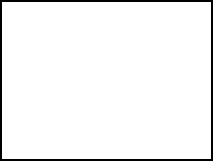
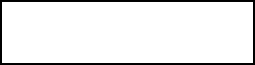
1. **ЕМК – структура, основни блокове, шинна организация.**

*Приложение:* за контрол и управление на различни обекти и процеси. Вид Embedded System (вградена МП система със спец. предназначение).

**Структура на микропроцесорно устройство**

*Класическа структура* (Чарлз Бабидж): включват следните устройства (компоненти):

* + Централен процесор (аритметично-логическо устройство, АЛУ, Arithmetic Logic Unit) с възможност за работа със специфична система от инструкции. Инструкциите се изпълняват за определено време в зависимост от тактовата честота и спецификата им;
  + RAM (Random Access Memory) памет - енергозависима;
  + Управляващо устройство (УУ), Control Unit;
  + Входно устройство (Input Unit);
  + Изходно устройство (Output Unit);



RAM

ВУ

УУ

АЛУ

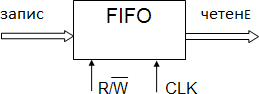
ИУ

*клавиатура*

* + Магистрална (Шинна) организация:
    - АМ (AB, AL, address bus) – адресна магистрала;
    - ДМ (DB, DL, data line – магистрала за данни;
    - УМ - управляваща магистрала
    - Буфериране на шините (bus buffers);
  + Блок за управление: управляващо устройство (control unit, CU), ROM за макрокода; стекова памет (STACK), указател на стека (SP, stack pointer), контролер на прекъсванията (IC, interrupt controller);
  + Блок за обработка на командите: програмен брояч / брояч на командите (PC, program counter), регистър на командния код / код на инструкцията (IR, instruction register), дешифратор на на кода на инструкцията (ID);
  + Блок за обработка на данните: АЛУ, акумулатор (ACC, accumulator); регистри с общо предназначение (RB, registers block), флагов регистър (CC, code condition)

1. **Памети с последователен достъп. Видове. Параметри.** Паметите биват още: адресни и безадресни.

## ***Безадресни*** *– конкретен механизъм (FIFO, LIFO) или признак (асоциативни)за* достъп без конкретен адрес;

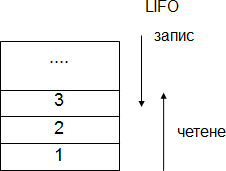
- FIFO (First-In First-Out) **–** първи “влязъл” първи излязъл. При първи тях се чете първия бит записана информация. *“Кюнец”*.

Приложение на FIFO паметите:

* При монитори – за четене на информация в процеса на визуализация;
* За обработка (трансфер) на “подредена” информация между несинхронизирани шини и устройства:

LIFO (Last-In First-Out) – последен влязъл първи излязъл. При тези памети първи се чете последния записан бит информация. *“Кофа”.*

Приложение: при Stack паметите (бърз достъп до данни).

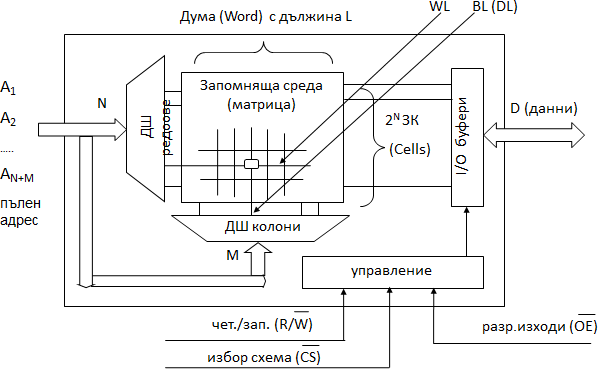


**3.Памети с непосредствен достъп SRAM – блокова схема , шини.** Видове:

* + RAM (random access memory)  енергозависими, за четене и за запис.
    - SRAM (static RAM); - биполярни;
      * CMOS;
      * BiCMOS
    - DRAM (dynamic RAM); - MOS
    - ROM (read-only memory)  енергонезависими, само за четене.

Режими на работа при RAM паметите:

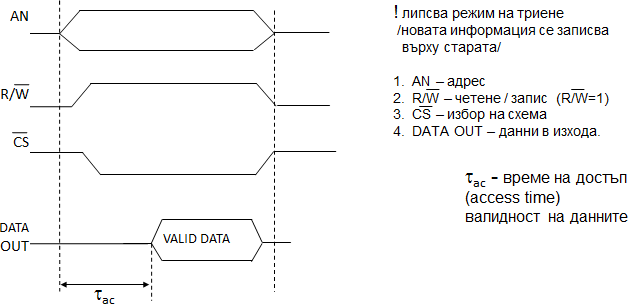
* + неизбрана;
  + четене;
  + запис;
  + standby (понижена консумация).



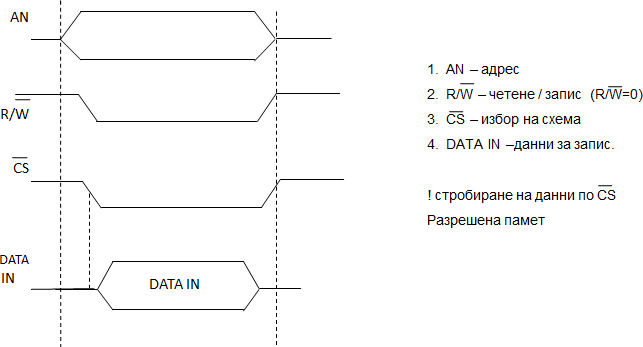
* Запомняща матрица – на база тригери (SRAM) или други специфични транзисторни структури (DRAM);
* Адресни шини: А0...АN+M – адресират 2N ЗК с дължина на думата L.
* RAM със сериен адрес (с допълнителен регистър само през 1 АШ).
* CS (chip select) – избор на схема (избрана при ниско ниво CS=0);
* R/W (read/write) – четене запис (при R/W=0 – запис), още – WE (write enable);

### Памети с непосредствен достъп SRAM – режими на работа.

* + **четене от паметта:**



* + **запис в паметта:**



1. **DRAM памети – блокова схема, шини, режими на работа.**



* + Запомняща среда (матрица от 3К)
  + Логически схеми за избор на адрес.
  + Усилватели – за сигнала от 3К
  + Логика за въвеждане на адресите
  + Схеми за четене/запис.
  + Вътрешни броячи за регенерация
  + Изходна логика

### Операции:

* + Четене( стандартно)
  + Четене(по страница– Fast Page Mode, FPM)
  + Запис
  + Четене-запис(Read-Modify-Write, IBM Corp.)

1. **DRAM памети – методи за регенерация. Режим DMA – предназначение.** Необходимост – *разреждане на зап.капацитет с времето* при необръщане към ЗК – загуба на данни.

### Методи за регенерация на DRAM:

* пакетна: спира се микропроцесора, регенерира се цялата памет;
* периодична: регенерира се 1 ред (дума) от паметта;
* принудителна – при липса на обръщение към паметта над минималното време за регенерация – преминава се към пакетна или периодична регенерация;
* скрита – в рамките на вътрешния цикъл на обработка на една команда (по време на изпълнението й в

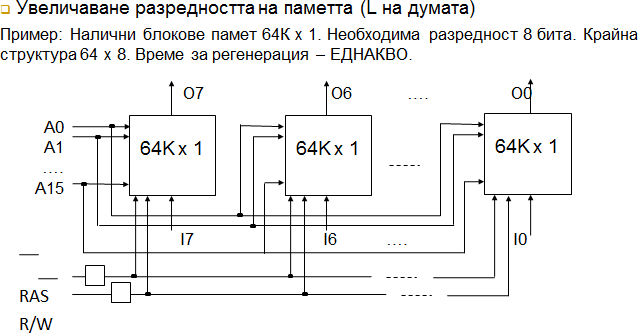
АЛУ на микропроцесора).

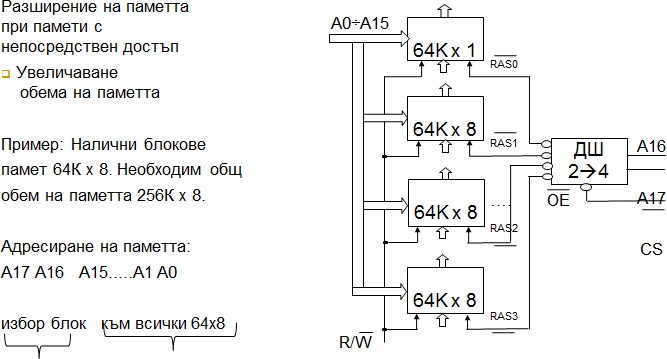
### DMA – direct memory access.

Същност – за достъп на устройството до паметта без да е необходима намесата на процесора. Осъществява обмен на блокове от паметта между две устройства.

Реализира се при персонални компютри: използва отделен **DMA** контролер или Bus mastering (PCI bus).

### Разширение на паметта при памети с непосредствен достъп (разширяване на дължината на думата, увеличаване на обема)





1. **ROM памети : основни типове и приложение.**

Общи характеристики:

* само за четене (MROM) или за четене и Програмиране по специален начин (PROM, EPROM, EEPROM, Flash);
* енергонезависими;
* с непосредствен достъп (адресни памети):
  + същата организация като RAM (pin to pin compatible);
  + еднакво време за достъп до всеки ЗЕ;
* програмирането – по специален начин при
* производството им или от потребителя.

Приложение:

1. За съхранение на програми за работа на микрокомпютъра и старт- програми в PC (BIOS, монитори и др.);
2. За съхранение на големи таблици от данни (ТИ, т.нар. look-up tables);
3. За реализация на логически функции (OR, NOR, AND, NAND) – ПЛМ (програмируеми логически матрици).

### PROM – структура, програмиране, особености. Основни характеристики:

* Еднократно програмируеми от потребителя (клиента) в лабораторни условия;
* Структура – матрична решетка (масив) от “бушони ” (array of fuses) – NiCr, poly-Si, W връзка с възможност за прегаряне – т.нар. Burning ROM;

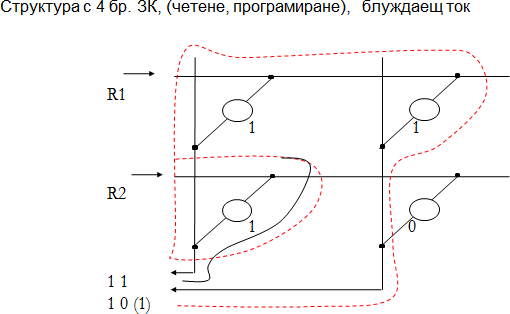
### Процес на програмиране:

* чрез устройство *– Програматор*.
* пропуска се ток с Висока стойност, при което се прегаря жичката между АШ и ШД (липса на връзка) или се оставя (наличие на връзка). Биполярни (TTL).

Запис – бавен (5 мин.). Високо бързодействие.

- програмиране – с ток (за разлика от EPROM);

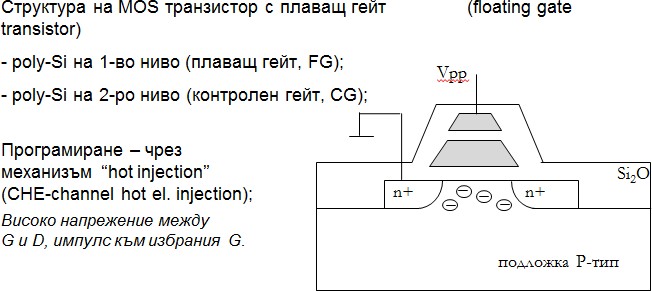
* еднократно програмиране;
* биполярни PROM – защитени от радиоактивно въздействие;
* високо бързодействие (до 1 ns);
* висока консумация.



### Памети ЕPROM – блокова схема, режими на работа.

* + вид ROM, енергонезависима памет (non-volatile memory);
  + представлява масив от MOS транзистори с плаващ гейт (floating gate transistors) – т.нар. UVEPROM;
  + програмиране с по-високо от захр.напрежение Vpp = 12,7 v;
  + изтриване – чрез облъчване с ултравиолетова UV светлина с определена дължина на вълната (w=253 nm), за определено време

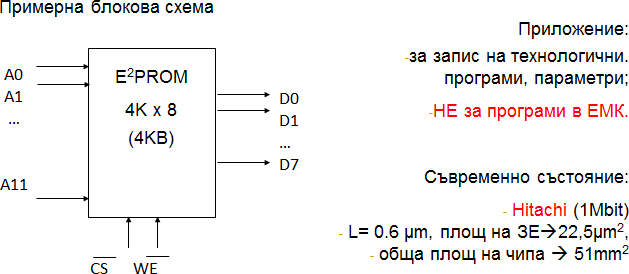
-



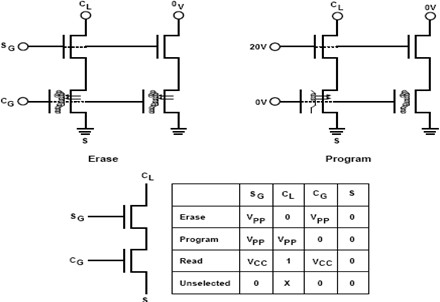
Режими на работа:

* нормални: неизбрана, четене, standby
  + изтриване (UV светлина); *N.B. Първоначално изтриване - около 20 min*
  + програмиране (запис).
  + запис (програмиране) – Отн. бавен х 50 ms, Адресира се цял байт;

### 11.Памети ЕЕPROM – блокова схема, режими на работа.



**Режими на работа:**



1. **FLASH памети специфика на ЗК. Типове.** Особености:
   * блоково програмиране/изтриване и индивидуален запис до конкретна ЗК;
   * изтриване – чрез F-N тунелиране;
   * програмиране – чрез F-N тунелиране или CHE механизъм.

Опасност при Flash ROM – евентуална промяна в дадена зона (portion disturbance) докато се записва в друга. Липсва селектиращ транзистор (СИ )

Предимство – напълно CMOS технология с допълнителни процеси за FG (до и под 0.18µm) – постигане на висока СИ!

Реализация на структурно ниво:

* Boot block (Sector erased) Flash - изтриване на сектори от 4КB до 128KB (16KB boot block – сигурност!);
* Bulk erased Flash - изтрива се целия Flash.

Операциите четене и запис – на принципа на непосредствен побайтов достъп.

Съвременни FLASH архитектури – имат включен КА за автоматизация на WRITE и ERASE операциите.

Първите FLASH-софтуерно управление на операциите.

**XTAL, EXTAL** - осигуряват интерфейс с външен ТГ (тактов генератор): кварцов(XTAL) или CMOS-съвместим(EXTAL) за тактуване на вътрешните схмеи. Честотата на външния генератор е 4 пъти по-висока от вътрешната за ЕМК. При използване на EXTAL , XTAL следва да се оставя свободен.

**RESET -** двупосочна шина. Вход - за инициализация на МП в начално състояние. Изход - с ОД за индикация на вътрешна грешка при Clock monitor или COP.

Поради невъзможност да се изпълнят правилно инструкции при понижаване на захранването се изисква включване на допълнителна LVI (low-voltage-inhibit) схема за предпазване на EEPROM.

* 1. **lock (E)** - изход от вътрешния ТГ. Честотата в изход Е е 1/4 от тази на шини XTAL/EXTAL. При E=low се извършва обработка във вътрешните структури на ЕМК, при E=high - постъпване на данни. В режим STOP E-Clock се спира. За намаляване на радиоемисиите изход E се забранява.

**IRQ (Заявка за прекъсване, InterruptRequest)** - вход за асинхронна заявка за прекъсване към ЕМК. Може да се окаже активиране по заден фронт или по ниво.

**(Немаскирано прекъсване, Non-Maskable Interrupt) -** вход за немаскирано прекъсване след Reset инициализация. При Reset , бита X в CCR(condition-code-register) се установява в 1 и всяко прекъсване се маскира, докато софтуерно не се разреши.

**MODA,MODB (MODA/LIR, MODB/Vstby) -** при Reset,MODA и MODB определят 1 от 4-те режима на работа: Single-chip(EMK), Expanded(МП), TEST, BOOT;

LIR(load-instruction-register) - след избор на режима осигурява изход ОД като индикация, че изпълнението на инструкцията е започнало; Vstby - използва се за свързване на захранване за RAM в Standby режим

**STRA/AS -** Strobe A (STRA), Address strobe(AS) - в зависимост от режима на работа: В ЕМК(Single-chip) режим, STRA извършва функцията "input handshake" (strobe input); В МП(Expanded)режим, AS осигурява строб за адреса (adress strobe).

STRB/R/W - Strobe B и Read/Write - извършват или строб по отношение на изхода (output strobe) или индицират посоката на предаване на данни в ШД в зависимост от режима.

**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;
* инструкции TAP, TPA прехвърлят данни от акумулатор А към регистъра на състоянието и обратно;
* Инструкция DAA (Decimal Adjust accumulator A) – след BCD аритметични операции;
* Еднопосочност на операциите събиране, изваждане и сравнение.

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

Допуска изпълнение на операции INX, DEX, CPX.

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

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

**PC** (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):

**CCR (регистър на състоянието).** НЕ съдържа данни.Съдържа 8 бита (индикатори, флагове), 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).

### режимите на работа на НС11.

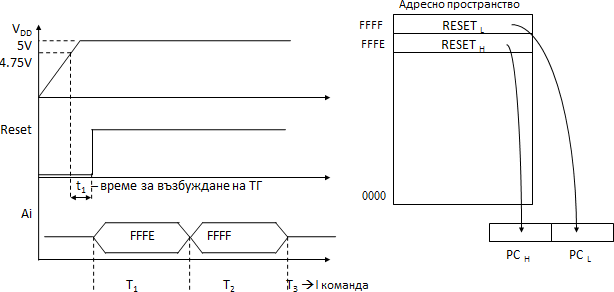
* + карти на паметта – различни за 3-те фамилии схеми от серията Е на ЕМК НС11 (еднакви за режими ЕМК и МП);
  + различни карти (разпределение на адресното пространство за отделните режими на работа (Bootstrap, Test);
  + RAM ($0000-$01FF, 512 bt) - от адрес $0000 след Reset, възможна промяна – от регистър INIT (всяка 4K област $x000);
  + регистров блок ($1000-$103F, 64 bt) – от адрес $1000 след Reset, възможна промяна – от регистър INIT (всяка 4K област $x000);
  + Приоритет: Регистри RAM  ROM. Презастъпване – спазване на приоритета!
* 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-clocks се спира, VDD e включено. PFраб (CMOS) 
    2. MODB/VSTBY – батерийно захранване (Reset=low при ниско VDD).
* Програма за начално установяване (bootloader) – във вътрешна “bootstrap ROM” ($BF00–$BFFF) – само в режим Bootstrap.
* В режим МП - ROM/EPROM/OTPROM (ако са налични) – достъпни след Reset – в горните адреси на паметта

(ROMON=1, CONFIG).

* ROM/ EPROM - налични в режими ЕМК и Bootstrap (ROMON – без значение).
* При EEPROM=512 bt. (EEPROM $B600–$B7FF) – НЕ може да се мести (само при Е1)!
* EEPROM се програмира/изтрива SW (charge помпа – разрешена).



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

КОД (КОП) – указан в карти, разпределени в 4 страници.

Допълнителен байт - PRE BYTE за увеличаване броя команди (препраща ЦП към съответната страница). Предшества КОД в общия формат. Пълна инструкция – дължина от 1 до 4 байта.

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 адресации в ЕМК НС11:

* + Непосредствена;
  + Директна;
  + Пълна;
  + Индексна;
  + Относителна;
  + Вътрешна.

### COP Watchdog система в НС11 – предназначение.

* + Ползва входен сигнал от Е-clock, разделен 15 пъти (E ÷ 215);
  + COP изработва изходен сигнал за генериране на ниско ниво към RESET извода за рестартиране на ЕМК и външните устройства *автомaтично след изтичане на зададения таймаут период*;
  + Таймаут периодът може да се настройва с помощта на prescaler битове CR[1:0].
* Включва логика за дефиниране на съответните I/O изводи РА0÷РА7 от Порт А като такива за таймерни функции (IC/OC) или за работа с общо предназначение;
  + Изводи PA3, PA2, PA1, PA0 – IC входове или с общо предназначение. Логика за прихващане (детекция) на входния сигнал по фронт (положителен или отрицателен – по избор);

## *Нивата им могат да се четат по всяко време от регистъра PORTA* безусловно.

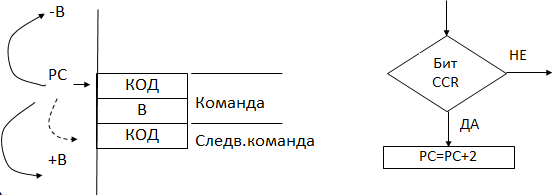
* + Изводи PA6, PA5, PA4, PA3 – ОC изходи или с общо предназначение; *При използване като ОС изводи – не може да се записва в тях*; Функциите на OC[5:2] – свързани с един от изводите от Порт А, само OC1

– допълнителна функция за контрол посредством комбинация от стойности на изводи PA[7:3];

o Извод PA7: с общо предназначение, като ОС1 или като вход за пулс- акумулатора.

* **Индексна адресация** (продължение) – адресиране на клетки от масив
* **Относителна** (Relative) В – относително отместване.

Използва се САМО за преходи под условие. При разклонение на програми! 2-байтова инструкция.

Ако условието e “True”, относителното отместване В от инструкцията се добавя към съдържанието на PC за формиране на ЕА. В противен случай при “False” се продължава към следващата поред инструкция

* **Вътрешна** (Inherent)

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

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

1 или 2-байтова инструкция.

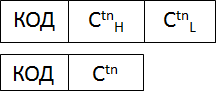
Extended - Съдържа дирекотно след КОД адресите на операндите , които ще се обработват. В зависимост от дължината на ползваните регистри и наличието на PRE - обща дължина от 3 или 4 байта.



Direct - При тази адресация AL съдържа в байта след КОД(DA), AH се предполага $00. Адреси в диапазона $00 - $FF са директни (2 байта инструкция). По-малко време за изпълнение. Обикновено тази 256 байта област се резервира за често ползвани данни (от вътрешни регистри, RAM, външна памет)



Immediate - Съдържа директно след КОД аргументите за съответната операция, която ще се извършва. В зависимост от дължината на ползваните регистри и наличието на PRE - обща дължина от 2, 3 или 4 байта.



Indexed - В този адресен режим байтът след КОД съдържа 8-битовото 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 клета от адр. пространство. Обща дължина - 2 байта.





B - относително отместване

Relative - използва се САМО за преходи под условие. При разклонение на програми!

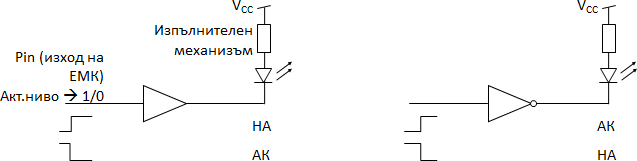
2-байтова инструкция.

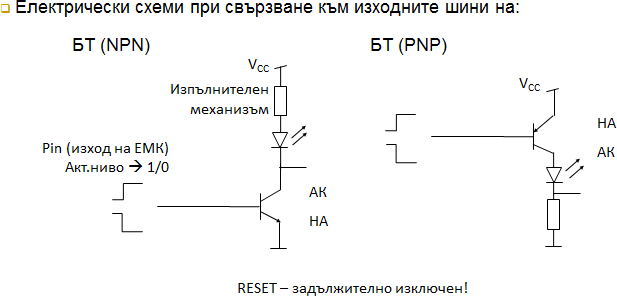
Ако условието е "True", относителното отместване B от инструкцията се добавя към съдържанието на PC за формиране на EA. В противен случай при "False" се продължава към следващата поред инструкция

Inherent - цялата информация за изпълнение се съдържа в КОД. Използва се за работа с вътрешните регистри (акумулатори, индексни регистри, контролни инструкции без аргументи).

1 или 2 байтова инструкция.

* Активно изходно ниво (low/high);
* RESET – неактивно състояние за ЕМК;
* Електрически схеми при свързване към изходните шини на: ЛС – повторител ЛС – инвертор





Системата за АЦ преобразуване (A/D система) използва:

* ЦАП с последователна апроксимация и редистрибуция на заряда за преобразуване на аналоговия сигнал в цифров;
* 8-канална система (съответно 8-битов мултиплексор), 8-битова точност на преобразуване;
* Не изисква външна S/H (sample and hold) верига поради ползване техника на разпределение на заряд;
* АЦП може да синхронизиран от системния (E clock) или от вътрешен RC осцилатор.

### Състав:

* мултиплексор;
* ЦАП;
* верига за цифров контрол;

структура за запомняне резултата от преобразуването

* АЦП: Преобразуване - извършва се за сигнала от съответния аналогов вход, избран от мултиплексора (MUX).

Съдържа: (1) капацитивен масив (DAC-digital-to-analog capacitor), (2) компаратор, (3) регистър за последователна апроксимация (SAR, successive approximation register).

* Принцип на извършване на преобразуването – последователност от 8 операции сравнение, започващо от бита MSB. Всяко сравнение определя стойността на съответния бит в регистъра SAR.

DAC масива извършва 2 функции: *действа като S/H верига, осигурява напрежението за сравнение по време на апроксимацията*.

Резултата от сравнението се съхранява в SAR и след приключване процеса на АЦП се прехвърля в друг регистър.

Т.нар. “charge” помпа осигурява напрежение (7-8V, за време поне100μs) на превключване към гейтовете на аналоговите ключове в MUX. Помпата се разрешава от бит ADPU на регистъра OPTION.

* **Синхронизация** – бит CSEL (clock select) от регистъра OPTION определя дали АЦП използва системната (Е) тактова поредица или вътрешния RC тактов генератор (задължително при честота на E<750KHz, но чувствителен на промяна на честотата - шум).

### Последователност на преобразуване

Interrupts - зарежда PC с конкретен вектор, сочещ към определен адрес от който да се заредят съответните инструкции за изпълнение.

**Временно се спира работата на основната програма до приключване изпълнение на обработката на прекъсването**(сервизна процедура). След това основната програма продължава да се изпълнява.

# **-** SWI (SoftWare Interrupt) - софтуерно (програмно) прекъсване:

* инструкция SWI не може да бъде прекъсната докато приключи изпълнението й
* НЕ МОЖЕ ДА СЕ ЗАБРАНИ глобално от маска в контролния регистър CCR
* SWI установява маска I в CCR в "1" - след това други прекъсвания не се допускат до края на обработката или до нулиране на маската I софтуерно
* HWI (HardWave Interrupt) - хардуерно (апаратно) прекъсване - IRQ, XIRQ
* Немаскирано прекъсване (X - вход) - немаскираните прекъсвания водят ВИНАГИ до прекъсване работата на ЦП. Прилагат се при възникнали сериозни проблеми - напр. програмни забиване, отпадане на захранването. IRQ

XIRQ - с най-висок приоритет /без Reset/ (по-висок от всички маскирани - с маска)

* Неправилен КОД (Ilegal Opcode Trap) - при среща на несъществуващ КОД (PRE+КОД)
* Прекъсване през вход IRQ:
  + допълнително маскирано прекъсване през външен вход за ЕМК
  + задействане - по ниско ниво на сигнала от източника на прекъсване
  + свързване на няколко източника в "жично-ИЛИ"

**Приоритет на прекъсванията:**

# Дефинира се хардуерно

* Определя кой тип RESET/прекъсване следва да се обработи първо при постъпили повече от една заявки за прекъсване
* На всяко маскирано прекъсване може да бъде зададен приоритет спрямо другите маскирани прекъсвания

**НЕМАСКИРАНИ**(not maskable) прекъсвания - източници, приоритет:

1. POR, външен Reset - HW
2. Reset от Clock monitor (CM) системата - HW
3. Reset от COP WD Reset системата - HW
4. От вход XIRQ - HW
5. Неправилен (несъществуващ) КОД - HW
6. Софтуерно прекъсване (SWI) - SW

**МАСКИРАНИ**(maskable) прекъсвания - източници, приоритет:

1. Вход XIRQ - HW 2.Прекъсване в реално време 3-4-5. -Таймер IC1-2-3 - HW

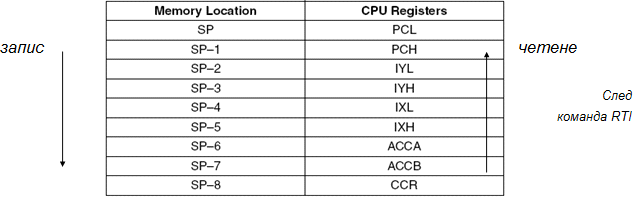
6-7-8-9. - Таймер OC1-2-3-4 - HW

1. Таймер IC4/OC5 - HW
2. Таймер - препълване - HW
3. Пулс акумулатор - препълване - HW
4. Пулс акумулатор - входен фронт - HW
5. SPI (асинхоронен сериен периферен интерфейс) - завършил трансфер - HW
6. SCI (синхронен сериен комуникационен интерфейс) - HW

### 30.Алгоритъм за обработка на прекъсване.

1. Приключва се изпълнението на текущата инструкция - време: зависи от броя на такта , необходими за изпълнението й.

1. Текущото състояние на вътрешните регистри се записва в стека:



1. Флагът за прекъсванe I ( и X при прекъсване от XIRQ) в CCR се установява в "1" и забранява последващи прекъсвания.
2. Зарежда се съответният на прекъсването вектор и изпълнението продължава от адреса, намиращ се в този вектор
3. След приключване изпълнението на обслужващата прекъсването програма, която трябва да завършва с RTI
4. Съдържанието на вътрешните регистри се възстановява от стека в обратен ред (LIFO памет)
5. Продължава изпълнението на основната програма.

### 31.Организация на стека- тип и организация.

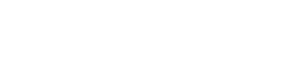
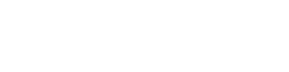
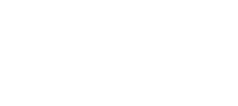
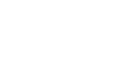
68HC11 съхранява байтове в определена част от паметта. Той ползва специални регистри, наречени *stack pointer* or *SP, който следят мястото на стека в паметта.*

*Когато число е сложено в стека се нарича stack push, числото се запазва в паметта на сегашния адрес на SP. След това* stack pointer минава към следващата позиция в паметта.

Когато число се взима от стека се нарича *stack pull,* stack pointer се връща назад към последната локация и тогава числото на тази част от паметта се взема.

### 32.Прекъсване тип IRQ – предназначение и възприемане.

* Допълнително маскирано прекъсванепрез външен вход за ЕМК;
* Задействане – по ниско ниво на сигналаот източника на прекъсване;
* Свързване на няколко източника всхема “жично ИЛИ”;
* Флагът I от контролния регистър CCR се установява в “1” (заявка за прекъсване) и може да се нулира SW след обслужване на прекъсването.



VCC

R

вход IRQ

Източник 1

на ЕМК

Източник N

* + Немаскирано прекъсване (XIRQ - вход)

Немаскираните прекъсвания водят ВИНАГИ до прекъсване работата на ЦП. Прилагат се при възникнали сериозни проблеми – напр. програмно забиване, отпадане на захранването.

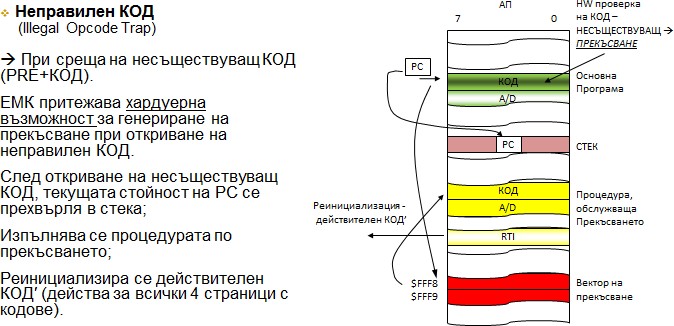
XIRQ (в предишни версии - NMI, non-maskable interrupt );

След Reset - битове X, I от регистъра CCR се установяват в “1” и забраняват всички маскирани прекъсвания и XIRQ. След инициализация, бит Х може да се нулира SW (инструкция TAP) и да се разреши входа XIRQ. След това Х НЕ МОЖЕ да се установи в “1” SW – т.е. XIRQ е НЕМАСКИРАНО ПРЕКЪСВАНЕ!

XIRQ – с най-висок приоритет /без Reset/ (по-висок от всички маскирани – с маска I)

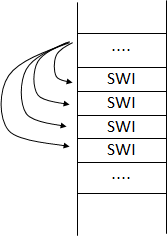
*При маскирано с I-бит прекъсване, битът I се установява автоматично хардуерно след изпълнение на прекъсването и прехвърляне съдържанието на CCR от стека. При това Х не се засяга!*

*При прекъсване с X-бит се засягат X и I битове – те се установяват хардуерно след прочитане на CCR от стека. Инструкцията RTI възстановява битовете X и I в предишните им стойности.*



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.

1. HW плъзгане: Адреси или Данни вместо КОД HW плъзгане (А/D вместо КОД) - Методи за борба
   * + SW “капани” (изкуствено влагане на 4 команди SWI – “капани”);
     + COP система. NOCOP бит (CONFIG) – вкл./изкл. СОР система;
     + Смяна статута на СОР системата – запис в CONFIG, след това Reset;
     + COPRST – за защита механизма на нулиране СОР системата ($55) или принудително нулиране на таймера ($АА);
     + Чакащ Мултивибратор (ЧМВ);



**Енергоспестяващи режими –** работата на ЦП се прекратява до постъпване на Reset или друго прекъсване!

* + **Wait режим –** спира обработката на данни и намалява консумацията на междинно ниво (до около 3-4 пъти);
  + **Stop режим –** спира всички тактови източници и намалява консумираната мощност на възможно най-ниско ниво (съдържанието на RAM се съхранява).

### Поставяне на ЕМК в режим Wait

* чрез команда (КОД) – WAI
* системните регистри се записват в стека;
* спира се работата на ЦП до пристигане на Reset или друго прекъсване (външно IRQ, XIRQ или вътрешно генерирано – от таймер-системата, SCI, SPI);
* кварцовият осцилатор остава включен.

Редукция на консумираната мощност – зависи от това колко периферни функции (поддържани от тактова поредица) могат да бъдат изключени.

!!! Шината за А/D се ЧЕТЕ НЕПРЕКЪСНАТО от адреса на стека, където e записан временно (при прекъсването) стойността на CCR регистъра.

### Изход от режим Wait - при постъпване на прекъсване, което не е било маскирано.

Изключване на таймер-системата – при бит I=“1” и забранена COP-система чрез NOCOP=“1” (регистър CONFIG).

**Консумацията на АЦП** – системата не се влияе от режима Wait. Ограничаване консумацията на АЦП – чрез поставяне бит ADPU=“0”.

**SPI** – разрешава се/забранява се – чрез бит SPE от съответния контролен регистър.

**SCI предавател** - разрешава се/забранява се – посредством бит TE; **SCI приемник** - разрешава се/забранява се – посредством бит RE.

Консумацията в режим Wait – **ЗАВИСИ ОТ КОНКРЕТНОТО ПРИЛОЖЕНИЕ**!

### Поставяне на ЕМК в режим Stop

* чрез команда – **STOP при бит** S=“0” от CCR;
* При S  0  Stop КОД се третира като безоператорен (NOP);
* Режим Stop – осигурява възможно най-ниска консумация (всички тактови източници, вкл.кварцовия генератор са спрени.

**Излизане от режим Stop** – подаване на ниско ниво на един от входове: (IRQ, XIRQ) или RESET. Опция – по фронт на сигнала IRQ.

* Вътрешни периферни функции – СПРЕНИ (изключени тактови източници);
* Данните в RAM - съхранени (VDD e включено);
* ЦП зпазва състоянието си, нивата на I/O шини – непроменени;

### При връщане към нормална работа (restart на системата) - ЦП възстановява обработката от позицията преди поставяне в режим Stop.

**Връщане към нормална работа (рестарт):**

* чрез Reset – извършва се нормалната последователност (всички I/O шини и функции – към техните начални състояния);
* през вход IRQ – трябва бит I=“0” от CCR (IRQ – вход за немаскирано прекъсване). Стойността на бит X е без значение, като:

 при X=“0” (немаскиран) – ЦП изпълнява последователност, съответстваща на заявка за прекъсване от вход XIRQ;

 при X=“1” (маскиран/забранен) – ЦП продължава изпълнението на следващата инструкция след Stop.

Необходимо e ВЪЗСТАНОВЯВАНЕ на кварцовия тактов генератор (спрян в този режим) при рестарт!

* При ползване на вътрешния генератор – влагане на изкуствено закъснение – контролен бит DLY=“1” (по подразбиране при Reset);
* При ползване стабилизиран външен генератор – DLY=“0” софтуерно при инициализация. В този случай изход от режим Stop НЕ ТРЯБВА да се прави с Reset (по подразбиране DLY=“1”).
* Отделна система в ЕМК 68НС11 (към порт А)
* Включва 5 отделни вериги за делене на честотата
* Предварителен делител на честотата от кварцовия осцилатор на 4
* Основна таймерна верига - 16-битов брояч с програмируем коефициент на броене
* Всички операции в таймерната система са съотнесени спрямо честотата на основния (таймерен) брояч
  + начало на броене $0000 (след излизане от Reset)
  + край на броене $FFFF. Флагът на препълване O(overflow) в контролния регистър CCR се вдига в "1".

След това - броене отново то $0000.

* При нормален режим на работна на ЕМК - НЕ Е ВЪЗМОЖНО СПИРАНЕ, нулиране или промяна състояните на брояча.



Записва(регистрира) момента на настъпване на външни събития от PA2, PA1, PA0

* по фронт. Запомня се стойността на таймерния брояч в момента на събитието. Регистри:
* PACTL ( пулс-акумулатор контролен регистър)
* TCTL2 (таймер-контрол регистър2) - за конфигуриране IC функцията по вид на фронта за всеки извод по отделно (2 бита за извод)
* TMSK1 (таймер масков регистър1)
* TFLG1 (таймер-флагов регистър1)
* TICx (таймерни регистри) - входни регистри за съхранение 16-битовото показание на брояча в съответния момент. НЕ ПРОМЕНЯТ СЪДЪРЖАНИЕТО СИ при Reset
* TI4/O5 - ползва се едновременно за IC или OC регистър към извод PA3

Програмира действие което да се изпълни в дефиниран момент време (достигане определено състояние на таймерния брояч).

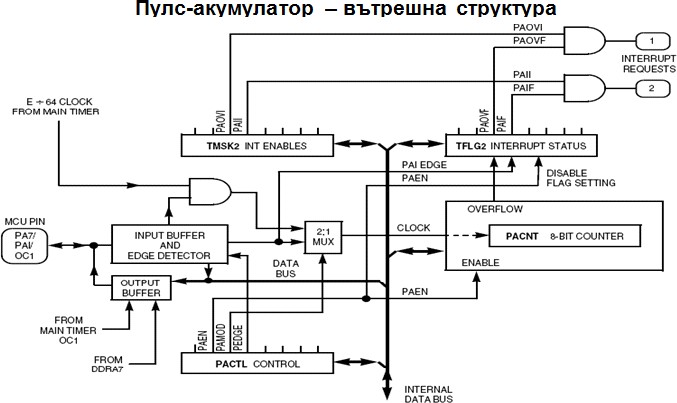
* отделни 16-битови регистри и 16-битови компаратори за всеки от 5-те OC изхода.

Регистри:

* TOCx - 16-битови (2 x 8-битови) изходни регистри (за четене и запис).
* CFORC - регистър за директно предизвикване на сравнение. Позволява сравнение м/у регистрите OCx и текущото показание на таймерния брояч.
* OC1M - OC масков регистър. Обвързва състоянието в изводите от Порт А PA[7:3] в зависимост от успешното сравнение (резултат в OC1) чрез залагане на маски за всеки от тях.
* OC1D - OC регистър за данни. Позволява съхраняване на данните в съответните изводи от Порт А от OC1 след успешно сравнение.
* TCNT - Таймерен броячен регистър - 16-битов в режим сумиране (от него само се чете), съдържа текущото състояние на таймерния брояч
* TCTL1 - Таймерен контролен регистър1 . Определя действието в резултат на на успешното OCx сравнение.
* TMSK1 - Таймерен масков регистър за прекъсване 1 - 8-битов, разрешава/забранява прекъсванията по отношение на функциите IC/OC.
* TFLG1 - позволява таймера флагов регистър 1
* TMSK2 - Таймерен масков регистър за прекъсване 2 - 8-битов, разрешава/забранява прекъсвания при препълване и в реално време (RTI)
* TFLG2 - Таймерен флагов регистър за прекъсване 2 - показва дали съответни събития в таймерната система за настъпили. Позволява таймера да работи в регистриращ режим или в режим с прекъсване.

Отделна подсистема, включваща 8-битов брояч за работа в два режима:

* Обикновен броячен режим - увеличава се стойността на брояча при постъпване на сигнал на външния му извод
* Разрешителен акумулиращ режим - в този случай E-clock/64 тактува 8- битовия брояч, но само докато външния извод PAI е активиран Четене/запис от Пулс-акумулаторът - по всяко време.



### честотата на обмен.

SCI - Сериен Комуникационен Интерфейс:

* универсален UART-тип интерфейс
* отделна подсистема в ЕМК 68НС11 Е серия
* стандартен NRZ (non-return to zero) формат: 1 старт-бит, 8/9 бита данни, 1 стоп-бит
* независими приемник и предавател, но една и съща скорост на предаване и общ формат на данните
* поддържа различни скорости на предаване

Формат на данните:

* свободна(Idle) линия преди старт на комуникацията в "1"
* старт-бит "0" - начало на цикъл комуникацията
* данни - предавани с LSB пръв (дължина 8 или 9 бита)
* стоп-бит "1" - край на цикъл комуникацията Честота на обмен:

Serial Peripheral Interface - Асинхронен сериен периферен интерфейс

* Всяко устройство има свой ТГ (без обща синхронизация). Асинхронните устройства имат изводи TxD, RxD
* Обмен на данни при сериен интерфейс м/у различни устройства: Използват се по 1 до 4 кабелни линии, бит по бит, различна скорост на предаване. Предаване (Transmitting, шина TxD). Приемане (Receiving, шина RxD)
* Използва дуплекс връзка за обмен на данни (duplex, full-duplex): 2 линии, двупосочен обмен. едновременно във времето.
* Предаване на данните бит по бит (1 бит на такт)
* Може да комуникира на големи разстояния
* Изисква минимален брой линии: за предаване, за приемане и маса
* Възможност за софтуерно дефиниране формата на предаване на данните, прекъсванията и др.

**-** синхронен двупосочен интерфейс от типа Multi-master.

* За връзка м/у различни ИС.
* Активни линии: SDA (Serial Data) SCL (Serial Clock) - двупосочни
* Всяко устройство може да работи и като предавател и като приемник
* Приложение - разширяване възможностите на ЕМК - връзки с други серийни устройства.

### 46.Интерфейс IEA232 (RS232) – тип, шини, кодиране, предимства, недостатъци.

* интерфейс за сериен обмен на цифрови данни между 2 устройства на Electronic Industries Association (EIA). Създадена първоначално за връзка между компютри през телефонни модеми;
* позволява дефиниране на до 20 сигнала. Достатъчни са само три - ТxD (send data), RxD (receive data), маса;
  + двуполярно предаване (2 нива) – от 5 дo 25V, с противопо-ложна полярност спрямо маса;
  + прилага се за индустриален обмен с формат: 1 старт-бит, 7(8) бита данни, опция-бит за контрол по четност, 1(2)стоп-бита;
* скорост на предаване (не фиксирана, зависи от устройствата) Ах150 (А=0÷7):

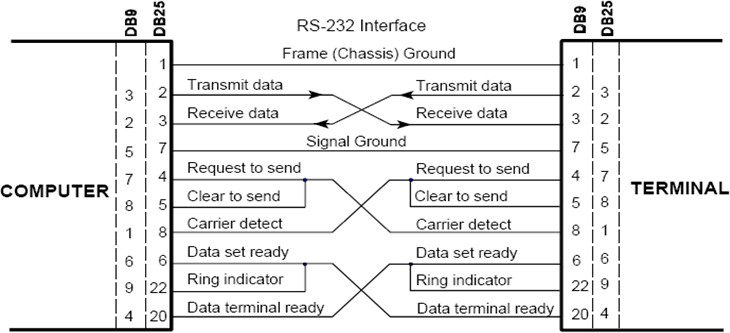
- 150;

- 300;

- 600 ,...., 19,200.

* + специфично дефинирани (при необходимост) скорости под 150 baud;
  + конектор DB-25: 25 извода (21 използваеми);
  + понастоящем – конектор DB-9 (СОМ) в компютрите.

Необходимо  2-те устройства да са свързани с конектори от един и съш вид и да ползват еднаква скорост на предаване.



### 47.Интерфейс RS485 – особености, предназначение.

Осигурява директна връзка м/у DTE устройства (без модеми), връзка м/у DTE в мрежа, комуникация на по-големи разстояния, обмен на данни с по-висока скорост. Отличен при връзка м/у много устройства

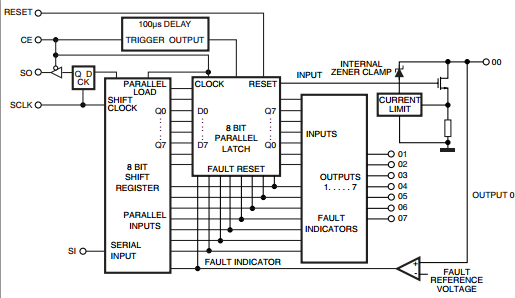
* Липса на обща маса и необходимост от презапасяване за ибягване на шима по линията.
* Сиганлите по RS485 са "плаващи" - всеки сигнал се предава спрямо линиите Sig+ и Sig- ,съответно
* Приемната част на RS485 сравнява разликата в напрежението м/у линиите, вместо абсолютната стойност на сигнала.
* Най-добро подтискане на шума
* диференциален тип връзка
* използва предаване тип "half-duplex"
* многовъзлов
* висока скорост на предаване
* висока чувствителност на приемната част
* специфициран интерфейс по отношение на Електрически, Механични и Протоколни изисквания. Отнася се за Host, Hub или Function.
* 7 адресни линии (до 127 устройства свързани заедно, топология звезда)
* диференциален сериен интерфейс
* 4 шини, 2 от тях заданни по усукана двойка -I2C
* данни предавани в NRZI код - асинхронно или синхронно м/у свързаните устройства.
* максимална дължина на кабела - 4м.
* 3(4) скорости на предаване от 10kbps до 400Mbps

Slow-Speed режим - обмен с бавни устройства; Full-Speed; High-Speed - USB 2.0

; Super-Speed - USB 3.0 и за оптична комуникация.

Информацията се предава сериино на устройствата използващи Serial Peripheral

Interface (SPI) protocol.



### Основни характеристики:

* Осем ниски RDSon DMOS изхода (0.5 O @ IO = 1 A @ 25 °C VCC = 5 V ± 5 %)
* 8 bit serial input data (SPI)
* 8 bit serial diagnostic output за претоварване
* Chip enable select function.
* Multipower BCD technology
* Package Multiwatt 15 and PowerSO-20