

# МИКРОПРОЦЕССОРНА ТЕХНИКА

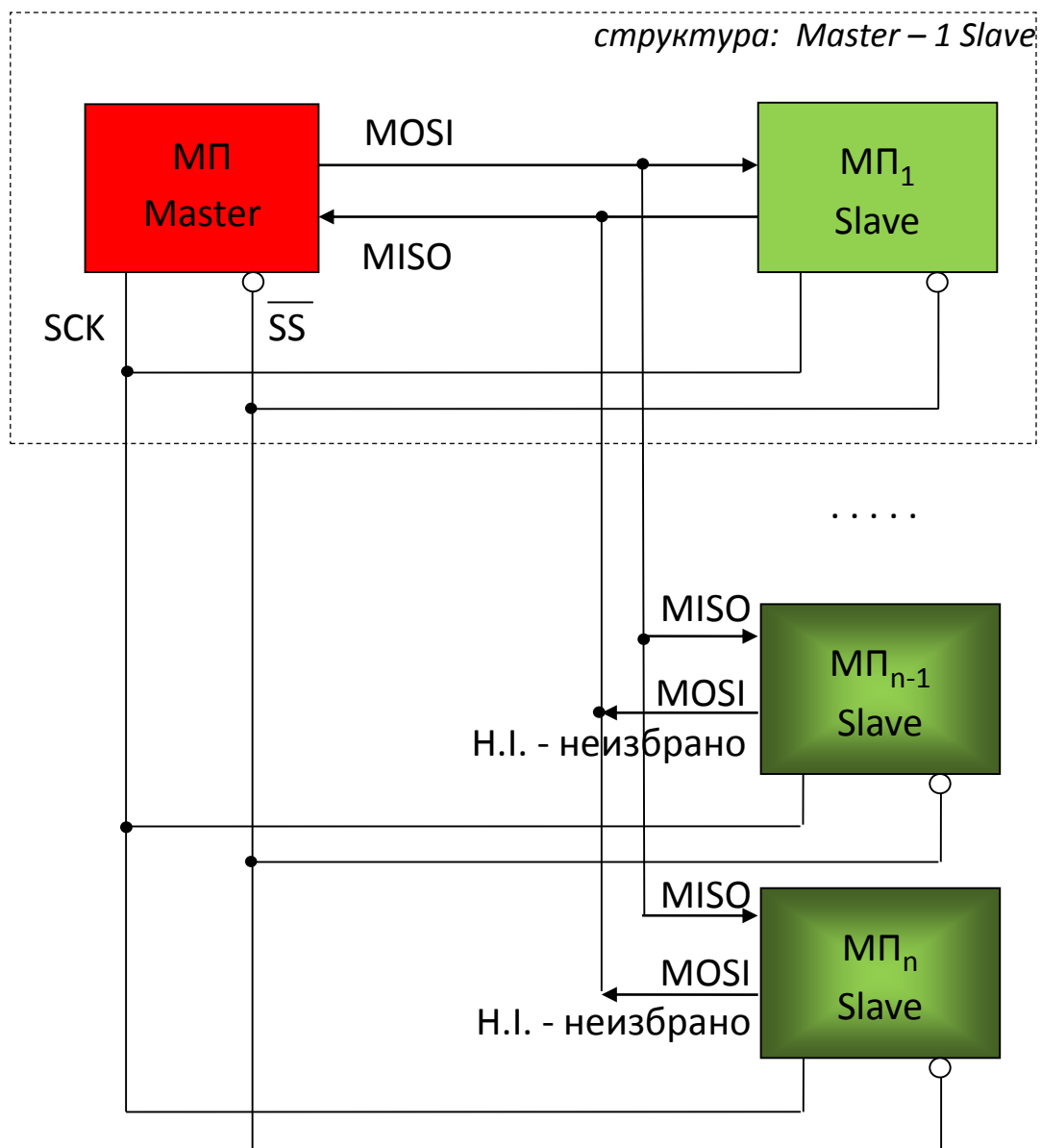
## ЛЕКЦИЯ #11

# Сериен синхронен интерфейс в HC11 – SPI

Независима комуникационна подсистема в 68HC11 за последователен (сериен) **СИНХРОНЕН** обмен на данни с **ВИСОКА СКОРОСТ** между ЕМК и външни устройства като:

- други микропроцесорни системи;
  - схеми за управление на LCD дисплеи;
  - специализирани памети (RAM, EEPROM);
  - драйверни схеми със сериен синхронен вход.
- ✓ за осъществяване на вътрешен обмен в т.нар. Multiple Master процесорни Embedded системи.
- SPI подсистемата – конфигурира се като **Master** или като **Slave**.
- При конфигуриране като **Master** – **скорост на обмен до  $\frac{1}{2}$  E-clock** честотата на ЕМК (напр. до 2Mbit/s при E-clock=4MHz);
  - При конфигуриране като **Slave** – **скорост на обмен до пълната E-clock честота** на ЕМК (до 4Mbit/s при E-clock=4MHz).

## Сериен синхронен интерфейс в HC11 – SPI (връзка между 2 и повече устройства)



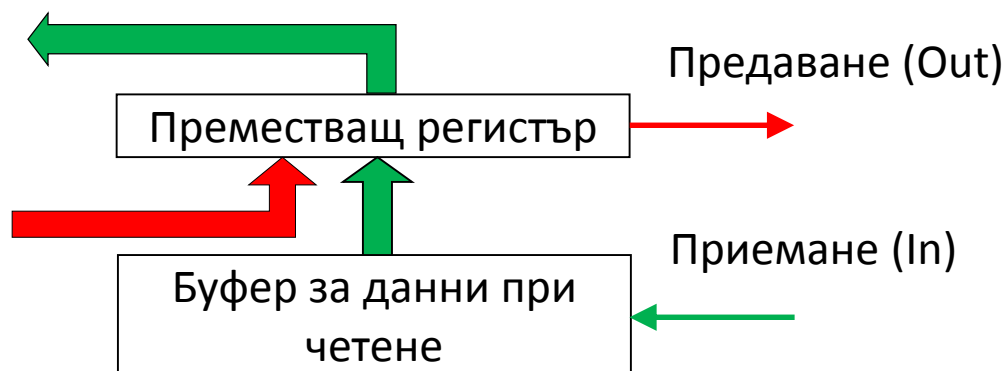
## Сериен синхронен интерфейс в HC11 – SPI

### Структура на SPI подсистемата:

- Преместващ регистър;
- Буфер за четене на данни.

Използват се следните принципи на буфериране при обмен:

- **единично буфериране при предаване** - записът на нова порция данни става едва след прочитане на предходната от преместващия регистър;
- **двойно буфериране при приемане** - данните при приемане се прехвърлят в отделен паралелен буфер за данни (**избягване на състояние презастъпване "overrun"**).

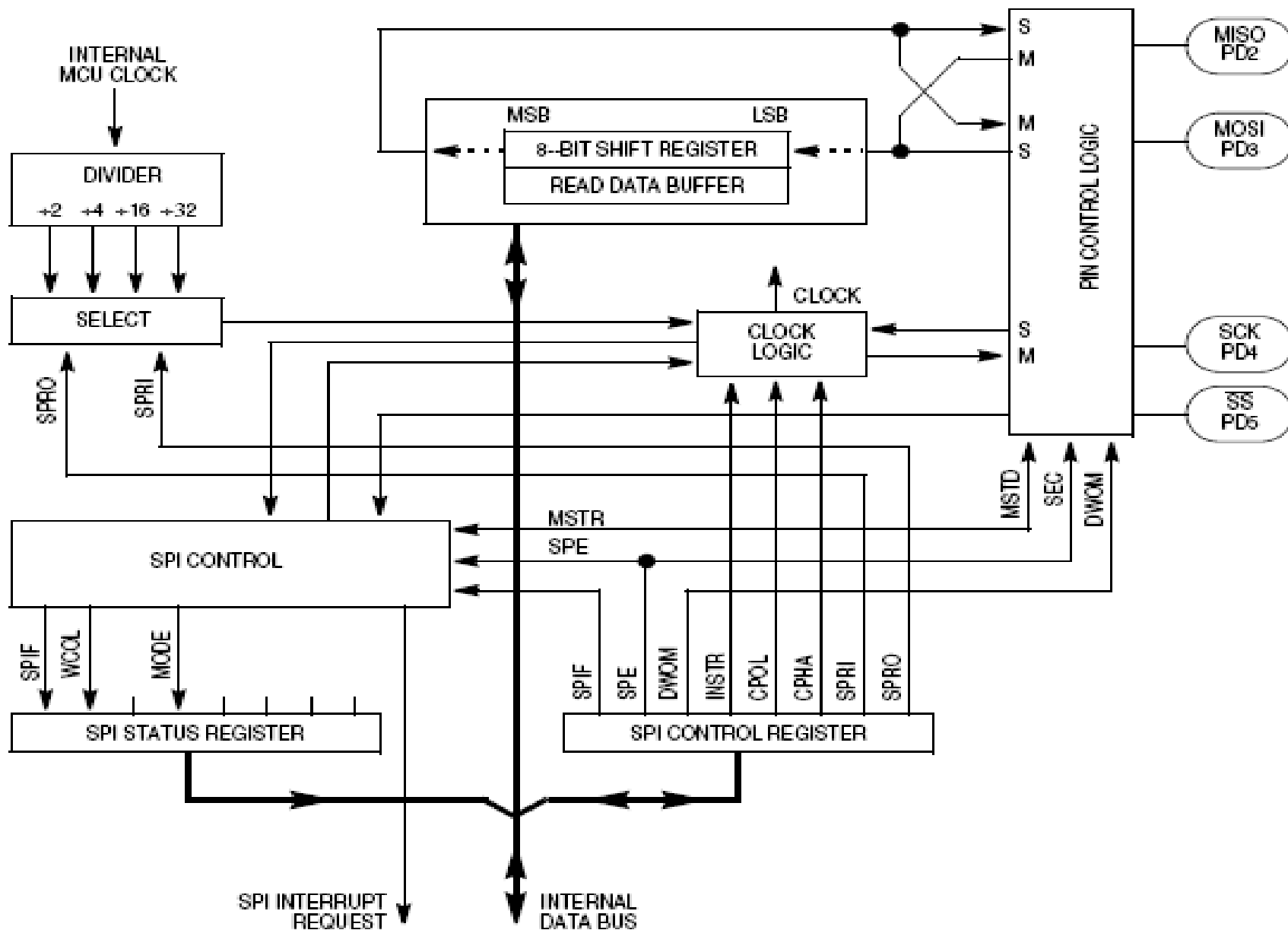


## Сериен синхронен интерфейс в HC11 – SPI

Структура на SPI подсистемата (регистри):

- буферен регистър и за данни **SPDR** - с ОБЩ АДРЕС както при четене, така и при запис от него;
  - статус регистър - **SPSR**;
  - контролен блок: контролен регистър - **SPCR**.
- 
- логика за управление/контролна логика (SPI control);
  - контрол състоянието на изходните шини (PIN control);
  - логика за задаване (избор) на тактовата честота (Clock logic) с включен делител (Divider).

## SPI подсистема – блокова структура



# Сериен синхронен интерфейс в HC11 – SPI

## Формати на обмен през SPI

- ✓ **Едновременно предаване/приемане** на данните през SPI.
- ✓ **Отделна тактова шина** - синхронизация на процесите на преместване и прехвърляне на данните от 2-та серийни канала за данни - Serial clock (SCK).
- ✓ **Допълнителна линия за избор на подчинено (Slave) SPI устройство** при конфигурация 1 Master и няколко Slave устройства - Slave Select (SS).

## Сериен синхронен интерфейс в HC11 – SPI (описание на шините)

- **MISO** (Master In/Slave Out) – входяща Master / изходяща Slave (PD2);
- **MOSI** (Master Out/Slave In) – изходяща Master / входяща Slave (PD3);
- **SCK** (Serial Clock) – сериен тактов сигнал (PD4);
- **SS** (Slave Select) – избор на Slave устройство (PD5).

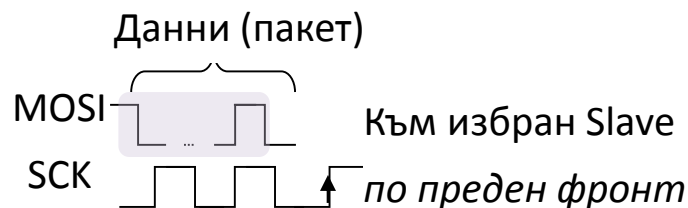
**SPI изходни шини** - всяка ИМА СЪОТВЕТЕН БИТ в регистъра DDRD за определяне посоката на предаване. Ако съответният бит в DDRD = "0", линията става с общо предназначение - изключва се от логиката на SPI.

**SPI входни шини** - всички за дефинирани като входове без значение от стойността на съответните битове в регистъра DDRD.

- **MISO** - ЕДНОПОСОЧНА линия за данни.
  - ВХОД към Master устройство;
  - ИЗХОД от Slave устройство.

При неизбрано Slave устройството – неговата MISO линия е изход в H.I.

- **MOSI** - ЕДНОПОСОЧНА линия за данни.
  - ИЗХОД от Master устройство;
  - ВХОД към Slave устройство.



**Master устройството разполага данните (пакета) на шината MOSI половин цикъл преди фронта на сигнала от Slave устройството по който то приема данните.**



## Сериен синхронен интерфейс в HC11 – SPI (описание на шините)

□ **SCK** – вход към Slave устройство.

- Генерира се от Master устройството;
- Синхронизира предаването на данни по шините MOSI и MISO;
- Master и Slave устройствата работят с обща синхронизация и обменят данни побайтово (в рамките на 8 последователни такта).

Възможни са 4 времеви съотношения – чрез контролни битове **CPOL**, **CPHA** от контролния регистър **SPCR**.

Скорост на обмен - битове **SPR[1:0]** от регистъра SPCR от Master устройството (стойностите на същите битове в регистъра на Slave устройството са *без значение*).

## Сериен синхронен интерфейс в HC11 – SPI (описание на шините)

### □ Slave Select ( $\overline{SS}$ ) – вход за избор на Slave устройство

$\overline{SS}$  на Slave устройството - трябва да е установен в ниско ниво преди обмена на данни и да остане в “0” през целия обмен.

$\overline{SS}$  на Master устройството трябва да е в “1”.

При ниско ниво → флагът MODF=“1” от статус регистъра SPSR (грешка на режима). Изход от това състояние – чрез запис на “1” в бит 5 от DDRD ( $\overline{SS}$  се дефинира като шина с общо предназначение, а не като вход на Slave устройството и се игнорира флагът).

**Битове CPHA в двете устройства – засягат действието на  $\overline{SS}$ . CPHA (трябва да са еднакви за Master и Slave устройствата):**

**CPHA=“0” –  $\overline{SS}$  следва да се установява в “1” между 2 последователни символа от съобщението;**

**CPHA=“1” –  $\overline{SS}$  може да се остави “0” между всеки 2 последователни символа.**

**При само 1 Slave устройство, неговото  $\overline{SS}$  може директно да се свърже към маса при CPHA=“1”.**

# Сериен синхронен интерфейс в HC11 – SPI

## Системни грешки в SPI интерфейса

Възможни са два вида грешки:

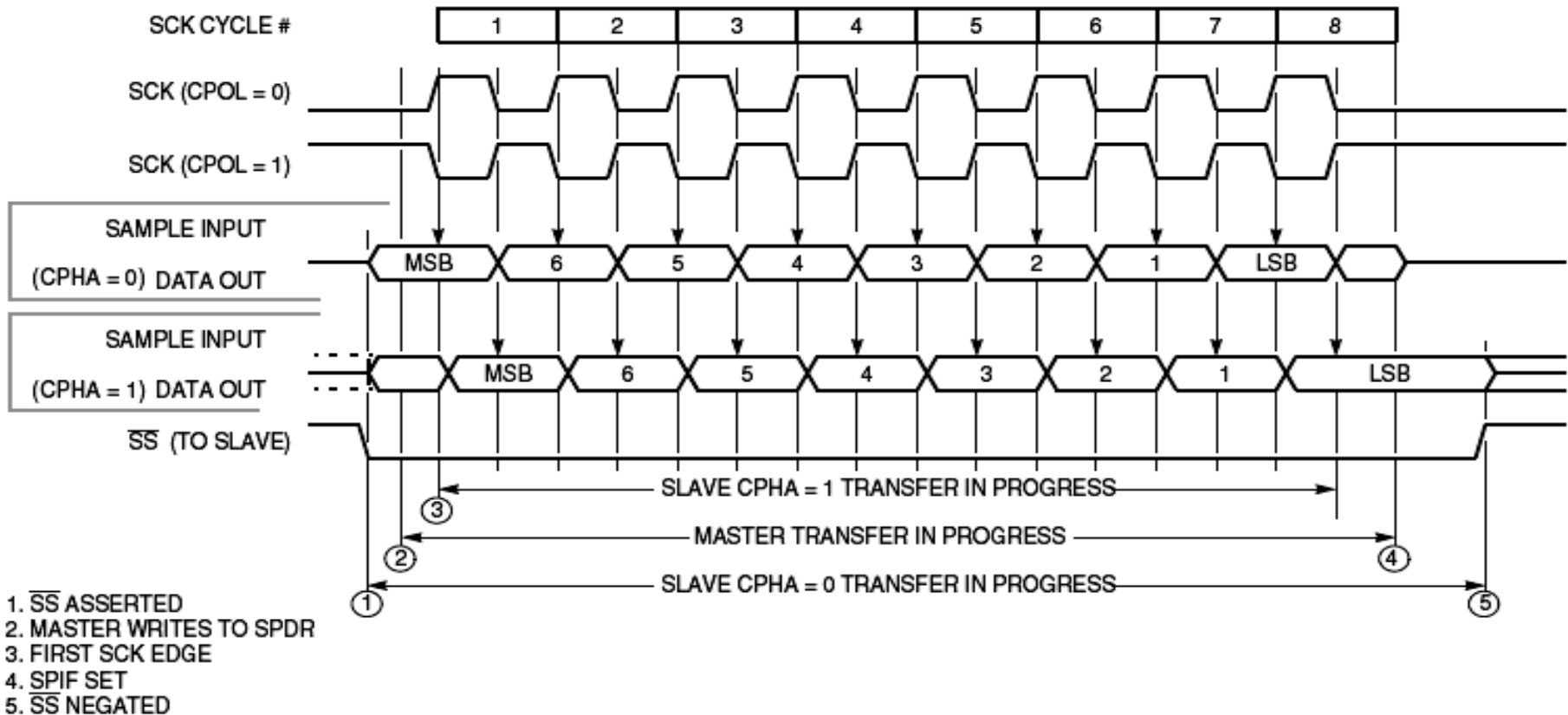
- при Multiple-master системи – ако **повече от 1 устройство се опитва да е Master** (mode fault);
- т.нар. “**write collision**” – при опит да се запишат данни в регистъра SPDR при все още незавършен обмен (предавателната част не е двойно буферирана – директно към преместващия регистър).

! При SPI конфигурирана като Master и  $\overline{SS} \rightarrow "0"$  (mode fault грешка) – повече от 1 Master устройство едновременно. При повече от 1 устройство като Master – възможност за повреда на драйверите на два извода (при CMOS драйвери: вероятна перманентна повреда).

**Механизъм за предпазване** - чрез забрана на драйверите: MSTR бита в контролния регистър SPCR и 4-те бита в DDRD се нулират и се генерира прекъсване (маскирано – чрез контролния бит SPIE и бита I от CCR).

## Сериен синхронен интерфейс в HC11 – SPI

### Формат на обмена по SPI интерфейса



Общо 4 комбинации, дефинирани от SPCR софтуерно по отношение на полярност (бит CPOL) и фаза на тактовия сигнал (бит CPHA).

- При CPHA="1"  $\overline{SS}$  към Slave устройствата остава в ниско ниво през целия обмен;
- При CPHA="0"  $\overline{SS}$  следва да се установява отново в "1" от нула при всеки нов предаван символ.

# Сериен синхронен интерфейс в HC11 – SPI

## Регистри в подсистемата SPI

- Регистър за данни – SPDR;
- Статус регистър – SPSR;
- Контролен регистър – SPCR.

### Контролен регистър – SPCR

#### **SPIE – бит за разрешаване на прекъсване**

Подава заявка за прекъсване при SPIE =“1” всеки път при SPIF=“1” или когато флагът MODF се установи в “1”. Прекъсванията се забраняват при SPIE =“0” или при маска I=1 от CCR.

#### **SPE – бит за разрешаване на SPI подсистемата**

При SPE=“0” – забранява цялата SPI подсистема. При SPE=“1” – разрешена и битове 2,3,4,5 от Порт D са шини на SPI интерфейса.

Ако е конфигурирана като Master и бит 5 от DDRD е “1”, то този бит става изход с общо предназначение вместо входа  $\overline{SS}$ .

Address:	\$1028							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
Write:								
Reset:	0	0	0	0	0	1	U	U

U = Unaffected

# Сериен синхронен интерфейс в HC11 – SPI

## Регистри в подсистемата SPI

### Контролен регистър – SPCR (продължение):

**DWOM (порт D Wired-Or Mode)**– бит за дефиниране изходите от порт D в режим “жично ИЛИ” (общо за всички изводи)

0 – “нормални” (противофазни) CMOS изходи;

1 – изходи с “отворен дрейн” (изисква резистор към захранване).

**MSTR – бит за конфигуриране на режим Master/Slave**

0 – режим Slave;

1 – режим Master;

**CPOL – бит за определяне на полярността на тактовия сигнал**

- При CPOL = “0” и липса на данни за трансфер, извод SCK на Master устройството има постоянно ниско ниво;

- При CPOL = “1” и липса на трансфер извод SCK е във високо ниво.

**CPHA – бит за определяне на фазата**

Заедно с бита CPOL определят формата на предаване на данните между Master и Slave устройствата. CPHA избира между 2-та протокола по отношение на SS.

# Сериен синхронен интерфейс в HC11 – SPI

## Регистри в подсистемата SPI

### Контролен регистър – SPCR (продължение):

**SPR[1:0]** - битове за определяне на честотата на тактовия сигнал SCK

Определят честотата на шината SCK на подсистемата SPI – валидни само при конфигурация като Master устройство.

! При конфигурация като Slave са без значение.

SPR[1:0]	Divide E Clock By	Frequency at E = 1 MHz (Baud)	Frequency at E = 2 MHz (Baud)	Frequency at E = 3 MHz (Baud)	Frequency at E = 4 MHz (Baud)
0 0	2	500 kHz	1.0 MHz	1.5 MHz	2 MHz
0 1	4	250 kHz	500 kHz	750 kHz	1 MHz
1 0	16	62.5 kHz	125 kHz	187.5 kHz	250 kHz
1 1	32	31.3 kHz	62.5 kHz	93.8 kHz	125 kHz

# Сериен синхронен интерфейс в HC11 – SPI

## Регистри в подсистемата SPI

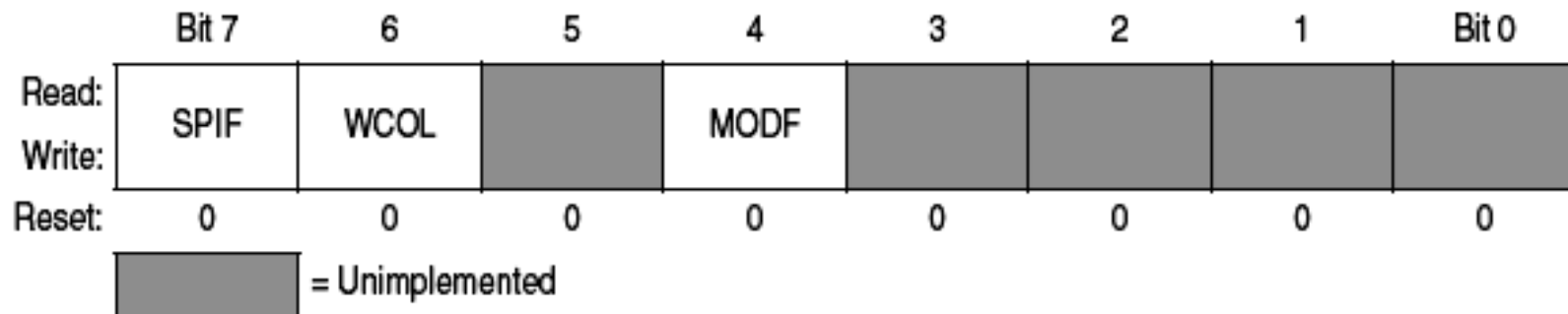
### Статус регистър – SPSR:

**SPIF – флаг за край на обмена /за прекъсване от SPI подсистемата/**  
 SPIF се установява в “1” след приключване обмена на данни между ЕМК и външното устройство. Прекъсване се генерира при SPIF→“1” и ако бита за разрешение на прекъсване SPIE=“1”. За нулиране на флага SPIF – четене се SPSR (при SPIF=“1”), след което се записва в SPDR.

### WCOL - бит за грешка “write collision”

Индицира с “1” грешка при опит за запис на данни в SPDR при все още незавършен обмен. Нулира се при четене от SPSR (при WCOL=“1”), след това – обръщане към SPDR.

Address: \$1029





# Сериен синхронен интерфейс в HC11 – SPI

## Регистри в подсистемата SPI

### Статус регистър – SPSR (продължение):

#### MODF – бит за индициране грешка от режима

Указва с “1” грешка от режима (mode fault) при повече от 1 устройство опитващо се да е Master. За нулиране се чете SPSR (при MODF=“1”), след това - запис в SPCR;

**бит 5, битове [3:0] - не се използват (чете се 0 от тях).**

### Регистър за данни – SPDR

Използва се и при предаване и приемане на данни. Чрез запис в него се инициира приемане/предаване на байт данни (само в Master устройството). След завършване на обмена - флагът SPIF=“1” и в Master и в Slave устройството.

Четене на SPDR – т.е. четене на буферния регистър. **За избягване на “overrun” грешка и загуба на данни – първият флаг SPIF следва да се нулира преди следващия байт данни.**

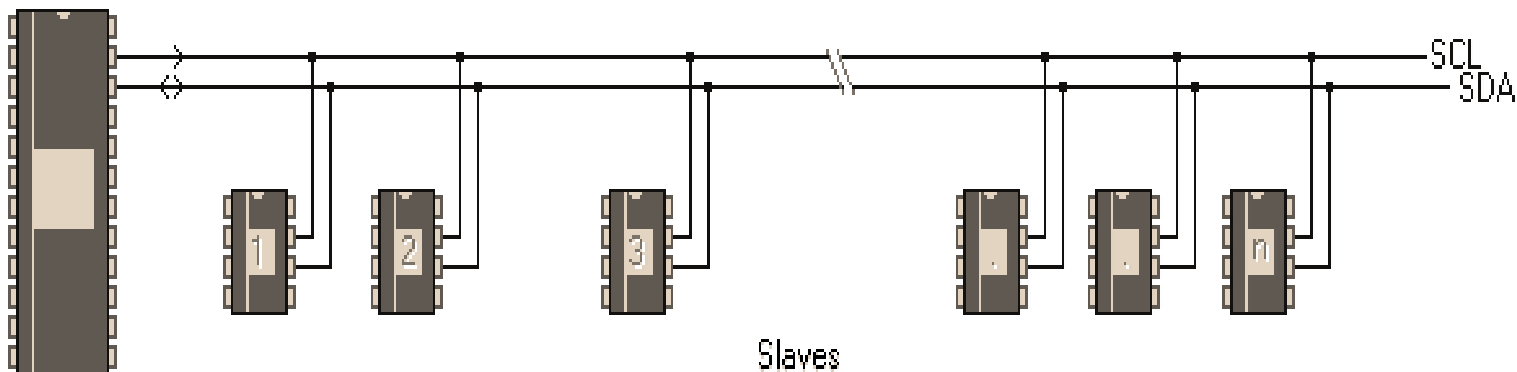
Address: \$102A

	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:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	Indeterminate after reset							

# Интерфейс - I<sup>2</sup>C (Inter-Integrated Circuit)

(електрическа схема, шини, приложения)

- Поява – ранните 80-те год. (Philips Semiconductors): синхронен двупосочен интерфейс от типа Multi-master. 2-жичен bus с обща маса;
  - за връзка между различни ИС - Inter-ICs (I<sup>2</sup>C);
  - активни линии: **SDA** (Serial **D**ata линия), **SCL** (Serial **C**lock линия) – двупосочни;
  - всяко устройство (ЕМК, памет, LCD, ASIC) може да работи и като предавател и като приемник.
- I<sup>2</sup>C е **мулти-мастер bus** – т.е. повече от 1 устройство (**Bus Master**, обикновено ЕМК) може да инициира трансфер. При трансфер останалите устройства са **Bus Slaves**;



# Интерфейс - I<sup>2</sup>C (Inter-Integrated Circuit)

(електрическа схема, шини, приложения)

- **Serial Data (SDA) и Serial Clock (SCL)** – двупосочни линии с ОД (отворен дрейн) и товарен резистор към захранване;
- **захранващи напрежения: 5V / 3.3V**. Възможни са и други, неспециф. стойности (напр. при I<sup>2</sup>C 2.0, 1998 – 2V);
- **максимум 112 устройства** (възела) в обща комуникация Зависи от броя адресни линии и капацитета на шините – 400pF (ограничава разстоянието до няколко метра);
- **7-битов** (до 10-битов) **адрес** (16 резервирани);
- стандартна скорост на обмен – 100Kbit/s до ... 400Kbit/s... 1Mbit/s ... 3.4Mbit/s (high speed mode, понастоящем);
- 10Kbit/s нискоскоростен вариант, възможни са и много по-ниски скорости при необходимост.

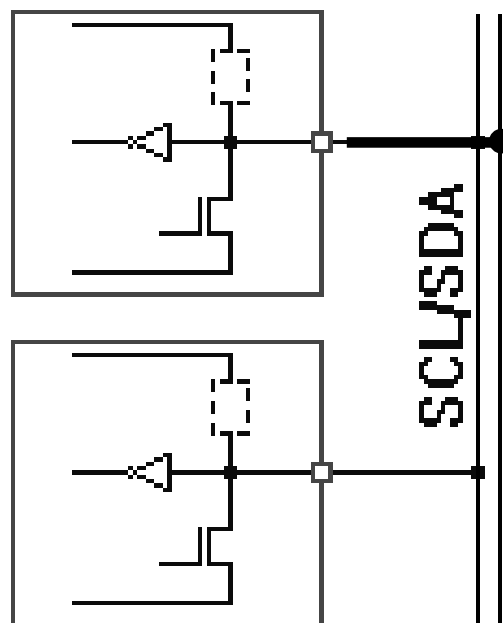
# Интерфейс - I<sup>2</sup>C

(електрическа схема, приложения)

## Електрическа схема

Линиите SDA, SCL – двупосочни. За целта: използва се схема ОК (ОД, отворен дрейн) и структура на входен буфер.

- в състояние IDLE (свободна линия): линията е в H.I.
- за привеждане във функционално състояние: отпушване на изходния транзистор и bus – в ниско ниво. Резистор към захранване (pull-up): играе роля на източник на ток.



# Интерфейс I<sup>2</sup>C

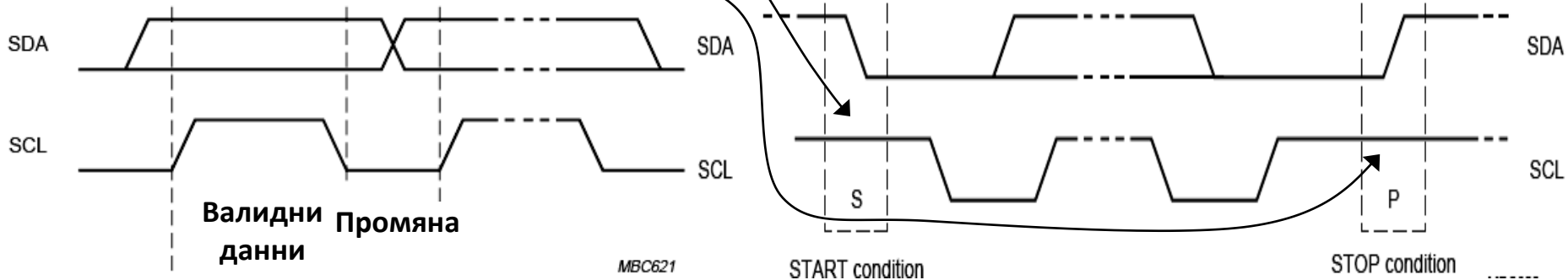
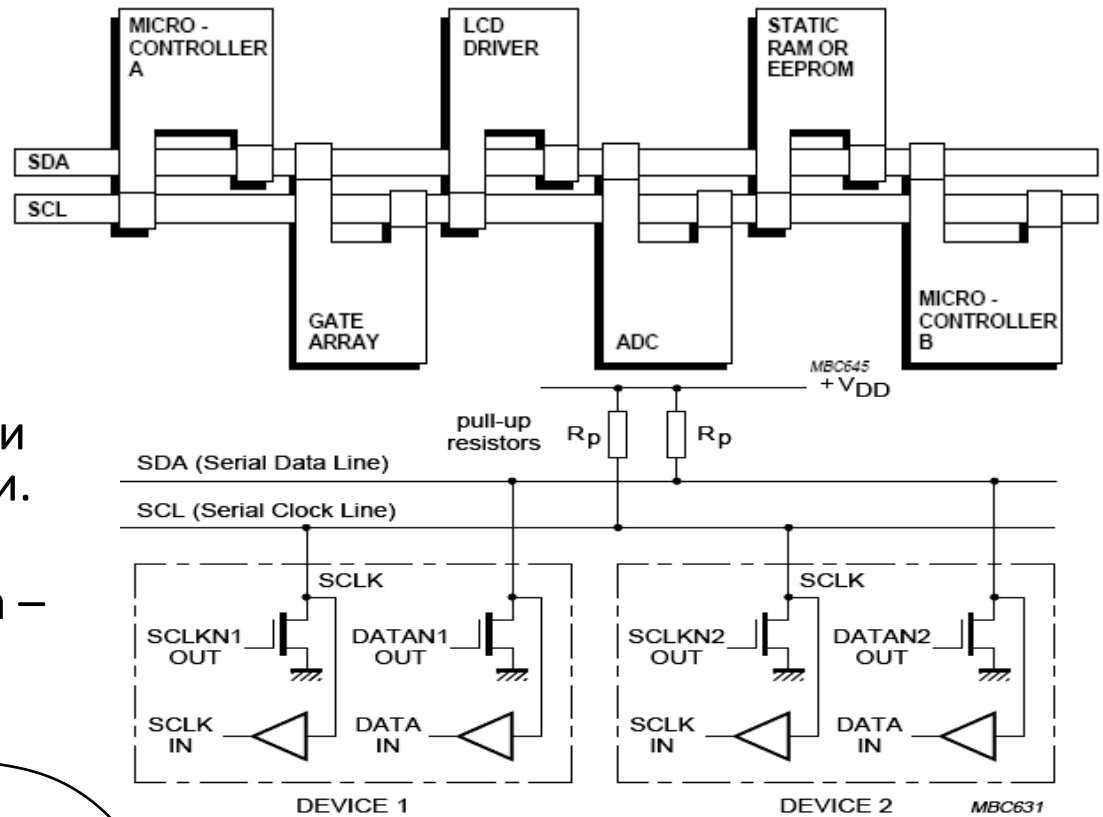
## функции, характеристики

- Примерна схема на свързване с 2 ЕМК и допълн. устройства към I<sup>2</sup>C;
- (Philips Semiconductors)
- електрическа схема на ОД изх.стъпала на устройства при свързване с pull-up резистори.

**Данни – валидни** → при високо ниво на SCL, промяна – само при ниско ниво на SCL.

**Старт** – при преход на SDA 1→0 при SCL="1";

**Стоп** – при преход на SDA 0→1 при SCL="1".



## Интерфейс - I<sup>2</sup>C (Inter-Integrated Circuit) (електрическа схема, приложения)

Последователност на осъществяване на комуникацията:

1. **ЕМК поставя линията в състояние START.** Всички устройства по интерфейса “слушат” за постъпващи данни;
2. **ЕМК (Master) изпраща ADDRESS** към устройствата с които иска да се свърже (операция Read или Write):
  - Адрес: 000...0 (General call) – повикване към всички устройства;
  - при изпращане на актуален адрес - всяко сравнява получения адрес със своя. При съвпадение – отговор със сигнал **ACKNOWLEDGE (ACK)** и то става **Slave устройство**. Ако липсва съвпадение - Not acknowledge;
3. **Начало на обmena на данни;**
4. **Край на обmena** – сигнал за поставяне на линията в състояние **STOP**. Може да се очаква нов обмен.



# Приложение на интерфейси SPI, I<sup>2</sup>C

## Разширяване възможностите на ЕМК

### Интерфейс SPI:

- Недостатък – не е напълно bus: НЕ МОЖЕ ДА АДРЕСИРА отделните устройства със специфичен адрес. Използва т.нар. push-pull технология;
- Предимства – висока скорост, простота на обmena. Предпочита се при: програмиране на EEPROM, Flash, връзка с LCD дисплеи, системи за събиране на данни от сензори.

### Интерфейс I<sup>2</sup>C:

- Предимства – адресира свързаните устройства, ралични захр.напрежения, вариации в скоростта на трансфер;
- Недостатък – сложен интерфейс ПК/ЕМК – bus. Philips предлага адаптер, обслужващ Single-Master връзка. Алтернативи варианти: Tracii XL 2.0, Connii MM 2.0 (със собствени логически схеми за връзка по I<sup>2</sup>C).



## Приложение на интерфейси SPI, I<sup>2</sup>C

Разширяване възможностите на ЕМК – връзка с други серийни устройства

### Синхронни серийни интерфейси:

- с 3 (4) линии за връзка: SPI, Queued SPI (QSPI), MICROWIRE (MICROWIRE PLUS);
- с 2 линии за връзка: I<sup>2</sup>C, SMBus (System Management Bus).

**с 3 (4) линии за връзка:** Използва линиите  $\overline{CS}$  ( $\overline{SS}$ ), SCLK и за данни MOSI (DIN, SI), допълнително MISO (DOUT, SO) .

### Предимства:

- висока тактова честота;
- не изискват резистори от линиите към захранване (pull-up резистори);
- full-duplex режим на работа (едновременно предаване/приемане на данни);
- синхронизация: по фронт (не по ниво) - по-висока шумозащитеност;
- по просто софтуерно управление.

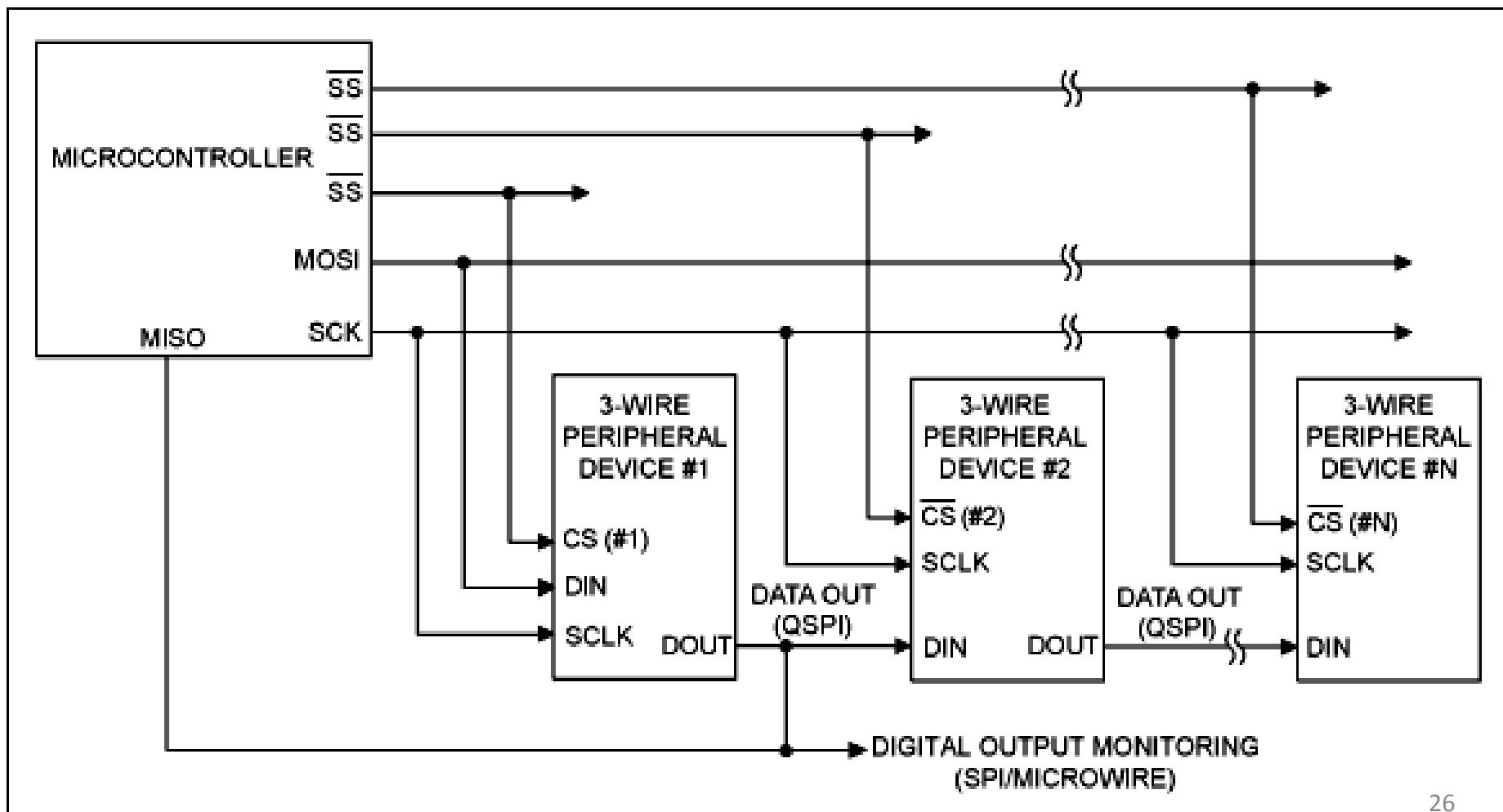
### Недостатъци:

- изискват потвърждение при обмена;
- необходимост от отделна линия  $\overline{SS}$  ( $\overline{CS}$ ) за всяко Slave устройство;  
Изключение – при т.нар. “daisy-chain” свързване (но повече линии  $\overline{SS}$ ).

## Приложение на интерфейси SPI, I<sup>2</sup>C

Разширяване възможностите на ЕМК – връзка с други серийни устройства

Свързване на устройства по **SPI интерфейс от типа “daisy-chain”**



## Приложение на интерфейси SPI, I<sup>2</sup>C

Разширяване възможностите на ЕМК – връзка с други серийни устройства

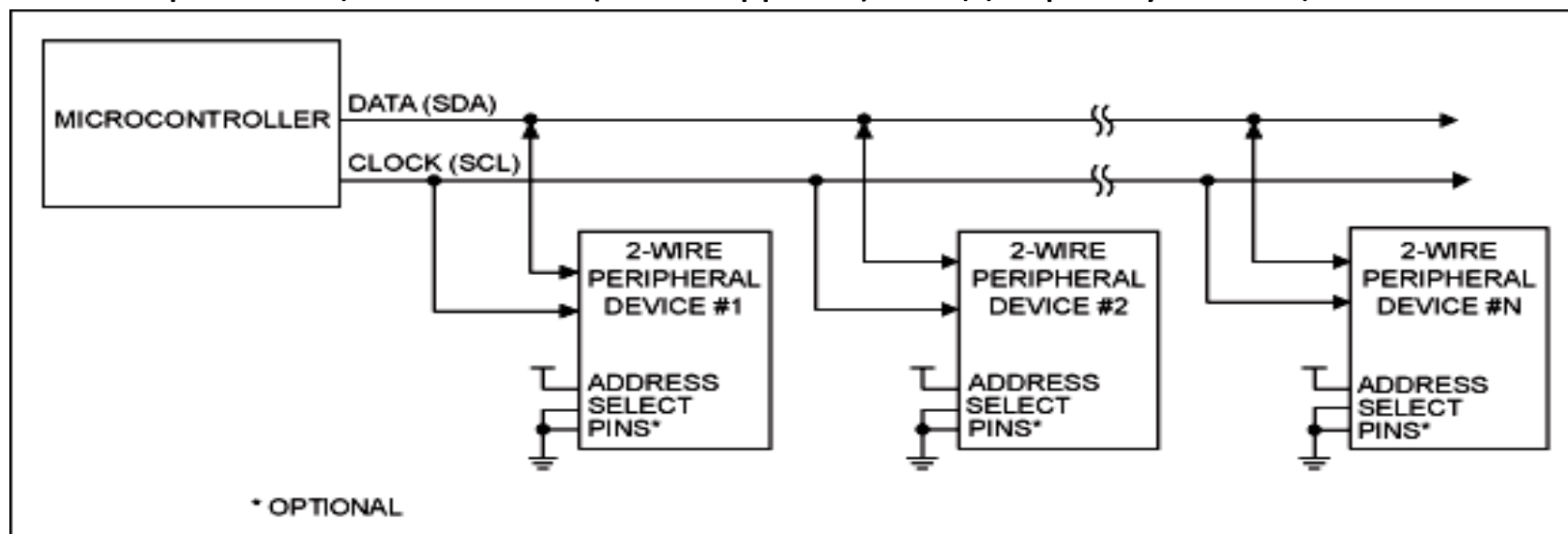
**с 2 линии за връзка**: Използват само линиите за данни SDA (SMBDATA) и за тактова SCL (SMBCLK).

### Предимства:

- при компактни устройства (клетъчни телефони, устройства с оптична връзка);
- включване на няколко Slave устройства без допълнителен CS (всяко устройство има свой собствен адрес);

### Недостатъци:

- допълнителен бит за потвърждаване в обмена;
- half-duplex режим на обмен (само 1 линия за данни);
- синхронизация по ниво (не по фронт) - недобра шумозащитеност.

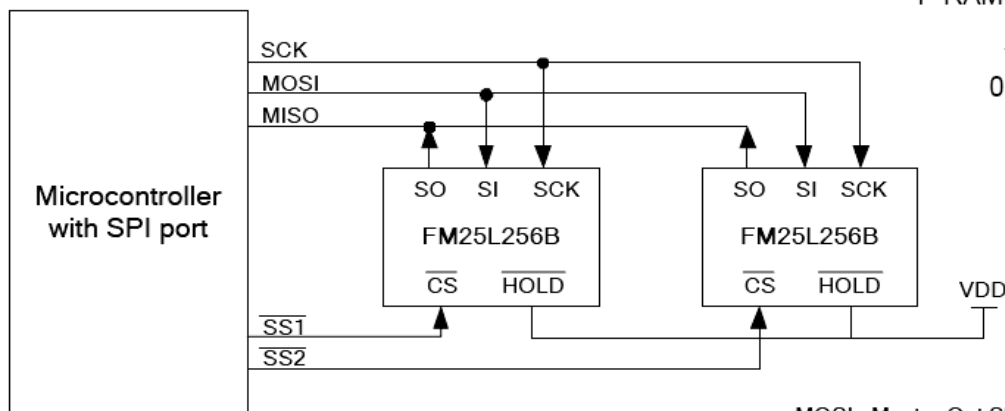
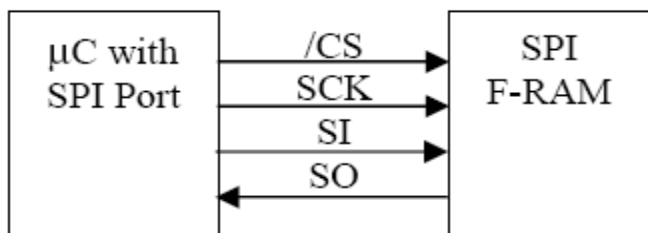


# Схеми със синхронен сериен интерфейс

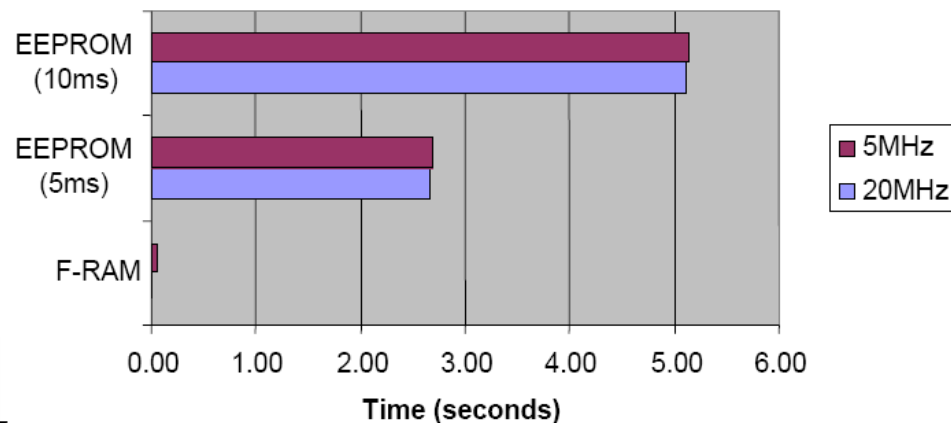
## F-RAM (Fast-RAM, Ramtron Corp.):

Специално разработен тип RAM за връзка по SPI.

- ВИСОКА СКОРОСТ на запис – до 40Mbit/s (не се предава адрес);
- не ползва Page буфер, както “стандартните” EEPROM и Flash т.е. – директен трансфер байт по байт;
- опростен и адаптиран интерфейс – само шини SI (Serial In), SO (Serial Out), SCK, CS.



MOSI : Master Out Slave In  
MISO : Master In Slave Out  
SS : Slave Select



# Схеми със синхронен сериен интерфейс

## Серийни EEPROM (Serial EEPROM):

Схеми на EEPROM за сериен обмен.

### Предимства:

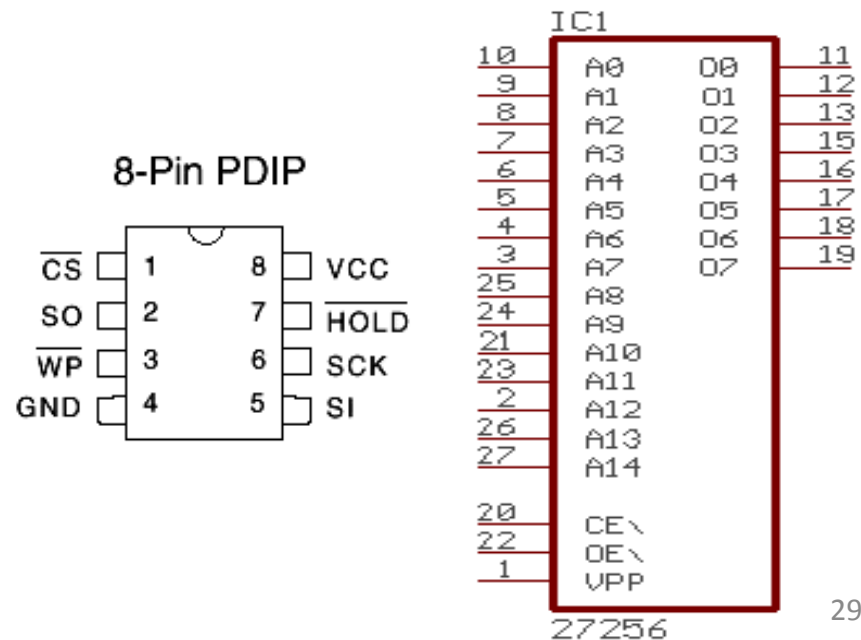
- ползва само 4 линии от ЕМК за връзка (SI,SO,CS,SCK), вместо A/D магистрала. Добавени още HOLD, WP;
- опростена логика в схемата за изграждане на интерфейса;
- ниска цена (малък брой изводи; липса на поддържащи схеми).

Недостатък: голямо време за достъп при четене (бит по бит)

### *Пример (сравнение):*

8-изводна серийна EEPROM

28-изводна (15 адресни шини, 8-за данни и др.) EEPROM 27256;



## USB интерфейс

(общи сведения, шини, протокол на обмен, приложение)

**USB (Universal Serial Bus)** – 1994, специфициран интерфейс по отношение на: Електрически, Механични и Протоколни изисквания /layers/. Отнася се за Host, Hub, или Function (Hub-Hub, Hub-USB Function, Hub-USB bus, Host-Hub). Добива популярност чрез драйвери в Windows'95 (1996).

- **7-битов адрес** (Master-slave структура с до 127 устройства свързани заедно, топология звезда);
- **диференциален сериен интерфейс /bus/, half-duplex** обмен (USB 3.0 –full duplex);
- **4 шини**, 2 от тях: за данни по усукана двойка. Осигурява захранване на свързаните устройства;
- данни, предавани в **NRZI код** – асинхронно или синхронно между свързаните устройства;
- максимална дължина на кабела – **5м**;
- три (четири) групи скорости на предаване от 10kbps до 400Mbps:
  - **Slow-Speed** режим /от 10Kbps до 100Kbps/ - обмен с бавни устройства (клавиатури, мишки);
  - **Full-Speed** режим /от 500Kbps до 10Mbps/;
  - **High-Speed** режим - USB 2.0 /от 25Mbps до 400Mbps/. Изисква допълнителен резистор  $45\Omega$  между всяка линия за данни и маса.  $U^1=2.8V, U^0=0.3V$ .
  - **Super speed** - USB 3.0 /до 4.8Gbps/ - и за оптична комуникация.

## USB интерфейс – протоколи, формат на данните, пакети

**USB** (Universal Serial Bus) – четири различни (пакетни) типа обмен:

- Interrupt** (при т.нар. “end-point” устройства – клавиатури, мишки);
- Isochronous** (обмен на аудио/видео данни);
- Bulk** (при файлов обмен);
- Control**.

**Формат на обмена.** Състои от 3 пакета:

- Начален пакет (*Token packet*) – указва какво ще се прави R/W и съдържа адреса на крайното устройство;
  - Пакет с данни (*Optional Data packet*) – съдържа данни. Предават се с LSB пръв;
  - Handshake (статус) пакет – за потвърждаване на предаването и за индициране на грешки, (напр. устройство с невъзможност да приема) , за край на обмена.
- **Кодирание на данните - NRZI** (1 – промяна; 0 – без промяна).
  - Използва т.нар. **“Bit shifting”** за предотвратяване на “зависване” на сигнала в 1 (добавя 0 след шест 1-ци) и генериране на обмен (седем 1-ци са грешка).
  - **Самосинхронизация** между предавател и приемник.

## USB интерфейс – протоколи, формат на данните, пакети

Полета (Fields) в пакетите - видове, структура:

- ❑ **Sync** – ВСИЧКИ видове пакети започват със Sync: 8-бита (Slow-speed, Full-speed) или 32-бита (High-speed). За синхронизация между предавател и приемник. Последните 2 бита – указват къде започва PID;
- ❑ **PID (Packet ID)** – за идентификация на изпращания пакет (4 бита). Добавят се още 4 бита в допълнителен код – общо 8 бита.

Тип протокол	PID стойност	Идентификатор на пакет
Token	0001	OUT Token
	1001	IN Token
	0101	SOF Token
	1101	SETUP Token
Data	0011	DATA0
	1011	DATA1
	0111	DATA2
	1111	MDATA
Handshake	0010	ACK Handshake
	1010	NAK Handshake
	1110	STALL Handshake
	0110	NYET (No Response Yet)
Special	1100	PREamble
	1100	ERR
	1000	Split
	0100	Ping

PID <sub>0</sub>	PID <sub>1</sub>	PID <sub>2</sub>	PID <sub>3</sub>	nPID <sub>0</sub>	nPID <sub>1</sub>	nPID <sub>2</sub>	nPID <sub>3</sub>
------------------	------------------	------------------	------------------	-------------------	-------------------	-------------------	-------------------



## USB интерфейс – протоколи, формат на данните, пакети

Полета (Fields) в пакетите - видове, структура (продължение):

- ADDR** – указва адреса на устройството за обмен – 7 бита.  
Адресира до 127 устройства (адрес 00...0 – невалиден);
- ENDP** – 4 бита (общо 16 възможни крайни точки, Endpoints).  
За обмен Slow-speed – макс. 4 точки;
- CRC** (Cyclic Redundancy Checksum): за всеки Token пакет - 5-битов CRC,  
за всеки пакет данни - 16-битов CRC;
- EOP** - край на пакет. Известява със SE0 (Single Ended Zero)-2 бита +1бит 0.

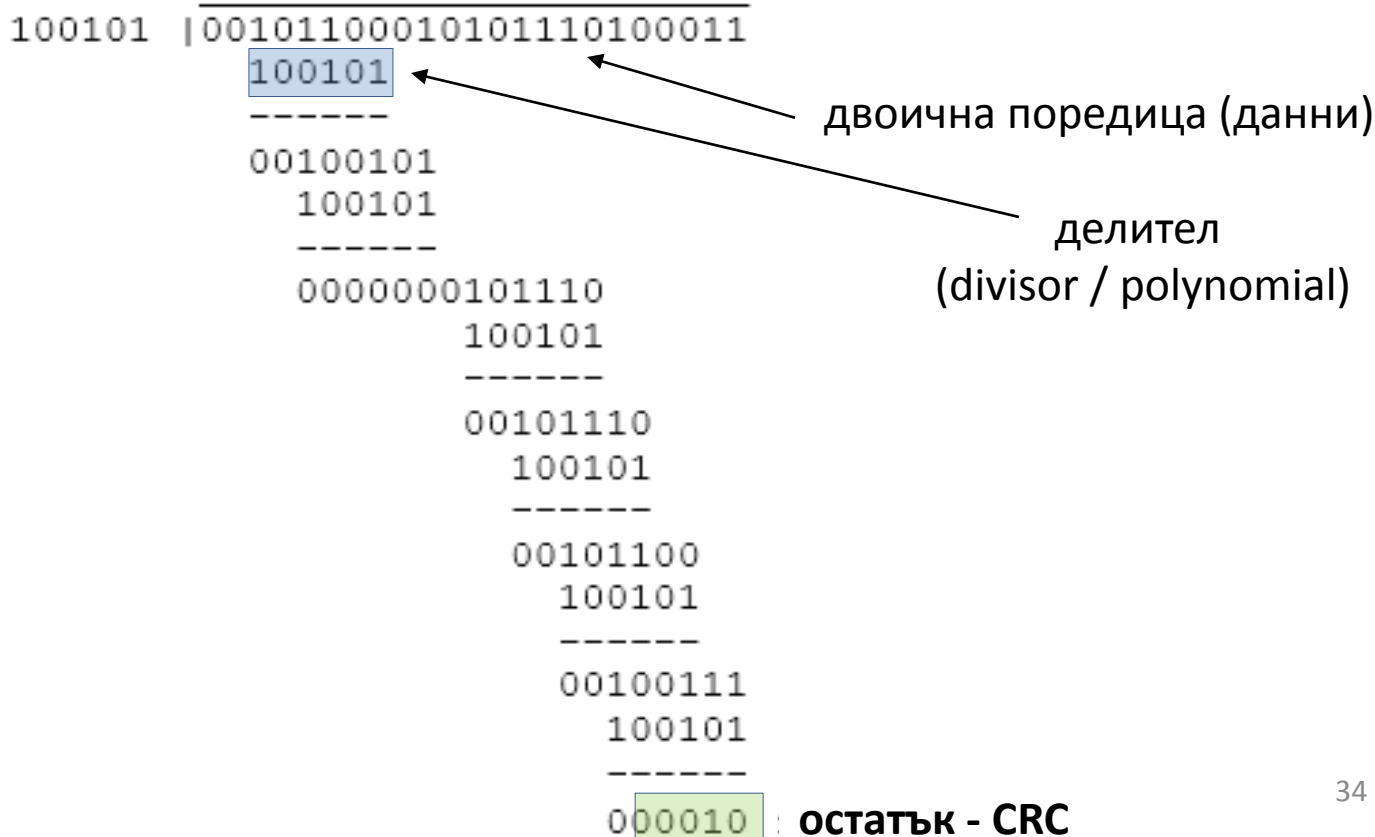
Типове пакети и формати:

**Token пакет** – видове: **In** (информира USB устройството че Host иска да получава информация), **Out** (информира USB устройството че Host иска да изпраща информация), **Setup** (за начало на контролен обмен). Формат:

**Sync    PID    ADDR    ENDP    CRC5    EOP**

## USB интерфейс – CRC (Cyclic Redundancy Checksum) – Wesley Peterson

- Базира се на теорията на т.нар. “**cyclic error-correcting codes**” – чрез добавяне на check-сума с фиксирана дължина - за откриване на случайни промени (грешки от шум) при предаване на големи последователности от двоични данни - при предаване по интерфейс или при съхранение (HDD);
- за всеки пакет данни се изчислява CRC код (с фиксирана дължина) и се изпраща заедно с данните. При получаване се изчислява отново – при разлика → грешка;
- Пример: изчисляване на 5-битов CRC код (6-битов делител).



## USB интерфейс – протоколи, формат на данните, пакети

### Типове пакети и формати:

**Data пакет** – съдържа данни. 2 типа пакети: 8 байта (Slow-speed) до 1024 байта (Full speed, High speed): **Data0, Data1**, (High Speed - **Data2, MData** ):

<b>Sync</b>	<b>PID</b>	<b>Data</b>	<b>CRC16</b>	<b>EOP</b>
-------------	------------	-------------	--------------	------------

**Handshake пакет** – 3 типа пакети (съдържат само PID):

**ACK** – потвърждава, че пакета е успешно приет;

**NAK** – съобщава, че устройството временно не може да изпраща/получава данни. Използва се и при Interrupt обмен за информирание на Host че не се изпращат данни;

**STALL** – устройството е в състояние, изискващо намеса на Host.

<b>Sync</b>	<b>PID</b>	<b>EOP</b>
-------------	------------	------------

## USB интерфейс – протоколи, формат на данните, пакети

Типове пакети и формати:

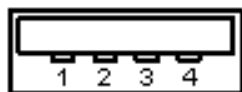
**SOF - Начало на фреймовете пакети (Start of Frame Packets)** – 11 битов пакет с номер на фрейма който се изпраща от Host на всеки  $1\text{ms} \pm 500\text{ns}$  (режим Full speed) или на всеки  $125\mu\text{s} \pm 0.0625\mu\text{s}$  (режим High speed).

Формат на пакета:



Конектори:

**A-тип**



**B-тип**



Краче	Цвят	Функция
1	Червен	$V_{\text{BUS}}$ (5V)
2	Бял	D-
3	Зелен	D+
4	Черен	GND (маса)