

## *Outras Características em Microprocessadores*

8051

AULA 11

*Prof. Adilson Gonzaga*

# Outras características importantes em Microprocessadores/Microcontroladores

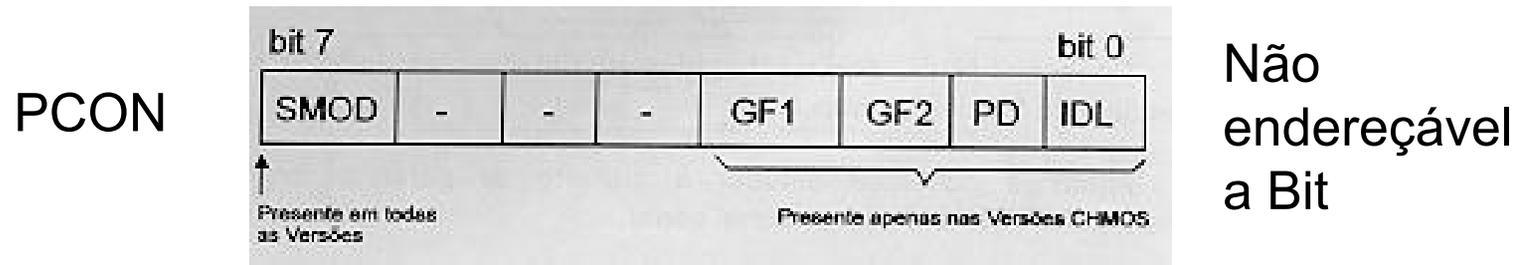
## Redução de Potência de Operação

As versões CHMOS (89C51, 89S52, etc...) da família MCS-51 possuem dois modos de controle de redução de potência de operação do chip .

1. Modo Idle

2. Modo Power Down

O Registrador (SFR) utilizado para o controle de potência é o PCON.



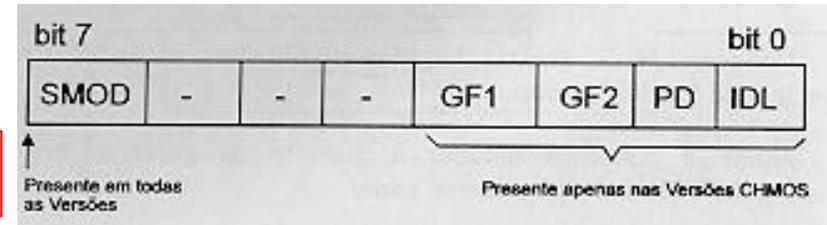
## Modo Idle

O consumo é reduzido em 85%

IDL = 1 → o oscilador funciona somente para as Interrupções e os Timers.

- Não existe sinal de clock para a CPU.
- O status da CPU é preservado: o SP, o PC, o PSW, o ACC e todos os outros registradores mantêm seus dados durante o tempo em que a CPU estiver em Modo Idle.
- As Portas mantêm seus estados lógicos que tinham quando o Modo Idle foi ativado .

Existem 2 maneiras de terminar o Modo Idle:



## 1. Ativação de uma Interrupção habilitada

- Zera automaticamente o bit IDL
- Atende a sub-rotina de Interrupção
- Ao executar o RETI entra novamente no Modo Idle
- Os Flags GF1 e GF2 podem ser utilizados para indicar se as interrupções ocorreram durante o modo normal de operação ou durante o Modo Idle

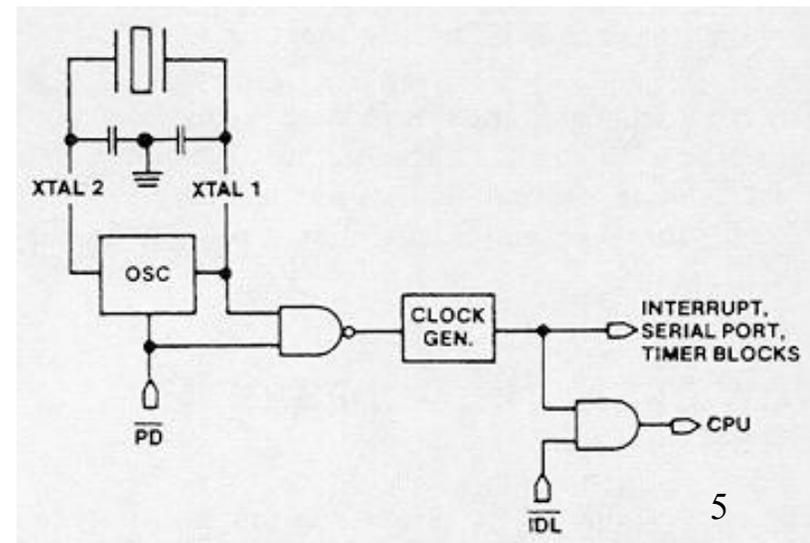
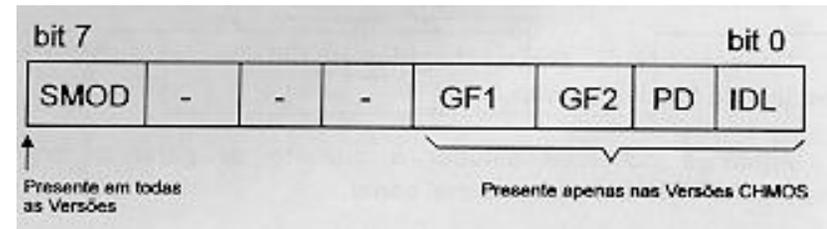
## 2. Reset de Hardware

## Modo Power Down

PD = 1 → O oscilador pára.

- Todas as funções param.
- O conteúdo da RAM interna e dos SFRs se mantém.
- Neste Modo Vcc pode ser reduzido para até 2 Volts.
- A única forma de sair deste modo é por meio do reset de hardware .
- O consumo é da ordem de 10 uA.

PD tem precedência sobre IDL



# Outras características importantes em Microprocessadores/Microcontroladores

## Bits de Proteção (Lock Bits)

- São Bits que permitem a proteção do código do usuário gravado em um Microcontrolador.
- Para a família MCS-51 existem 3 Bits de proteção que podem ser gravados durante a fase de programação do chip.

Lock Bit Protection Modes<sup>(1)(2)</sup>

	Program Lock Bits			Protection Type
	LB1	LB2	LB3	
1	U	U	U	Sem nenhuma proteção(A Memória de Programa pode ser lida)
2	P	U	U	As instruções MOVC executadas da Memória de Programa Externa são desabilitadas. A programação do chip também é desabilitada.
3	P	P	U	Mesmo do Modo 2, mas também a verificação do programa é desabilitada.
4	P	P	P	Mesmo do Modo 3, mas a execução de programa externo é desabilitada.

Notes: 1. U = Unprogrammed  
2. P = Programmed

- Uma vez programados, os Bits de Proteção só podem ser alterados por uma operação de Chip Erase.

# Outras características importantes em Microprocessadores/Microcontroladores

## Watchdog Timer



Dispositivo de temporização de Hardware que “Reseta” o Microcontrolador se o programa, devido a falhas, negar fornecer o serviço do “watchdog” (por exemplo, escrever em uma determinada posição de memória ou em determinado Bit).

- Simples → apenas Reseta o Micro
- Complexo → Salva informações de “debug” em mídias fixas.

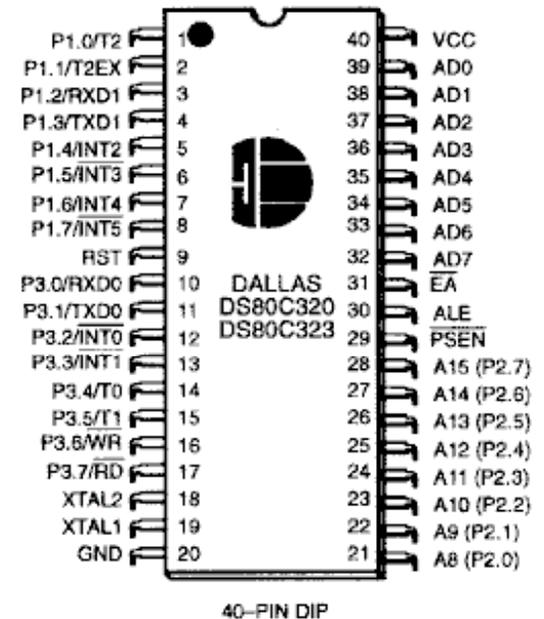
A rotina de atendimento ao “Watchdog Timer” pode colocar o Microcontrolador em um “estado seguro” desligando motores, saídas de alta tensão, etc, até que a falha seja resolvida.

## Exemplo de Microcontrolador com Watchdog Timer implementado no chip.

- DS80C320 fabricado pela **Dallas Semiconductor** (núcleo do 8051)

- Quando o Watchdog é habilitado, o programa do usuário deve escrever em um determinado SFR regularmente permitindo ao Watchdog saber se o programa está executando corretamente.

- Se o programa não realizar a escrita dentro de um determinado período de tempo, o Watchdog assume que o programa falhou e realiza um Reset ou uma Interrupção, dependendo da configuração.



# Outras características importantes em Microprocessadores/Microcontroladores

## Programação de Microcontroladores

**Microcontroladores com Memória de Programa Flash podem ser programados de 3 maneiras:**

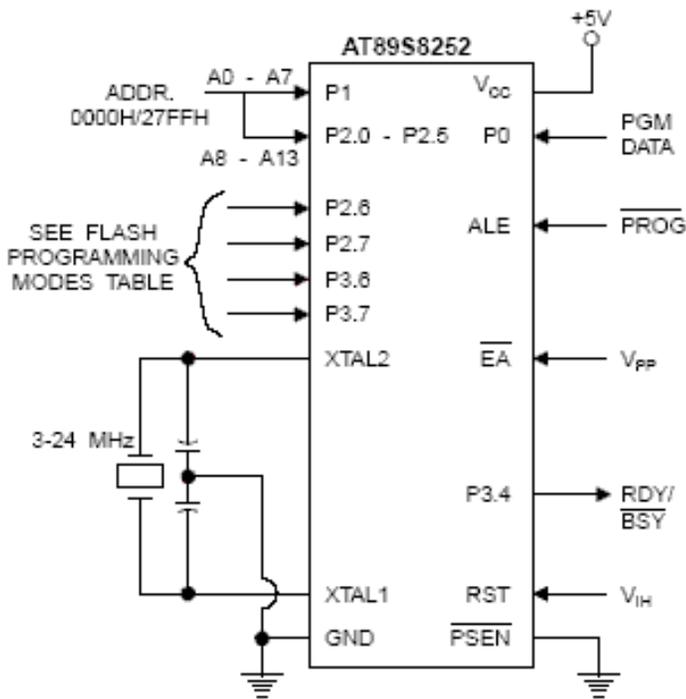
1. Programação Paralela (método Tradicional) → Exige um dispositivo – Programador – O chip do Microcontrolador é programado separadamente. Única metodologia de programação para dispositivos com memória EPROM.

2. In-System Programming (ISP)

3. In Application Programming (IAP)

**Programação In-Circuit**

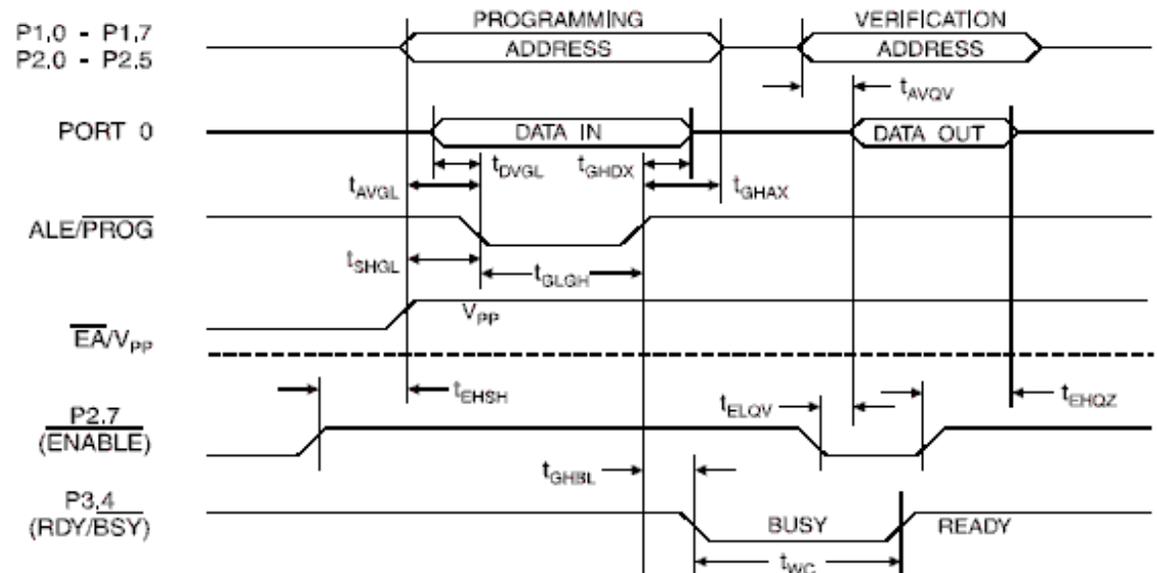
## Programação paralela do Microcontrolador 89S52.



- Os Pinos P2.6, P2.7, P3.6, P3.7 selecionam a operação: Apagar, Ler, Escrever, Gravar Lock Bit, etc...

- Os endereços de cada linha do código deve ser fornecido pelas portas P1 e P2 (P2.0 a P2.5).

- Os dados (códigos do programa) devem entrar pela porta P0.



## **In-System Programming (ISP)**

**Processo no qual um Microcontrolador montado em uma placa de circuito impresso pode ser apagado ou programado com o código final do usuário sem removê-lo da placa.**

- Para a programação o Microcontrolador deve ser ligado no “Modo ISP” .**
- O Microcontrolador recebe comandos e dados de um software localizado por exemplo em um PC.**
- Terminada a operação de ISP o dispositivo é reconfigurado para que possa operar normalmente.**

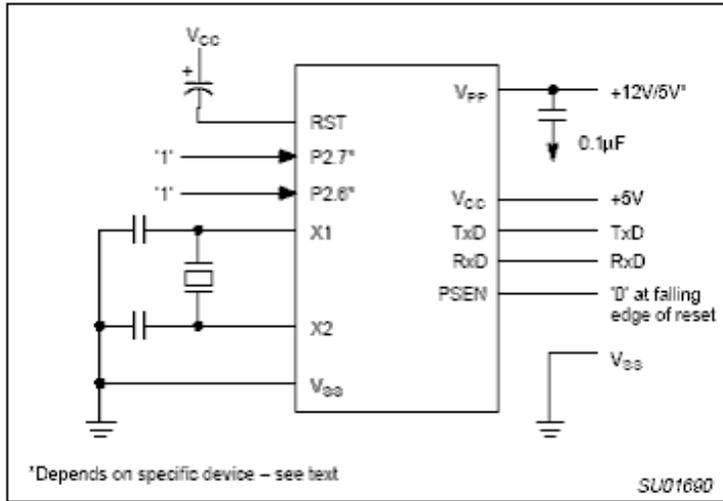
## In-System Programming (ISP)

- A programação “in-circuit” via ISP pode ser realizada pelos canais seriais do Microcontrolador.

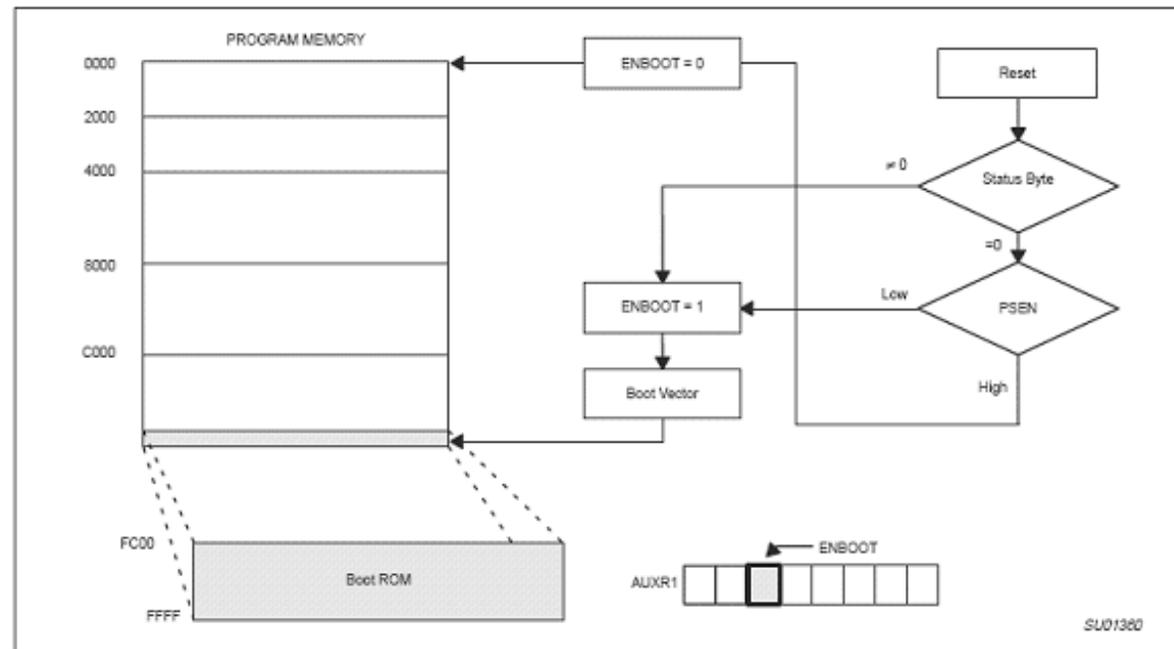
- a. Via RxD e TxD da interface serial RS232
- b. Via Interface SPI (Serial Peripheral Interface)

- Os Microcontroladores que possuem a possibilidade de programação ISP tem um programa localizado na parte superior da Memória de Programa (**BOOTROM**) chamado de **BOOT LOADER**.

# In-System Programming (ISP)



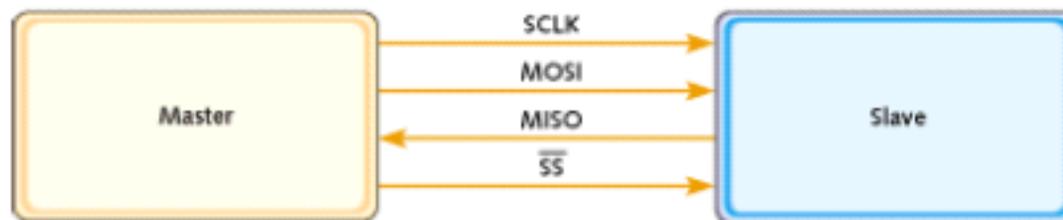
## Exemplo: Via interface Serial RS232



ISP Flow Chart

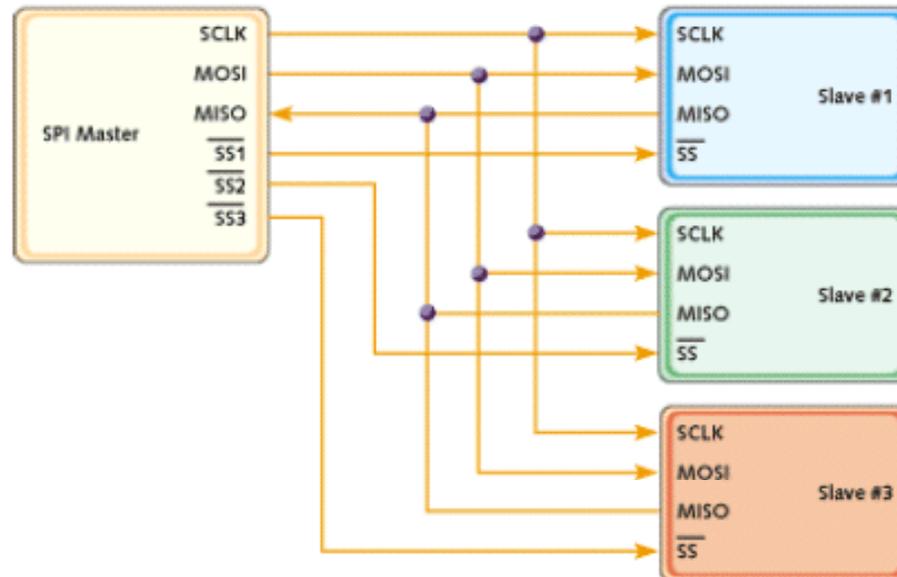
## In-System Programming (ISP)

**SPI (Serial Peripheral Interface)** é um duto padrão serial, síncrono, desenvolvido pela Motorola e utilizado por diversos fabricantes de Microcontroladores.



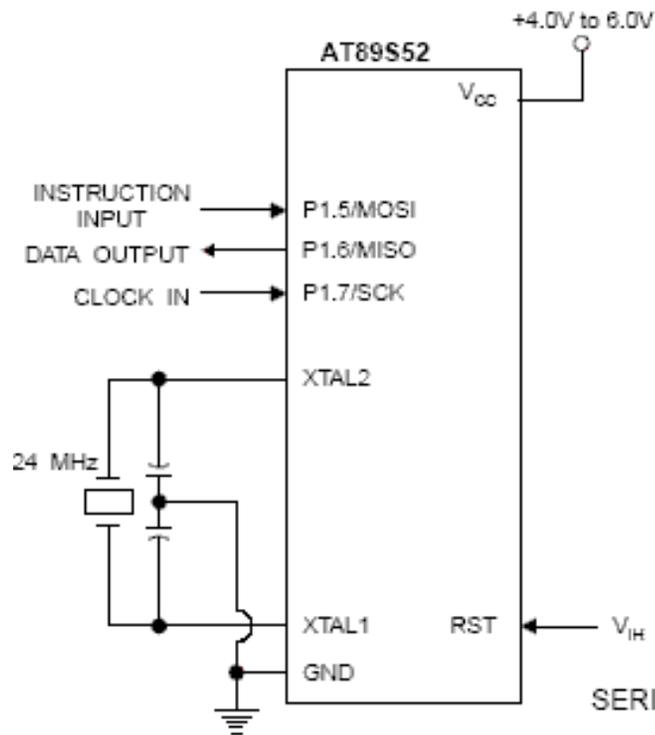
- Os dispositivos se comunicam em uma relação Mestre/escravo onde o Mestre inicia a comunicação.
  - **SCLK** → Clock gerado pelo Mestre
  - **MOSI** → Master data Output, Slave data Input
  - **MISO** → Master data Input, Slave data Output
  - **$\overline{SS}$**  → Slave Select

**Em um esquema de múltiplos escravos, o Mestre deve gerar sinais separados de Slave Select (SS) para cada Escravo.**



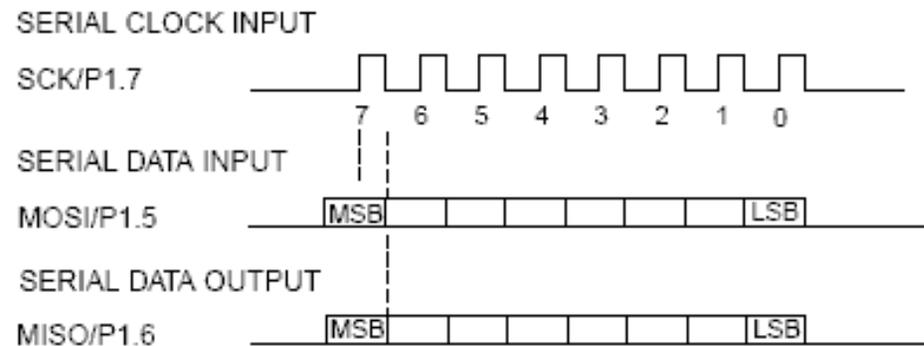
- O SPI não tem um mecanismo de confirmação de dado recebido (acknowledgement).
- O SPI é indicado para aplicações que transferem diretamente um cordão de Dados (Data Stream).
- Razão de Transmissão do SPI = 1Mbps

## Utilização do duto SPI do Microcontrolador 89S52 para programação “in-circuit”.



- Durante a transferência dos dados o pino de RESET deve ficar em nível lógico 1.

- Após a gravação, deve-se colocar o pino de RESET em zero para operação normal do chip.



## In Application Programming (IAP)

- Algumas aplicações tem a necessidade de apagar ou re-programar o código do programa sob controle da própria aplicação.

### **Exemplo:**

- Armazenar informações de calibração “em voo”
  - Carregar novas partes do código do programa “em voo”
- 
- A capacidade de um Microcontrolador de apagar ou re-programar o código do programa na aplicação do usuário (**end-user**) é conhecida como IAP.
  - Em geral, as rotinas do Boot Loader que realizam a programação no modo ISP, também são responsáveis pelo Modo IAP.
  - Não existe um Mestre para gravar o programa. A própria aplicação pode realizar a tarefa.

# Outras características importantes em Microprocessadores/Microcontroladores

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

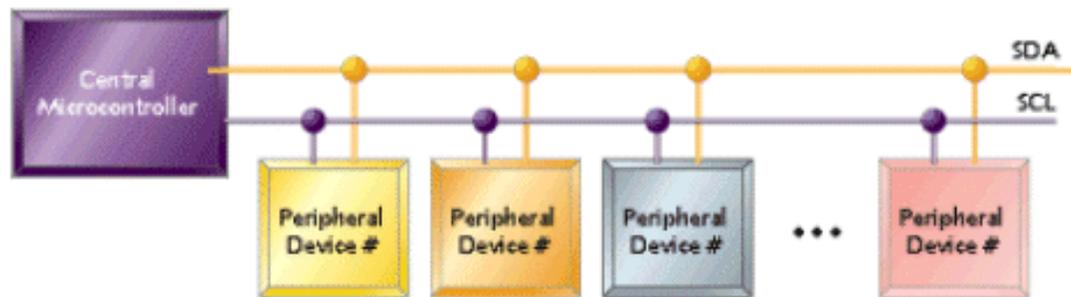
- Duto de comunicação Serial desenvolvido pela Phillips para interligação de periféricos dentro da mesma placa de circuito impresso.

### • Exemplos de dispositivos I<sup>2</sup>C:

- Microcontroladores
- Memórias
- Sensores
- Conversores A/D e D/A
- Sistemas Embarcados

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

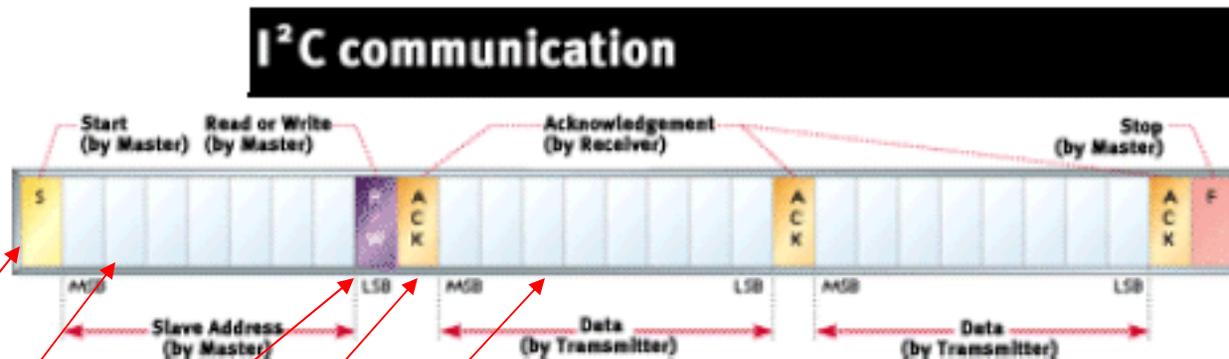
- I<sup>2</sup>C é um Duto serial de 2 fios
- Não existe Chip Select
- É barato e simples de implementar em hardware



**SDA → Dado Serial**  
**SCL → Clock Serial**

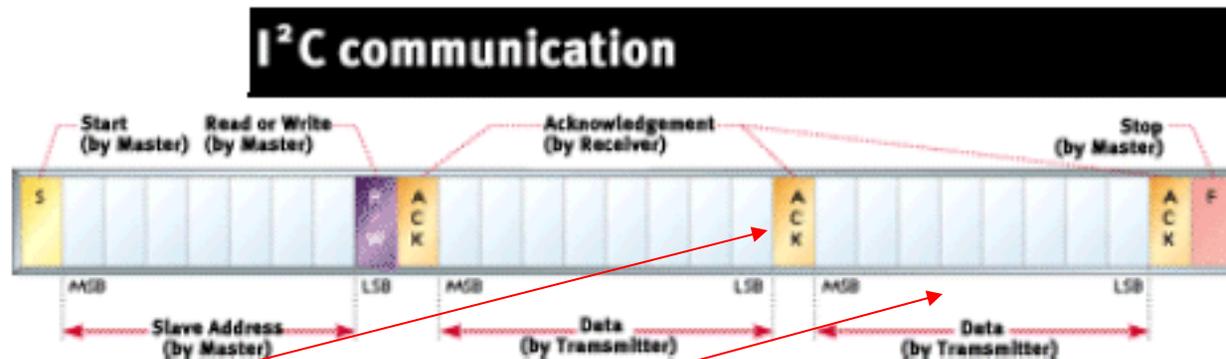
- O dispositivo que inicia a transação é o Mestre e os dispositivos endereçados são os Escravos.
- Cada dispositivo compatível com I<sup>2</sup>C vem com um endereço pré-definido.

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



1. O Mestre inicia a comunicação enviando a condição START → S=1
2. O Mestre envia o endereço de 7 Bits do dispositivo Escravo
3. O oitavo Bit (R/W) especifica se a operação é de Leitura (1) (Escravo Transmite) ou de Escrita (0) (Escravo Recebe)
4. O Escravo envia um ACK =1 se o Byte anterior foi recebido
5. O Transmissor (Mestre ou Escravo) transmite um Byte de dados iniciando pelo MSB

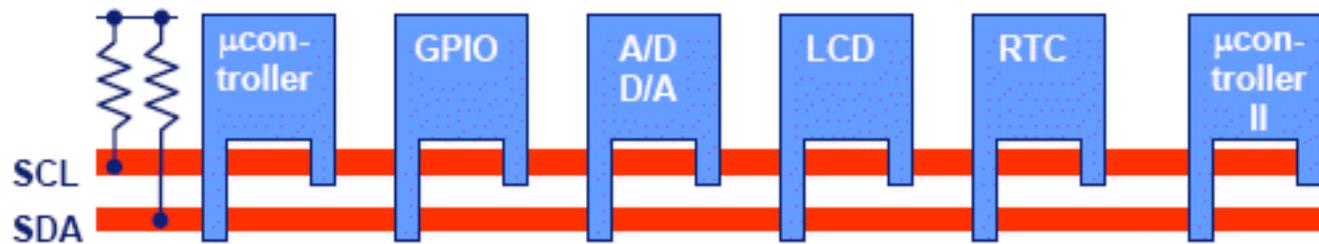
## Duto I<sup>2</sup>C ( Inter Integrated Circuit)



6. No final do Byte, o receptor (Mestre ou Escravo) envia um bit de ACK
7. O padrão de 9 Bits (8 Bits de Dados e ACK) é repetido se novos bytes necessitarem ser transmitidos.
8. Numa Transação de Escrita (Escravo Recebe), quando o Mestre termina a transmissão de todos os Bytes ele monitora o último ACK enviado pelo Escravo e termina enviando uma condição de Parada (STOP=1)
9. Numa Transação de Leitura (Escravo Transmite), o Mestre não envia o último ACK, o que informa ao Escravo que a transação acabou. O Mestre envia então a condição de STOP.

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

- Os Dispositivos I<sup>2</sup>C são interconectados diretamente no duto.
- O Mestre é quem envia o Clock



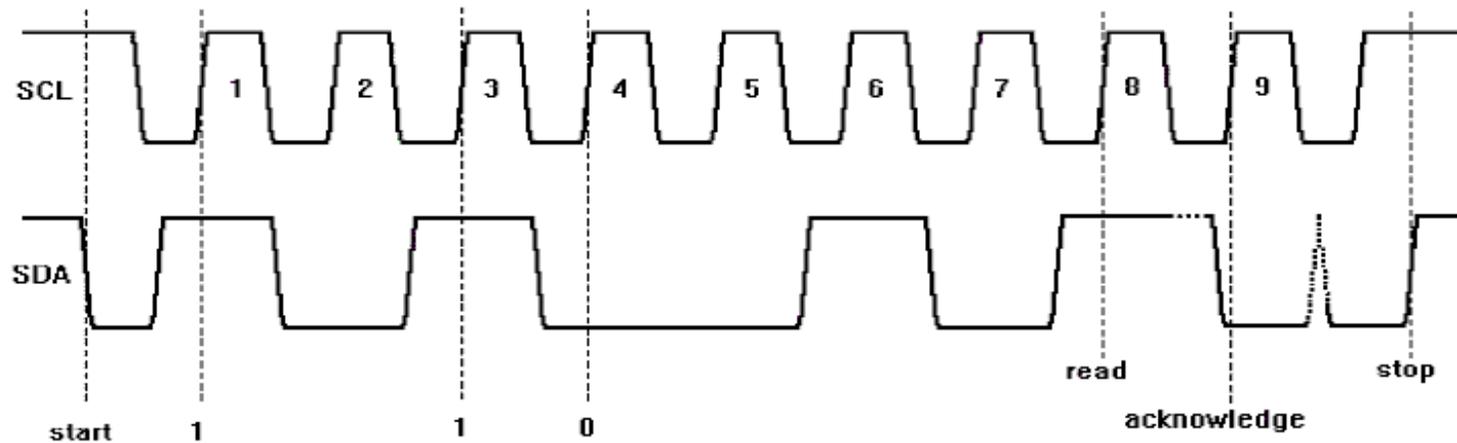
### I<sup>2</sup>C by the numbers

	Standard-Mode	Fast-Mode	High-Speed-Mode	
			0 to 1700	0 to 3400
Bit Rate (kbits/s)	0 to 100	0 to 400	0 to 1700	0 to 3400
Max Cap Load (pF)	400	400	400	100
Rise time (ns)	1000	300	160	80
Spike Filtered (ns)	N/A	50	10	
Address Bits	7 and 10	7 and 10	7 and 10	

- Capacidade para comunicação de 20 a 30 dispositivos simultâneos em uma distância de até 4 metros.

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

### Operação do Duto I2C



O Dado (SDA) é lido durante uma subida de borda da linha de Clock (SCL)