

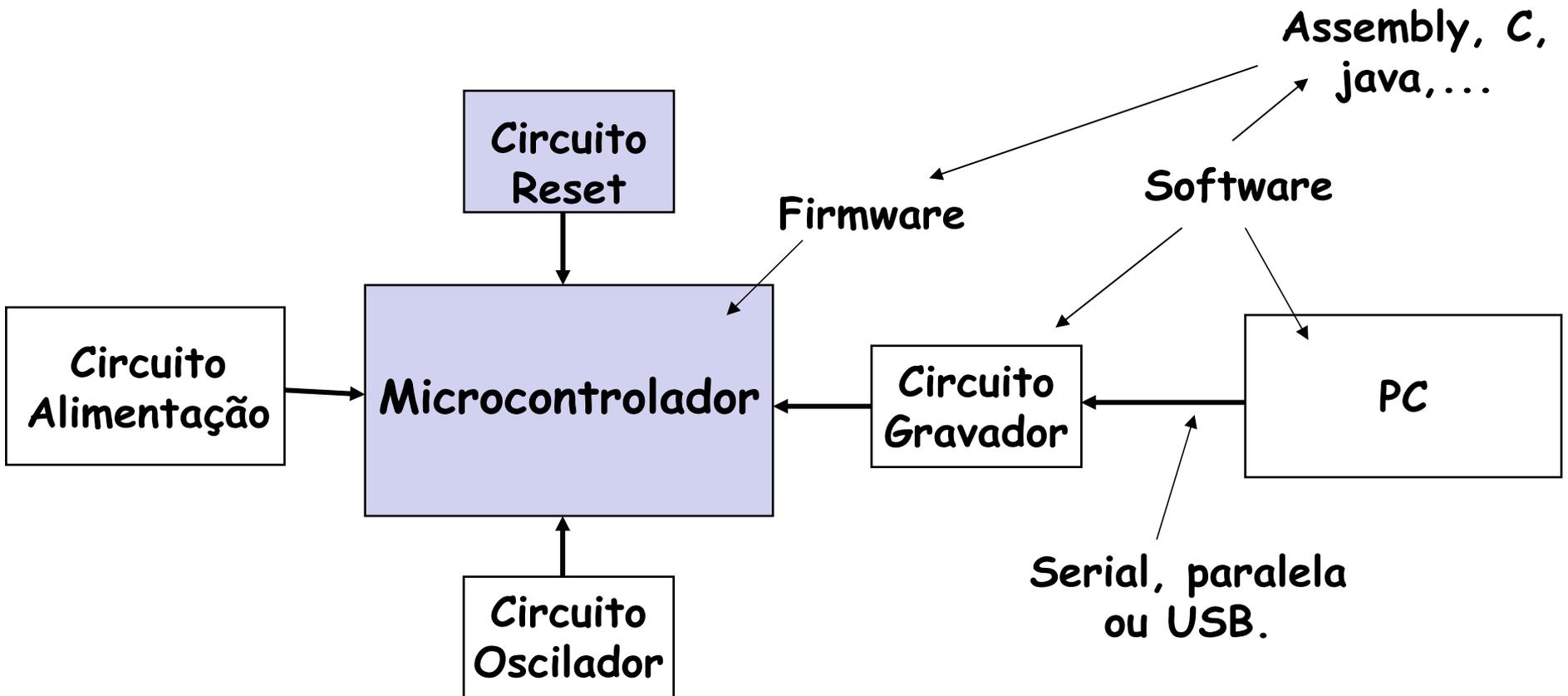
Sistema de Controle e Reset

Jadsonlee da Silva Sá

Jadsonlee.sa@univasf.edu.br

www.univasf.edu.br/~jadsonlee.sa

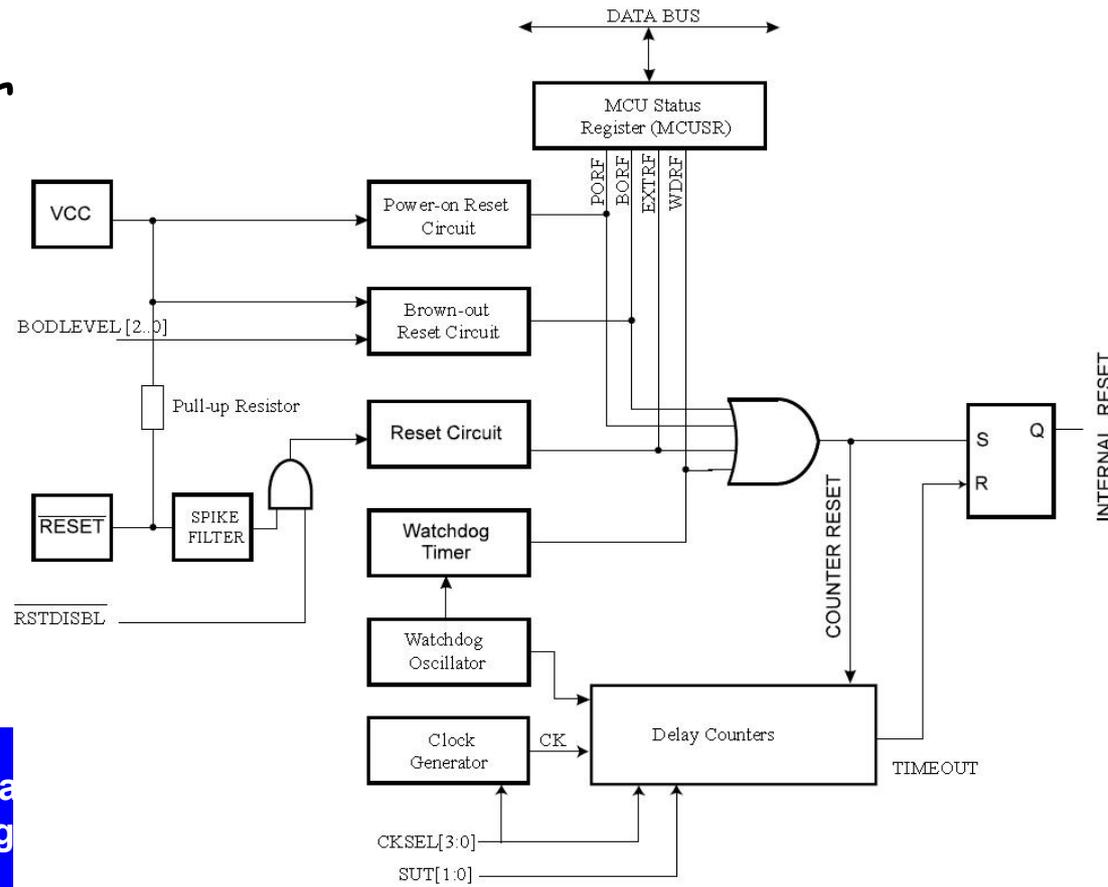
Introdução aos Sistemas Microcontrolados



Fontes de Reset

✓ Há quatro fontes de reset:

- Power-on reset;
- External reset;
- Watchdog system r
- Brown-out reset.



Fontes de Reset

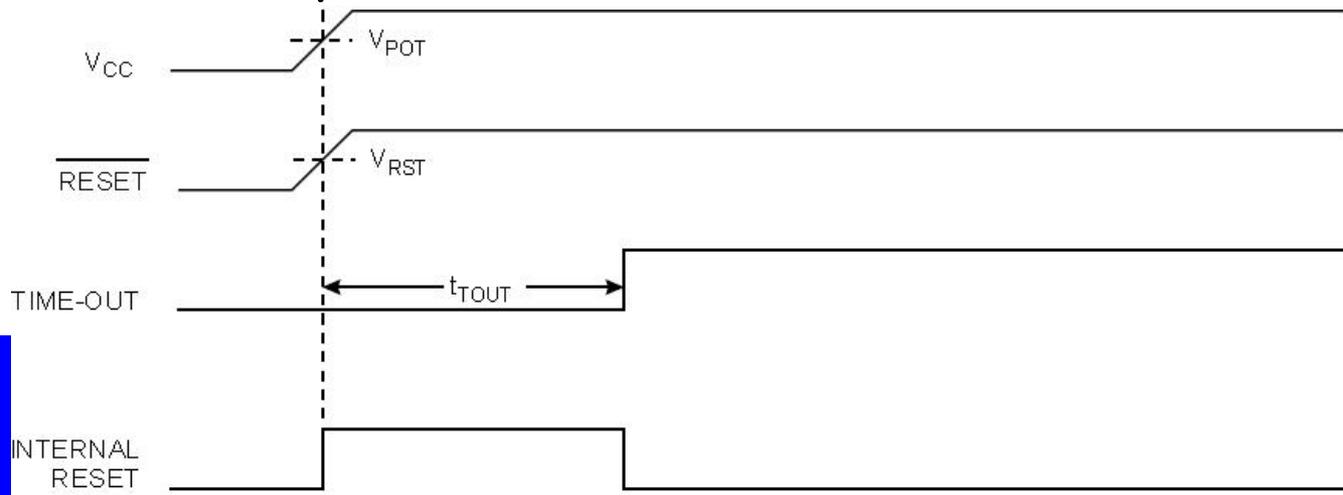
✓ Power-on reset (POR).

- Reset $\rightarrow V_{cc} < V_{pot}$ (Power-on Reset threshold).
- Um pulso POR é gerado por um circuito de detecção interno ao chip, quando V_{cc} estiver abaixo do nível detectado.
- O circuito POR pode ser usado para gatilhar o start-up reset, assim como detectar um defeito na tensão da fonte.

Fontes de Reset

✓ Power-on reset (POR).

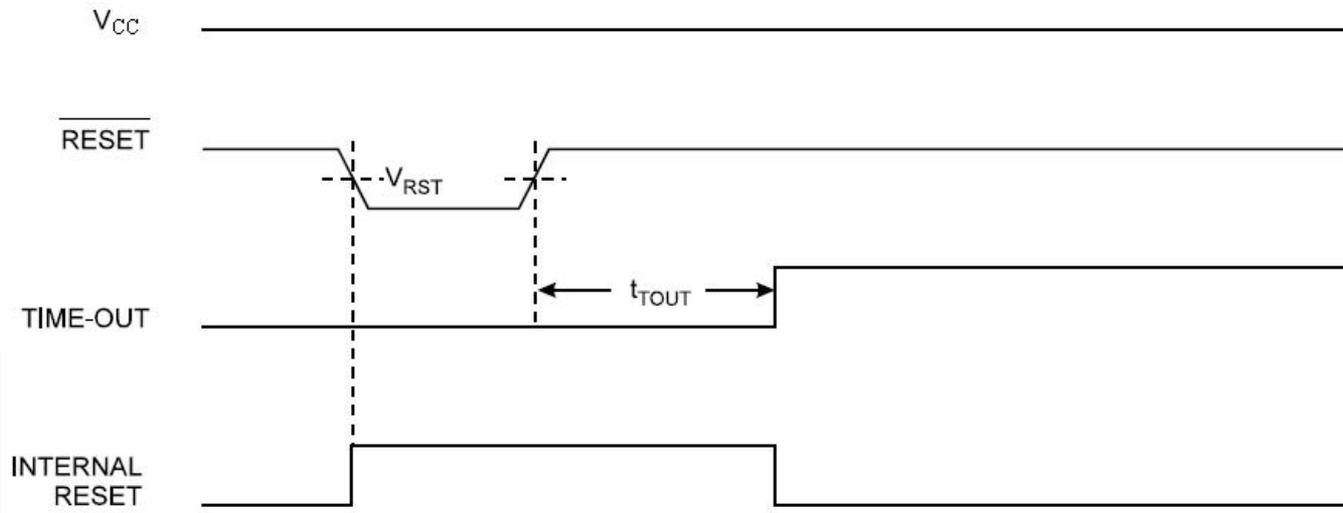
- O circuito POR garante que o dispositivo será resetado no power-on.
- Ao atingir o limite de tensão V_{pot} , um contador de atraso é chamado, o qual determina o tempo em que o dispositivo é mantido em reset após $V_{CC} = V_{pot}$.
- Caso $V_{CC} < V_{pot}$, o sinal de reset é ativado.



Fontes de Reset

✓ External reset.

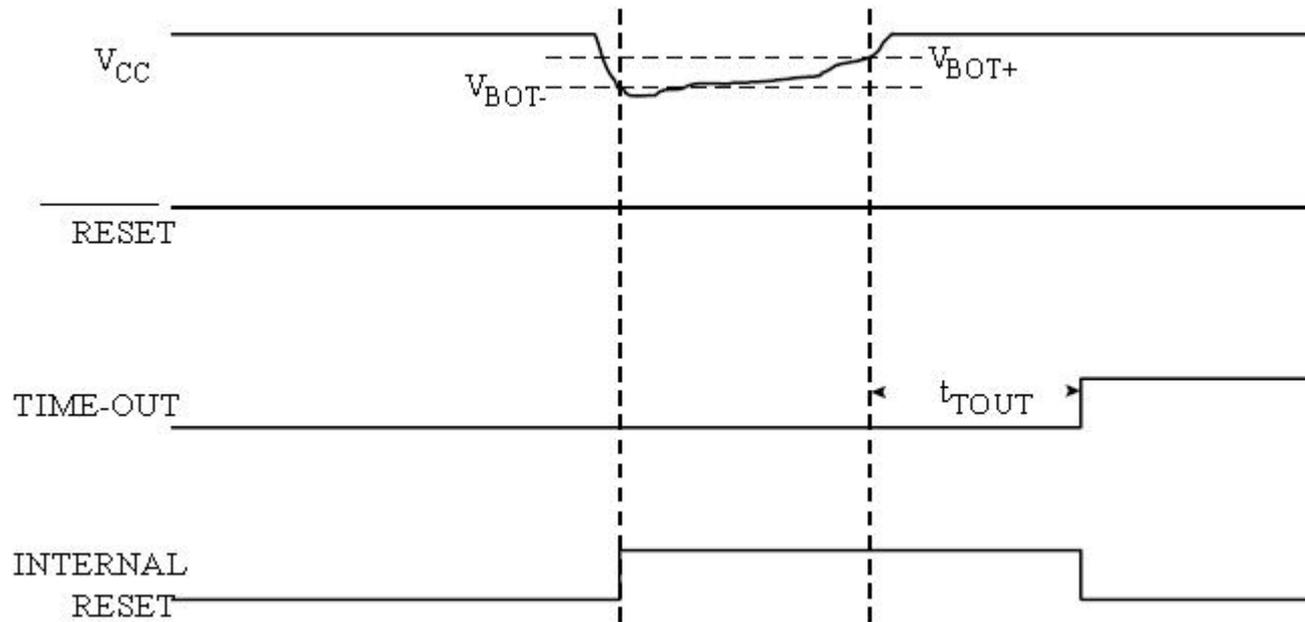
- É gerado por um nível baixo no pino /RESET.
- Quando o sinal aplicado no pino /RESET atinge o valor V_{rst} , o contador de atraso iniciará o uC após o período de time-out expirar.
- O reset externo pode ser desabilitado pelo fuse RSTDISBL.



Fontes de Reset

✓ Brown-out reset.

- O uC possui um circuito BOD (*Brown-out Detect*).

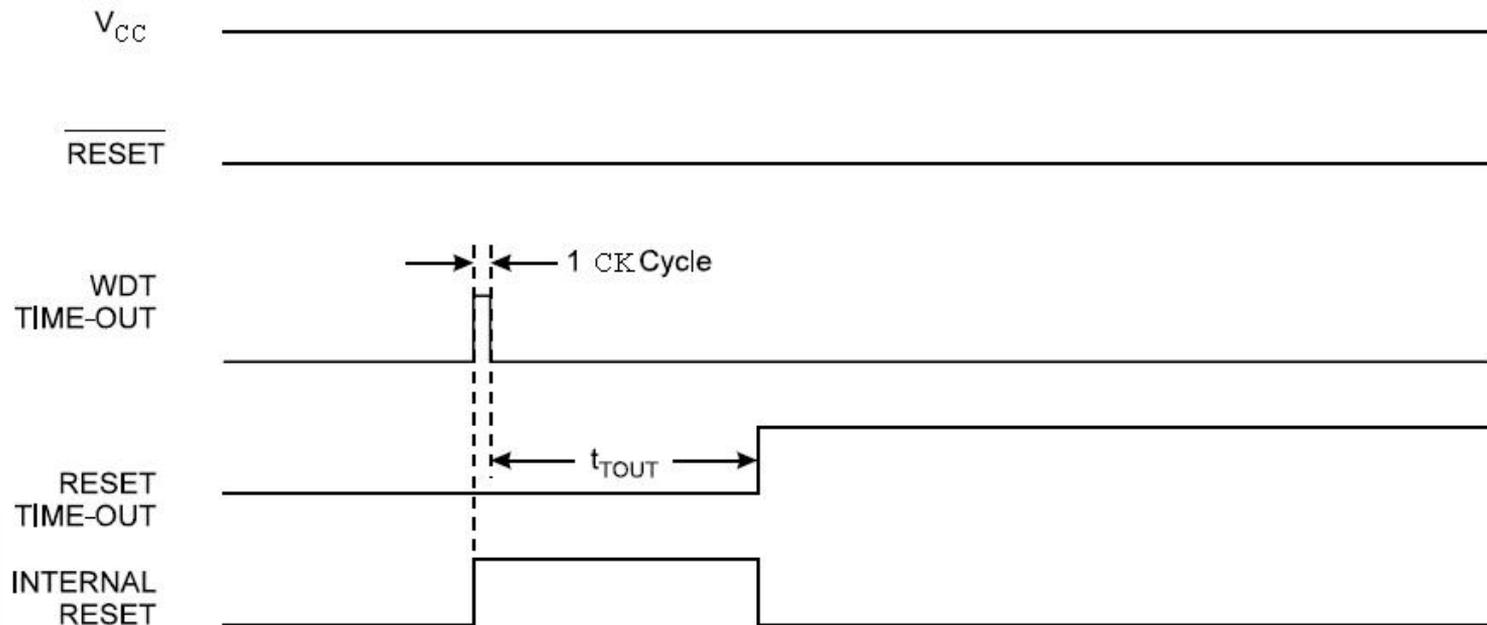


- VBOT é selecionado pelos fuses BODLEVEL.

Fontes de Reset

✓ Watchdog system reset.

- Quando o relógio do watchdog expira, um pulso de reset será gerado. Na borda de descida desse pulso, o temporizador de atraso começa a marcar o período time-out (t_{TOUT}).



Portas de E/S

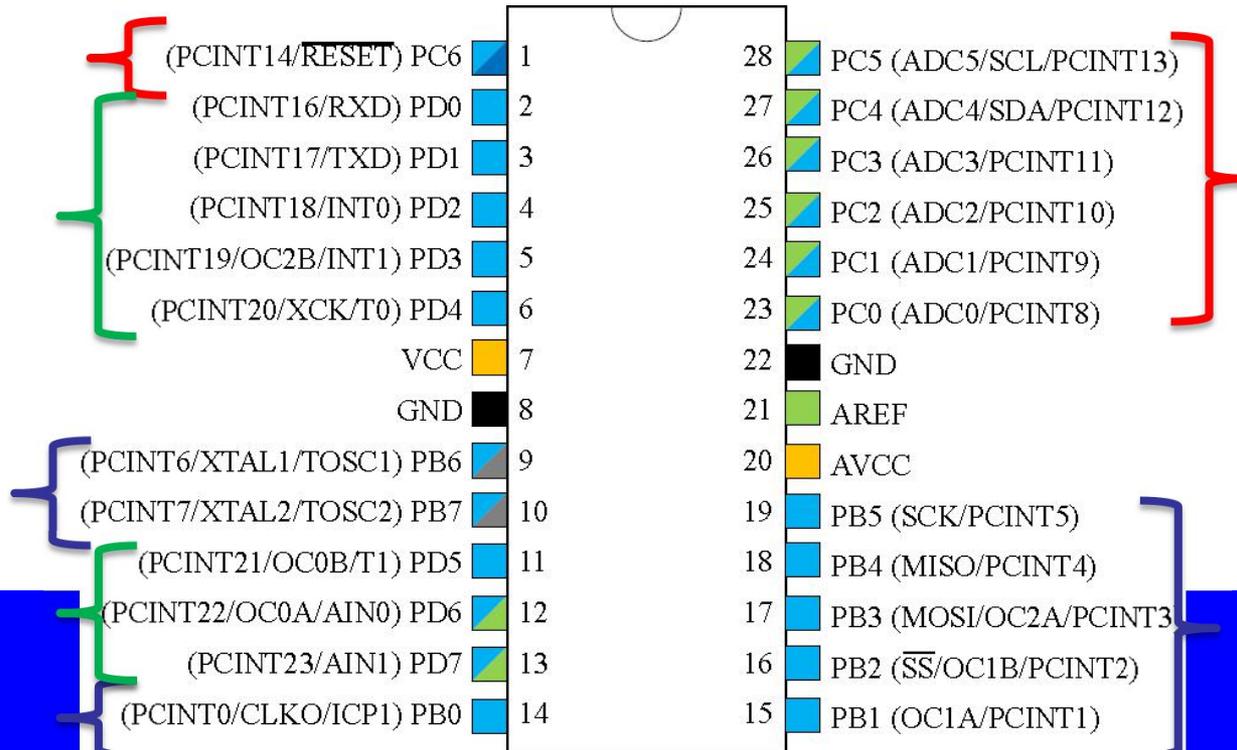
Jadsonlee da Silva Sá

Jadsonlee.sa@univasf.edu.br

www.univasf.edu.br/~jadsonlee.sa

Introdução

- ✓ Há três portas de entrada/saída (E/S).
 - Portas B, C e D.
- ✓ Os pinos das portas podem ser usados como E/S digital ou outras funcionalidades dos periféricos.

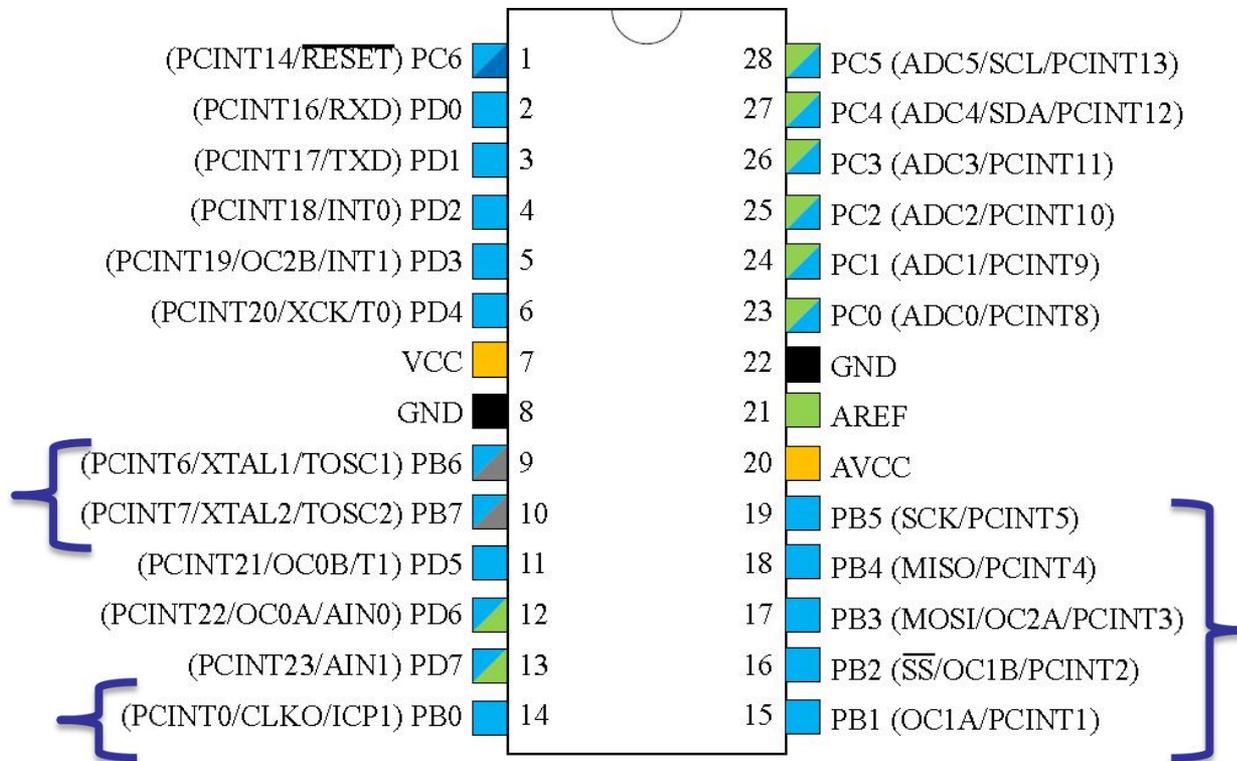


Introdução

- ✓ Há três registradores associados a cada porta, sendo x igual a B, C ou D:
 - **DDR_x** - Indica a direção da porta (E ou S) - Pode-se ler e escrever;
 - **PORT_x** - Contém os valores dos pinos de saída - Pode-se ler e escrever;
 - **PIN_x** - Contém os valores dos pinos de entrada - Pode-se apenas ler.

Porta B

Há 8 pinos de E/S.



Portas B

✓ Registrador DDRB - Direção dos pinos da porta.

Bit	7	6	5	4	3	2	1	0
	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Access	R/W							
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DDRBn: Port B Data Direction [n = 7:0]

- Cada pino possui um bit relacionado.
 - Ex.: O bit DDRB0 indica a direção (E ou S) do pino PBO (pino 14).
 - '1' → Configura o respectivo pino como saída.
 - '0' → Configura o respectivo pino como entrada.
- Use as instruções **SBI** e **CBI** para alterar a direção individual dos pinos.

Portas B

✓ Registradores PORTB – Valor dos pinos (saída).

Bit	7	6	5	4	3	2	1	0
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
Access	R/W							
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – PORTBn: Port B Data [n = 0:7]

✓ Registradores PINB – Valor dos pinos (entrada).

Bit	7	6	5	4	3	2	1	0
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
Access	R/W							
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – PINBn: Port B Input Pins Address [n = 7:0]

Writing to the pin register provides toggle functionality for IO. Refer to [Toggling the Pin](#).

Portas B

- ✓ Exemplo: configurar o pino PB5 como saída e escrever '1' nesse pino.

```
ldi r16, 0b00100000  
out DDRB, r16
```

← Configura o pino 5 da porta B como saída.

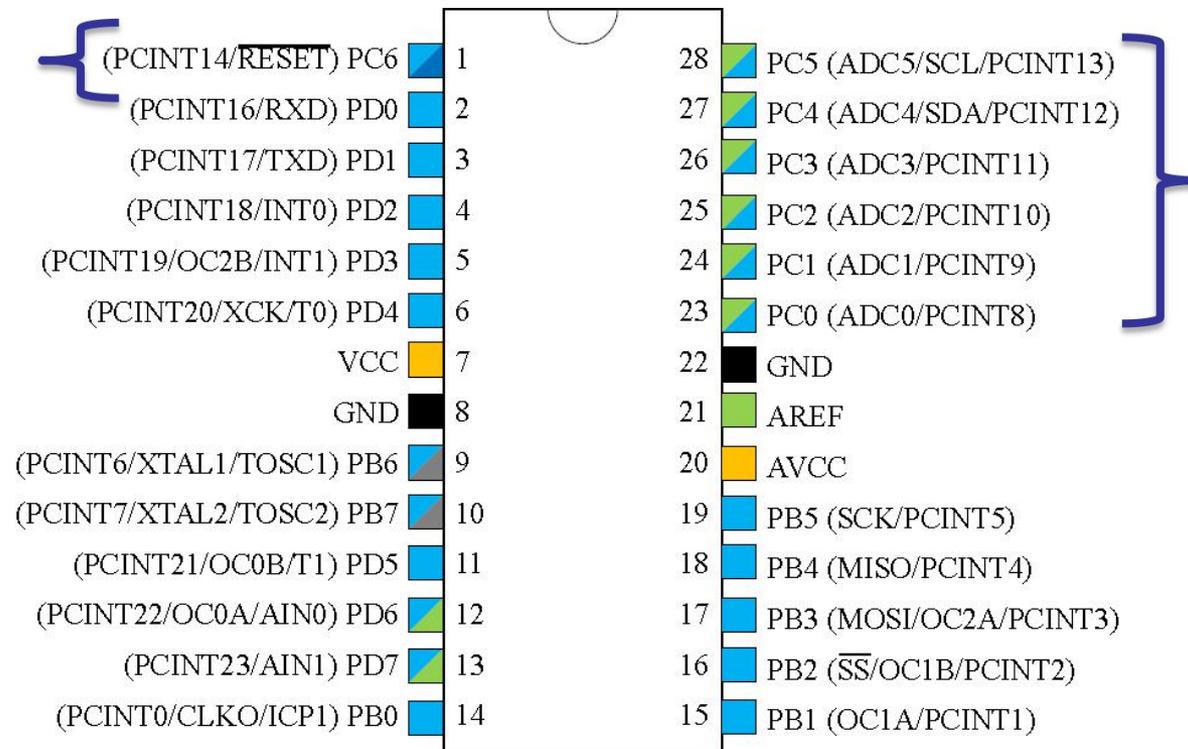
```
ldi r16, 0b00100000  
out PORTB, r16
```

← PB5 = 1.

DDRB = 0b00100000; ou DDRB = 0x20; ou
PORTB = 0b00100000; ou PORTB=0x20;

Porta C

Há 7 pinos de E/S.



Portas C

✓ Registrador DDRC - Direção dos pinos da porta.

Bit	7	6	5	4	3	2	1	0
		DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
Access		R/W						
Reset		0	0	0	0	0	0	0

Bits 6:0 – DDRCn: Port C Data Direction [n = 6:0]

- Cada pino possui um bit relacionado.
 - Ex.: O bit DDRC6 indica a direção (E ou S) do pino PC6 (pino 1).
 - '1' → Configura o respectivo pino como saída.
 - '0' → Configura o respectivo pino como entrada.
- Use as instruções **SBI** e **CBI** para alterar a direção individual dos pinos.

Portas C

✓ Registradores PORTC – Valor dos pinos (saída).

Bit	7	6	5	4	3	2	1	0
		PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
Access		R/W						
Reset		0	0	0	0	0	0	0

Bits 6:0 – PORTCn: Port C Data [n = 6:0]

✓ Registradores PINC – Valor dos pinos (entrada).

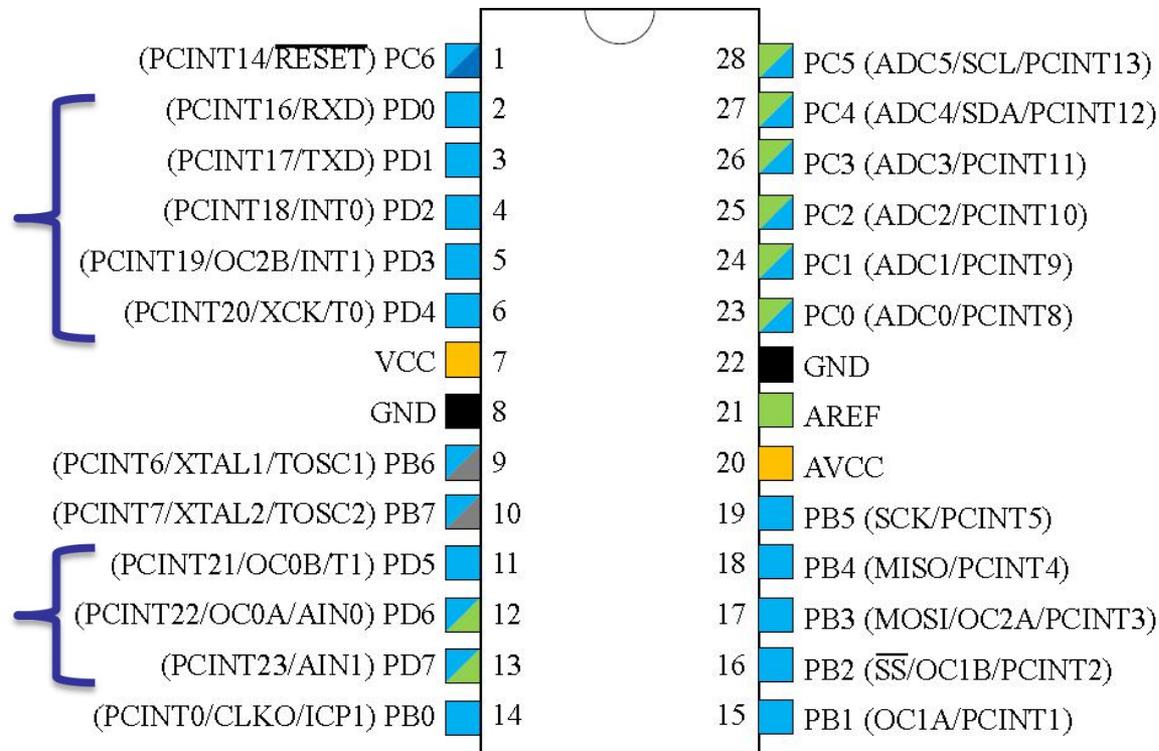
Bit	7	6	5	4	3	2	1	0
		PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
Access		R/W						
Reset		x	x	x	x	x	x	x

Bits 6:0 – PINCn: Port C Input Pins Address [n = 6:0]

Writing to the pin register provides toggle functionality for IO. Refer to [Toggling the Pin](#).

Porta D

Há 8 pinos de E/S.



Portas D

✓ Registrador DDRD - Direção dos pinos da porta.

Bit	7	6	5	4	3	2	1	0
	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Access	R/W							
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DDRDn: Port D Data Direction [n = 7:0]

- Cada pino possui um bit relacionado.
 - Ex.: O bit DDRD5 indica a direção (E ou S) do pino PD5 (pino 11).
 - '1' → Configura o respectivo pino como saída.
 - '0' → Configura o respectivo pino como entrada.
- Use as instruções **SBI** e **CBI** para alterar a direção individual dos pinos.

Portas D

✓ Registradores PORTD - Valor dos pinos (saída).

Bit	7	6	5	4	3	2	1	0
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
Access	R/W							
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – PORTDn: Port D Data [n = 7:0]

✓ Registradores PIND - Valor dos pinos (entrada).

Bit	7	6	5	4	3	2	1	0
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
Access	R/W							
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – PINDn: Port D Input Pins Address [n = 7:0]

Writing to the pin register provides toggle functionality for IO. Refer to [Toggling the Pin](#).

Pinos das Portas

- ✓ Opcionalmente, cada pino, quando configurado como **entrada**, possui resistor pull-up.
- ✓ **Resistor pull-up** → Resistor usado em circuitos lógicos para garantir que as tensões de entrada se ajustem a níveis lógicos esperados - Evita flutuações de tensão.
- ✓ **Habilitar resistor pull-up** → Escreva '1' no respectivo bit de PORTx.
- ✓ **Desabilitar resistor pull-up** → Escreva '0' no respectivo bit de PORTx ou configure o pino como saída.

Pinos das Portas

- ✓ Para desabilitar todos os resistores pull-up, independentemente dos valores de DDRx e PORTx, escreva '0' no bit PUD de MCUCR.

Name: MCUCR

Offset: 0x55

Reset: 0x00

Property: When addressing as I/O Register: address offset is 0x35

Bit	7	6	5	4	3	2	1	0
		BODS	BODSE	PUD			IVSEL	IVCE
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bit 4 – PUD: Pull-up Disable

Pinos das Portas

- ✓ Para alternar (toggling) o valor de um bit em `PORTx`, escreva '1' no respectivo bit de `PINx`, independentemente do valor do respectivo bit de `DDRx`.
- ✓ Use a instrução `SBI` para alternar o valor de um único bit de uma porta.

Exercício

- ✓ Faça um programa em assembly e em C para piscar o LED no pino 19 (PB5) a cada 500 ms. Considere que a frequência de clock é igual a 16 MHz.
- ✓ Faça um programa em assembly e em C que configura os pinos PB0, PB2 e PB4 como saídas, e os pinos PB1, PB3 e PB5 como entradas. Conecte os pinos PB0 a PB1, PB2 a PB3 e PB4 a PB5. A cada 100 ms, alterne o valor dos pinos de saída. Considere que a frequência de clock é igual a 16 MHz.
- Nos códigos em C, use `#include <util/delay.h>` para gerar os delays. Ex.: `_delay_ms(500);`