

2. O ATMEGA328

Neste capítulo, apresentam-se, sucintamente, o microcontrolador ATmega328, suas funcionalidades e a descrição de seu hardware: memórias, pinos, sistema de *clock*, sistema de *reset* e sistema de gerenciamento de energia.

O ATmega328⁵ será abordado por ser um microcontrolador compacto que apresenta a maioria das características da família AVR e uma memória *flash* de maior tamanho comparado aos microcontroladores AVR com o mesmo número de pinos. Outro fator decisivo, é que ele é utilizado atualmente na plataforma de desenvolvimento Arduino, de fácil aquisição. O importante a saber é que, ao se programar esse microcontrolador, os conceitos de programação para qualquer outro da família AVR serão iguais, devido às similaridades entre os componentes. Na maioria dos casos a mudança se dá no uso e nome de alguns registradores e periféricos. Na dúvida, qualquer mudança de hardware e software pode ser resolvida com uma busca ao manual do fabricante.

As principais características do ATmega328 são:

- Microcontrolador de baixa potência, com arquitetura RISC avançada.
- 131 instruções, a maior parte executada em 1 ou 2 ciclos de *clock* (poucas em 3 ou 4 ciclos).
- 32 registradores de trabalho de propósito geral (8 bits cada). Alguns trabalham em par para endereçamentos de 16 bits.
- Operação de até 20 MIPS a 20 MHz.
- Multiplicação por hardware em 2 ciclos de *clock*.
- 32 kbytes de memória de programa *flash* de auto programação *In-System* (8 k, 16 k, nos respectivos ATmega88 e ATmega168).
- 1 kbytes de memória EEPROM.
- 2 kbytes de memória SRAM.

⁵ Os ATmega48/88/168/328 são idênticos com diferenças sutis, principalmente na quantidade de memória disponível. A Atmel disponibiliza um único manual para eles.

- Ciclos de escrita e apagamento: memória *flash* 10 mil vezes, EEPROM 100 mil vezes.
- Seção opcional para código de *boot* para programação *In-System* por *boot loader*⁶.
- Bits de bloqueio para proteção contra a cópia do firmware.
- Possui os seguintes periféricos:
 - ➔ 23 entradas e saídas (I/Os) programáveis.
 - ➔ 2 Temporizadores/Contadores de 8 bits com *Prescaler* separado, com modo de comparação.
 - ➔ 1 Temporizador/Contador de 16 bits com *Prescaler* separado, com modo de comparação e captura.
 - ➔ Contador de tempo real (com um cristal externo de 32,768 kHz conta precisamente 1 s).
 - ➔ 6 canais PWM.
 - ➔ 8 canais AD com resolução de 10 bits na versão TQFP (*Thin profile plastic Quad Flat Package*) e 6 canais na versão PDIP (*Plastic Dual Inline Package*).
 - ➔ Interface serial para dois fios orientada a byte (TWI), compatível com o protocolo I2C.
 - ➔ Interface serial USART.
 - ➔ Interface serial SPI *Master/Slave*.
 - ➔ *Watchdog Timer* com oscilador interno separado.
 - ➔ 1 comparador analógico.
- Características especiais:
 - ➔ *Power-on Reset* e detecção *Brown-out* programável.
 - ➔ Oscilador interno RC (não há a necessidade do uso de cristal externo ou de outra fonte de *clock*).
 - ➔ Fontes de interrupções internas e externas (em todos os pinos de I/O).
 - ➔ Saída de *clock* em um pino de I/O (PB0).
 - ➔ *Pull-up* habilitáveis em todos os pinos de I/O
 - ➔ 6 modos de *Sleep*: *Idle*, Redução de ruído do ADC, *Power-down*, *Power-save*, *Standby* e *Extended Standby*.
 - ➔ Medição de temperatura do encapsulamento.
- Tensão de operação: 1,8 - 5,5 V.
- Consumo de corrente a 1 MHz (1,8 V, 25 °C): modo ativo = 0,2 mA e modo *Power-down* = 0,1 µA.

⁶ **Boot loader** é um pequeno programa que pode ser escrito no início ou no final da memória de programa e serve para que o microcontrolador gerencie a gravação de sua memória. Para tal, é necessário uma interface de comunicação externa com o software de desenvolvimento. No AVR o espaço destinado ao *boot loader* fica no final da memória de programa e a comunicação se dá através de um dos seus periféricos de comunicação, a USART.

Na fig. 2.1, é apresentado o diagrama esquemático da estrutura interna de um ATmega328. O barramento de dados é de 8 bits, caracterizando o número de bits do microcontrolador. As instruções do ATmega são de 16 ou 32 bits (a maioria é de 16 bits). Assim, cada instrução consome 2 ou 4 bytes na memória de programa (um byte par e um ímpar). O acesso às posições de memória, dado pelo contador de programa (*Program Counter* - PC), é realizada de dois em dois bytes, começando sempre por uma posição par. Portanto, o barramento de endereços deve ser capaz de endereçar sempre posições pares da memória de programa. Logo, o bit menos significativo do barramento de endereços pode ser desprezado. Desta forma, para a memória de 32 kbytes do Atmega328 são necessários 14 bits de endereçamento ($2^{14} = 16.384$ endereços) e não 15 ($2^{15} = 32.768$ endereços). Na fig. 2.1, é ilustrada a arquitetura Harvard empregada pelo ATmega328 (barramento de instruções e dados separados).

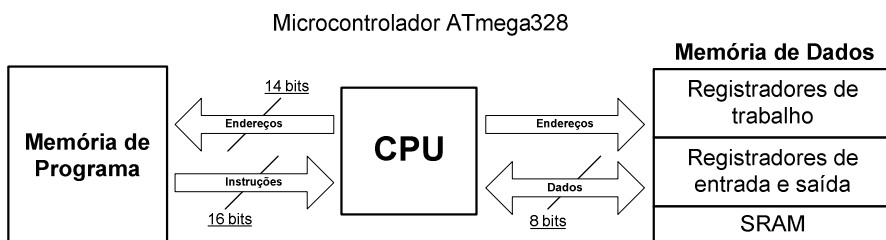


Fig. 2.1 – Diagrama esquemático da estrutura de um microcontrolador ATmega328.

O núcleo AVR utiliza 32 registradores de trabalho diretamente conectados à Unidade Lógico-Aritmética (ALU – *Arithmetic Logic Unit*), permitindo que dois registradores independentes sejam acessados com uma simples instrução em um único ciclo de *clock*. Seis desses registradores podem ser usados como registradores de endereçamento indireto de 16 bits (ponteiros para o acesso de dados), denominados X, Y e Z. Na fig. 2.2, são ilustrados esses registradores e seus respectivos endereços na memória de dados. Os 32 registradores não podem ser

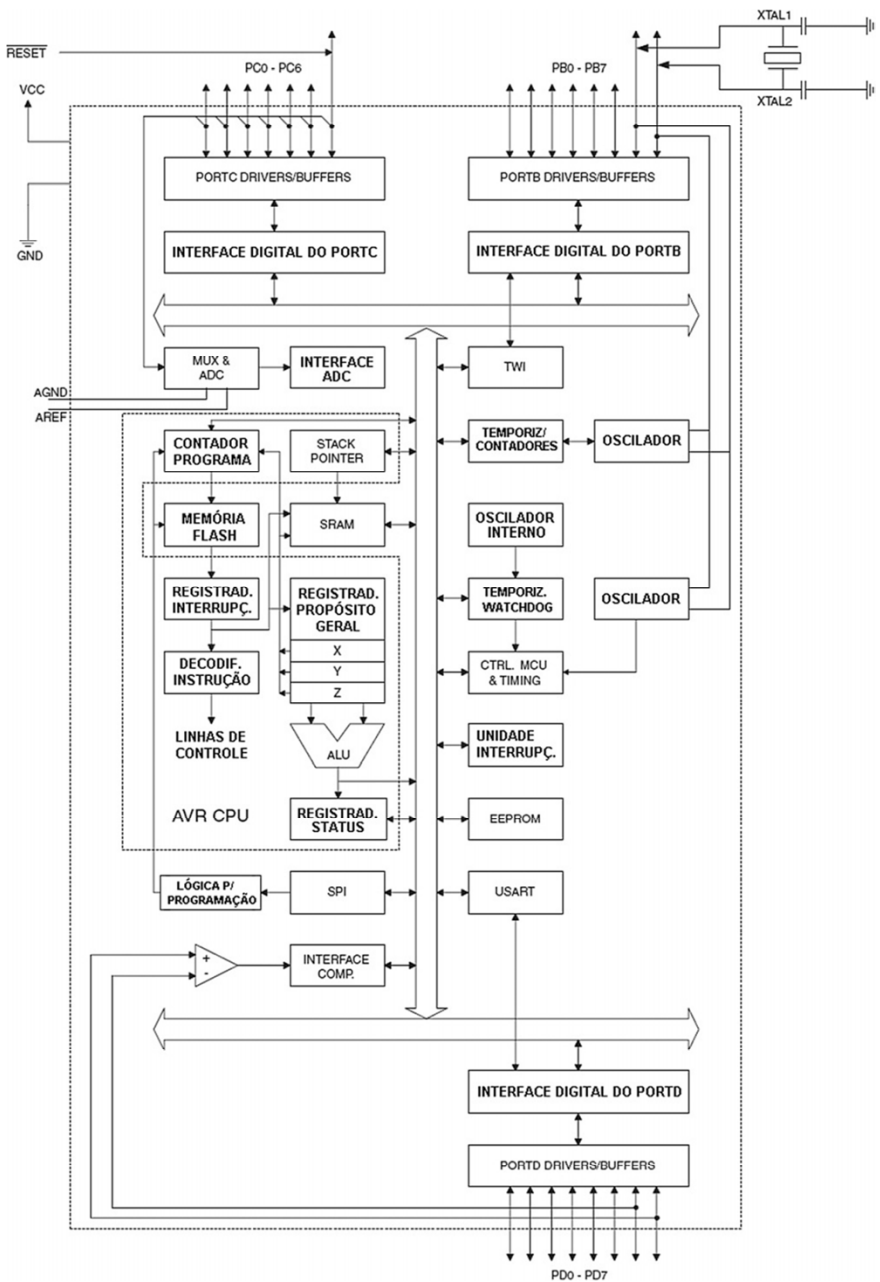


Fig. 2.3 – Diagrama em blocos do ATmega328.

A arquitetura do AVR permite a busca e execução de instruções em paralelo devido ao emprego da técnica de *Pipeline*. Assim, o desempenho alcançado pode chegar a 1 MIPS por MHz. O diagrama de tempo da busca e execução de instruções, com referência ao *clock* da CPU, é apresentado na fig. 2.4. Observa-se que uma instrução é executada enquanto a próxima é lida. O ATmega emprega um *pipeline* de 2 estágios, dessa forma uma instrução é lida e decodificada dentro de um mesmo ciclo de *clock*, enquanto a instrução anterior é executada.

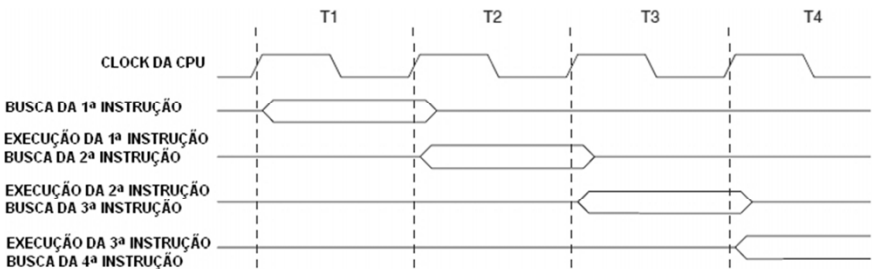


Fig. 2.4 – Diagrama de tempo para a busca e execução de instruções no ATmega.

2.1 AS MEMÓRIAS

Um diagrama da memória de dados e memória SRAM do ATmega328 pode ser visto na fig. 2.5. A organização da memória é linear, começando no endereço 0 e indo até o endereço 2303 (0x8FF). Destas posições de memória, 32 pertencem aos registradores de uso geral (0x000 até 0x01F, fig. 2.2), 64 aos registradores de entrada e saída (0x020 até 0x05F) e 2048 bytes pertencem à memória SRAM (0x060 até 0x8FF), cujos 160 primeiros endereços são empregados para a extensão dos registradores de entrada e saída. Isso é necessário porque o número de periféricos no ATmega328 é superior ao que pode ser suportado pelos 64 registradores originais (dos primeiros ATmegs) e, também, para permitir o acréscimo de futuras

funcionalidades. Desse modo, os engenheiros da Atmel utilizaram a SRAM⁷ para aumentar o número de registradores de I/O sem a necessidade de alterações profundas no projeto do chip.

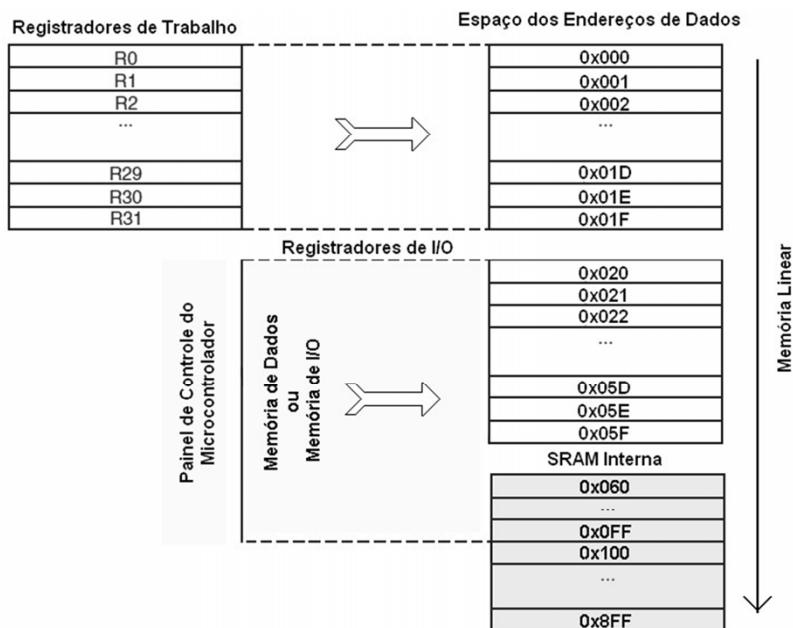


Fig. 2.5 – Memória de dados e memória SRAM do ATmega328.

Os registradores de I/O são o painel de controle do microcontrolador, pois todas as configurações de trabalho, incluindo acesso às entradas e saídas, se encontram nessa parte da memória. É fundamental compreender que são esses registradores que dão acesso às funcionalidades do microcontrolador, pois possuem todas as informações referentes aos periféricos e ao processamento da CPU. É com esses registradores que o programador terá que se familiarizar para trabalhar com os periféricos (as ‘chaves’ que ligam e desligam tudo). Os registradores

⁷ Literalmente, a memória SRAM não pode ser denominada memória de dados ou memória de I/O da CPU, pois ela é projetada para armazenar variáveis temporárias do programa e não os registradores de I/O que possuem endereço fixo.

são apresentados na tab. 2.1 (observar seu endereçamento na fig. 2.5) e, dado sua importância, os mesmos serão vistos com detalhes posteriormente (um resumo é incluído no apêndice G).

Tab. 2.1: Registradores de entrada e saída da memória de dados (painel de controle do microcontrolador)*.

End.	Nome	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x23	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
0x24	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
0x25	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
0x26	PINC	-	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
0x27	DDRC	-	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
0x28	PORTC	-	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
0x29	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
0x2A	DDRD	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
0x2B	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
0x35	TIFR0	-	-	-	-	-	OCF0B	OCF0A	TOV0
0x36	TIFR1	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1
0x37	TIFR2	-	-	-	-	-	OCF2B	OCF2A	TOV2
0x3B	PCIFR	-	-	-	-	-	PCIF2	PCIF1	PCIF0
0x3C	EIFR	-	-	-	-	-	-	INTF1	INTF0
0x3D	EIMSK	-	-	-	-	-	-	INT1	INT0
0x3E	GPOR0	Registrador de I/O de propósito geral 0							
0x3F	EECR	-	-	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE
0x40	EEDR	Registrador de dados da EEPROM							
0x41	EEARL	Byte menor do registrador de endereço da EEPROM							
0x42	EEARH	Byte maior do registrador de endereço da EEPROM							
0x43	GTCCR	TSM	-	-	-	-	-	PSRASY	PSRSYNC
0x44	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00
0x45	TCCR0B	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00
0x46	TCNT0	Temporizador/Contador 0 (8 bits) – Registrador de contagem							
0x47	OCR0A	Registrador de comparação de saída A do Temporizador/Contador 0							
0x48	OCR0B	Registrador de comparação de saída B do Temporizador/Contador 0							
0x4A	GPOR1	Registrador de I/O de propósito geral 1							
0x4B	GPOR2	Registrador de I/O de propósito geral 2							
0x4C	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
0x4D	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X
0x4E	SPDR	Registrador de dados da SPI							
0x50	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
0x53	SMCR	-	-	-	-	SM2	SM1	SM0	SE
0x54	MCUSR	-	-	-	-	WDRF	BORF	EXTRF	PORF
0x55	MCUCR	-	BODS	BODSE	PUD	-	-	IVSEL	IVCE
0x57	SPMCSR	SPMIE	RWWSB	-	RWWSRE	BLBSET	PGWRT	PGERS	SELFPRGEN
0x5D	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
0x5E	SPH	-	-	-	-	-	SP10	SP9	SP8
0x5F	SREG	I	T	H	S	V	N	Z	C
0x60	WDTCR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0
0x61	CLKPR	CLKPCE	-	-	-	CLKPS3	CLKPS2	CLKPS1	CLKPS0
0x64	PRR	PRTW1	PRTIM2	PRTIM0	-	PRTIM1	PRSPI	PRUSART0	PRADC
0x66	OSCCAL	Registrador de calibração do oscilador							
0x68	PCICR	-	-	-	-	-	PCIE2	PCIE1	PCIE0
0x69	EICRA	-	-	-	-	ISC11	ISC10	ISC01	ISC00
0x6B	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
0x6C	PCMSK1	-	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
0x6D	PCMSK2	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16
0x6E	TIMSK0	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0

End.	Nome	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x6F	TIMSK1	-	-	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1
0x70	TIMSK2	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2
0x78	ADCL	Registrador de dados do ADC, byte menor							
0x79	ADCH	Registrador de dados do ADC, byte maior							
0x7A	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
0x7B	ADCSRB	-	ACME	-	-	-	ADTS2	ADTS1	ADTS0
0x7C	ADMUX	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0
0x7E	DIDR0	-	-	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
0x7F	DIDR1	-	-	-	-	-	-	AIN1D	AIN0D
0x80	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10
0x81	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
0x82	TCCR1C	FOC1A	FOC1B	-	-	-	-	-	-
0x84	TCNT1L	Temporizador/Contador 1 (16 bits) – Registrador de contagem do byte menor							
0x85	TCNT1H	Temporizador/Contador 1 (16 bits) – Registrador de contagem do byte maior							
0x86	ICR1L	Temporizador/Contador 1 (16 bits) – Registrador de captura de entrada, byte menor							
0x87	ICR1H	Temporizador/Contador 1 (16 bits) – Registrador de captura de entrada, byte maior							
0x88	OCR1AL	Temporizador/Contador 1 (16 bits) – Registrador da saída de comparação A, byte menor							
0x89	OCR1AH	Temporizador/Contador 1 (16 bits) – Registrador da saída de comparação A, byte maior							
0x8A	OCR1BL	Temporizador/Contador 1 (16 bits) – Registrador da saída de comparação B, byte menor							
0x8B	OCR1BH	Temporizador/Contador 1 (16 bits) – Registrador da saída de comparação B, byte maior							
0xB0	TCCR2A	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20
0xB1	TCCR2B	FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20
0xB2	TCNT2	Temporizador/Contador 2 (8 bits) – Registrador de contagem							
0xB3	OCR2A	Temporizador/Contador 2 (8 bits) – Registrador da saída de comparação A							
0xB4	OCR2B	Temporizador/Contador 2 (8 bits) – Registrador da saída de comparação B							
0xB6	ASSR	-	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB
0xB8	TWBR	Registrador da taxa de bits da interface serial a dois fios - TWI							
0xB9	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0
0xBA	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
0xBB	TWDR	Registrador de dados da interface serial a dois fios - TWI							
0xBC	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
0xBD	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	-
0xC0	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0
0xC1	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80
0xC2	UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01 / UDORD0	UCSZ00 / UCPHA0	UCPOL0
0xC4	UBRR0L	Registrador da taxa de transmissão da USART, byte menor							
0xC5	UBRR0H	-	-	-	-	Registrador da taxa de transmissão da USART, byte maior			
0xC6	UDR0	Registrador I/O de dados da USART							

* Os endereços que não aparecem são reservados e foram suprimidos para não ocupar espaço na tabela, em negrito os que os seguem.

A organização da memória de programa pode ser vista na fig. 2.6. Cada endereço da memória possui 2 bytes, pois as instruções do AVR são de 16 ou 32 bits. Dessa forma, a memória possui um total de 16384 endereços (de 0x0000 até 0x3FFF), correspondendo a 32 kbytes de memória. Existe uma seção específica para carregar o *boot loader*, que pode ou não ser utilizada para esse fim. A memória *flash* suporta, no mínimo, 10 mil ciclos de escrita e apagamento.

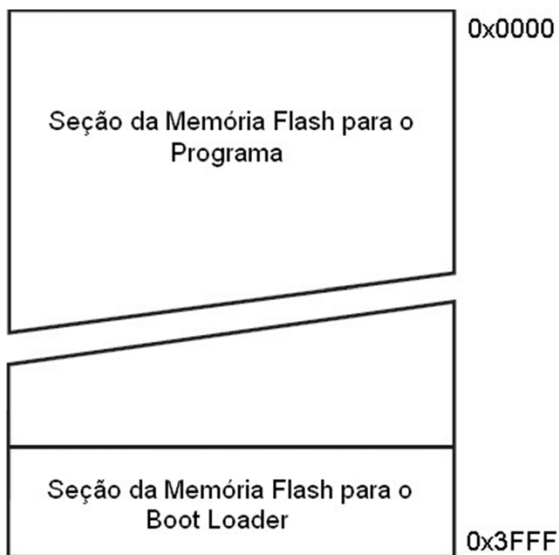


Fig. 2.6 – Organização da memória de programa.

A memória EEPROM é de 1 kbytes e é organizada separadamente. Cada byte individual pode ser lido ou escrito e a memória suporta, no mínimo, 100 mil ciclos de escrita e apagamento.

2.1.1 O REGISTRADOR DE STATUS - SREG

Um dos registradores mais importantes do painel de controle do microcontrolador é o SREG (*Status Register*), que indica, através de bits individuais, o estado das operações lógicas e aritméticas da CPU e permite habilitar ou não as interrupções (chave geral). Seus bits são empregados na programação para tomadas de decisões e na realização de operações lógico-aritméticas. O detalhamento do registrador SREG é apresentado a seguir.

Bit	7	6	5	4	3	2	1	0
SREG	I	T	H	S	V	N	Z	C
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

Bit 7 – I – Global Interrupt Enable

Esse bit é a chave geral para habilitar as interrupções. Cada interrupção individual possui seus registradores de controle. O bit **I** é limpo quando uma interrupção ocorre (impedindo que outras ocorram simultaneamente) e volta a ser ativo quando se termina o tratamento da interrupção (instrução RETI).

Bit 6 – T – Bit Copy Storage

Serve para copiar o valor de um bit de um registrador ou escrever o valor de um bit em um registrador (instruções BLD e BST).

Bit 5 – H – Half Carry Flag

Indica quando um *Carry* auxiliar (em um *nibble*⁸) ocorreu em alguma operação aritmética. Útil em aritmética BCD (Binário Codificado em Decimal).

Bit 4 – S – Sign Bit, $S = N \oplus V$

O bit **S** é o resultado de uma operação ou-exclusivo entre o bit de sinalização negativo **N** e o bit de sinalização de estouro do complemento de dois **V**.

Bit 3 – V – Two's Complement Overflow Flag

O bit de sinalização de estouro do complemento de dois, ajuda na aritmética em complemento de dois.

Bit 2 – N – Negative Flag

O bit de sinalização negativo indica quando uma operação aritmética ou lógica resulta em um valor negativo.

Bit 1 – Z – Zero Flag

O bit de sinalização zero indica quando uma operação aritmética ou lógica resulta em zero.

Bit 0 – C – Carry Flag

O bit de sinalização de *Carry* indica quando houve um estouro numa operação aritmética.

⁸ Um *nibble* = 4 bits, dois *nibbles* = 1 byte.

2.1.2 O STACK POINTER

O *Stack Pointer* (SP, ponteiro de pilha) é um registrador que armazena um endereço correspondente a uma posição da memória RAM, a qual é utilizada na forma de uma pilha para armazenagem temporária de dados: variáveis locais e endereços de retorno após chamadas de sub-rotinas e interrupções. Em resumo, o SP indica a posição onde um determinado dado foi armazenado na pilha alocada na RAM, esse conceito é ilustrado na fig. 2.7.

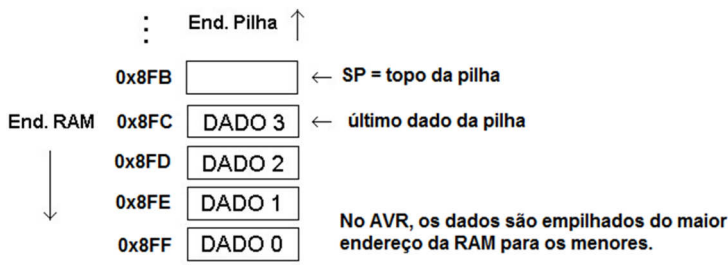


Fig. 2.7 – *Stack Pointer*.

O endereço do SP é pós-decrementado toda vez que um dado é colocado na pilha. Assim, a cada novo dado colocado na pilha, o endereço do SP apresenta um valor menor que o anterior. Da mesma forma, quando um dado é retirado da pilha, o endereço do SP é pré-incrementado, sempre apontando uma posição acima do último dado válido da pilha. Isso implica que o comando PUSH, que coloca um dado na pilha, diminui o valor do SP e o comando POP, que retira um dado da pilha, o incrementa. Na tab. 2.2, são apresentadas as instruções do ATmega que afetam o SP.

Dado o funcionamento do SP, na sua inicialização ele deve apontar para o endereço final da RAM, no caso do ATmega328 o endereço é 0x8FF (seu valor *default* após a energização). Entretanto, existem microcontroladores da família ATmega que precisam ter o SP inicializado pelo programador. Nesse caso, quando o programa é escrito em *assembly* a

inicialização deve ser feita pelo programador de forma explícita (ver os exemplos de programação *assembly* no capítulo 5). Essa inicialização é feita automaticamente quando o programa é escrito na linguagem C, pois o compilador se encarrega da tarefa.

Tab. 2.2: Instruções do ATmega que afetam o *Stack Pointer*.

Instrução	<i>Stack Pointer</i>	Descrição
PUSH	Decrementa 1	Um dado é colocado na pilha (1 byte).
CALL ICALL RCALL	Decrementa 2	O endereço de retorno é colocado na pilha quando uma chamada de sub-rotina ou interrupção acontece (o endereço possui 2 bytes).
POP	Incrementa 1	O dado do topo da pilha é retirado (1 byte).
RET RETI	Incrementa 2	O endereço de retorno é retirado da pilha quando se retorna de uma sub-rotina ou interrupção (o endereço possui 2 bytes).

Como o SP armazena um endereço da RAM, ele deve ter um número de bits suficiente para isso. Como o ATmega possui registradores de 8 bits, são necessários dois registradores para o SP, um armazena a parte baixa do endereço (*SP Low*) e outro armazena a parte alta do endereço (*SP High*), resultando num registrador de 12 bits (os 4 bits mais significativos do SPH não são utilizados). Abaixo são apresentados os registradores do SP (ver os endereços 0x5D e 0x5E da tab. 2.1).

Bit	15	14	13	12	11	10	9	8
SPH	-	-	-	-	SP11	SP10	SP9	SP8
SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
Bit	7	6	5	4	3	2	1	0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	-	-	-	-	1	0	0	0
	1	1	1	1	1	1	1	1

Fig. 2.8- Detalhamento dos registradores do *Stack Pointer* (valor inicial 0x8FF).

2.2 DESCRIÇÃO DOS PINOS

Na fig. 2.9, os nomes dos pinos do ATmega328 são apresentados para os encapsulamentos PDIP e TQFP. Cada pino acumula várias funções selecionáveis, as siglas nos pinos resumem as funcionalidades desses e serão abordadas no momento adequado. A tab. 2.3 contém a descrição sucinta dos referidos pinos.

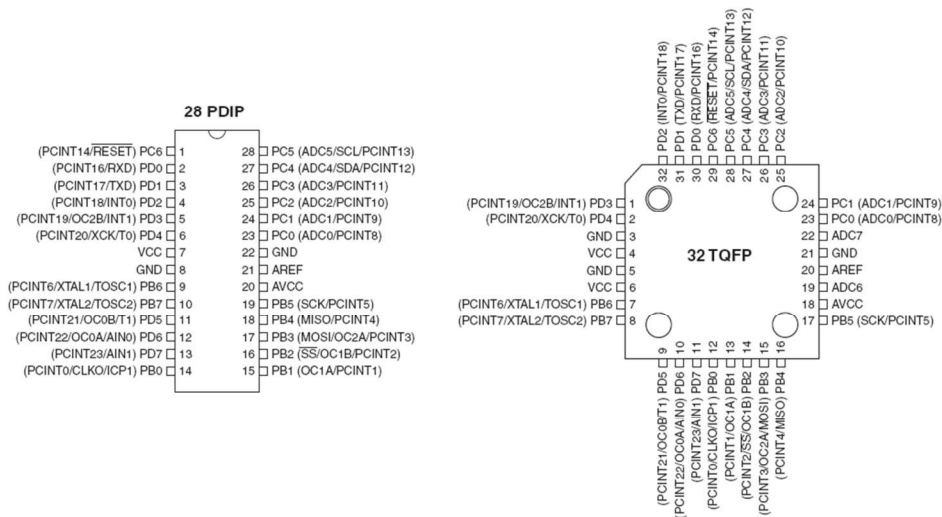


Fig. 2.9- Encapsulamentos PDIP e TQFP para o ATmega328.

Tab. 2.3 - Descrição dos pinos do ATmega328.

PINOS DE ALIMENTAÇÃO	
VCC	Tensão de alimentação.
AVCC	Pino para a tensão de alimentação do conversor AD. Deve ser externamente conectado ao VCC, mesmo se o ADC não estiver sendo utilizado.
AREF	Pino para a tensão de referência analógica do conversor AD.
GND	Terra.

PORTB	
PB0	ICP1 – entrada de captura para o Temporizador/Contador 1. CLKO – saída de <i>clock</i> do sistema. PCINT0 – interrupção 0 por mudança no pino.
PB1	OC1A – saída da igualdade de comparação A do Temporizador/Contador 1 (PWM). PCINT1 – interrupção 1 por mudança no pino.
PB2	SS – pino de seleção de escravo da SPI (<i>Serial Peripheral Interface</i>). OC1B – saída da igualdade de comparação B do Temporizador/Contador 1 (PWM). PCINT2 – interrupção 2 por mudança no pino.
PB3	MOSI – pino mestre de saída e escravo de entrada da SPI. OC2A – saída da igualdade de comparação A do Temporizador/Contador 2 (PWM). PCINT3 – interrupção 3 por mudança no pino.
PB4	MISO – pino mestre de entrada e escravo de saída da SPI. PCINT4 – interrupção 4 por mudança no pino.
PB5	SCK – pino de <i>clock</i> da SPI. PCINT5 – interrupção 5 por mudança no pino.
PB6	XTAL1 – entrada 1 do oscilador ou entrada de <i>clock</i> externa. TOSC1 – entrada 1 para o oscilador do temporizador (RTC). PCINT6 – interrupção 6 por mudança no pino.
PB7	XTAL2 – entrada 2 do oscilador. TOSC2 – entrada 2 para o oscilador do temporizador (RTC). PCINT7 – interrupção 7 por mudança no pino.
PORTC	
PC0	ADC0 – canal 0 de entrada do conversor AD. PCINT8 – interrupção 8 por mudança no pino.
PC1	ADC1 – canal 1 de entrada do conversor AD. PCINT9 – interrupção 9 por mudança no pino.
PC2	ADC2 – canal 2 de entrada do conversor AD. PCINT10 – interrupção 10 por mudança no pino.
PC3	ADC3 – canal 3 de entrada do conversor AD. PCINT11 – interrupção 11 por mudança no pino.
PC4	ADC4 – canal 4 de entrada do conversor AD. SDA – entrada e saída de dados da interface a 2 fios (TWI – I2C). PCINT12 – interrupção 12 por mudança no pino.
PC5	ADC5 – canal 5 de entrada do conversor AD. SCL – <i>clock</i> da interface a 2 fios (TWI – I2C). PCINT13 – interrupção 13 por mudança no pino.
PC6	RESET – pino de inicialização. PCINT14 – interrupção 14 por mudança no pino.

PORTD	
PD0	RXD – pino de entrada (leitura) da USART. PCINT16 – interrupção 16 por mudança no pino.
PD1	TXD – pino de saída (escrita) da USART. PCINT17 – interrupção 17 por mudança no pino.
PD2	INT0 – entrada da interrupção externa 0. PCINT18 – interrupção 18 por mudança no pino.
PD3	INT1 – entrada da interrupção externa 1. OC2B – saída da igualdade de comparação B do Temporizador/Contador 2 (PWM) PCINT19 – interrupção 19 por mudança no pino.
PD4	XCK – <i>clock</i> externo de entrada e saída da USART. T0 – entrada de contagem externa para o Temporizador/Contador 0. PCINT 20 – interrupção 20 por mudança no pino.
PD5	T1 – entrada de contagem externa para o Temporizador/Contador 1. OC0B – saída da igualdade de comparação B do Temporizador/Contador 0 (PWM). PCINT 21 – interrupção 21 por mudança no pino.
PD6	AIN0 – entrada positiva do comparador analógico. OC0A – saída da igualdade de comparação A do Temporizador/Contador 0 (PWM). PCINT 22 – interrupção 22 por mudança no pino.
PD7	AIN1 – entrada negativa do comparador analógico. PCINT 23 – interrupção 23 por mudança no pino.

Como pode ser visto na tab. 2.3, os pinos do ATmega são organizados em conjuntos denominados: PORTB, PORTC e PORTD. Cada um deles possui 8 pinos (com exceção do PORTC) organizados pelos nomes: PB0-7, PC0-6 e PD0-7. Cada PORT é um porta bidirecional de I/O de 8 bits com resistores internos de *pull-up* selecionáveis para cada bit. Os registradores de saída possuem características simétricas com capacidade de fornecer e receber corrente, suficiente para o acionamento direto de cargas de até 40 mA. Outra característica importante é que todos os pinos apresentam pelo menos duas funções distintas, até mesmo o pino de *reset* pode ser utilizado como pino de I/O.

2.3 SISTEMA DE CLOCK

Na fig. 2.10, é apresentado o diagrama esquemático do sistema de *clock* do AVR e sua distribuição. Para a redução do consumo de potência, os módulos de *clock* podem ser suspensos usando diferentes modos de programação. O AVR suporta as seguintes opções de *clock*: cristal ou ressonador cerâmico externo, cristal de baixa frequência externo, sinal de *clock* externo e oscilador RC interno, como é exemplificado na fig. 2.1.1.

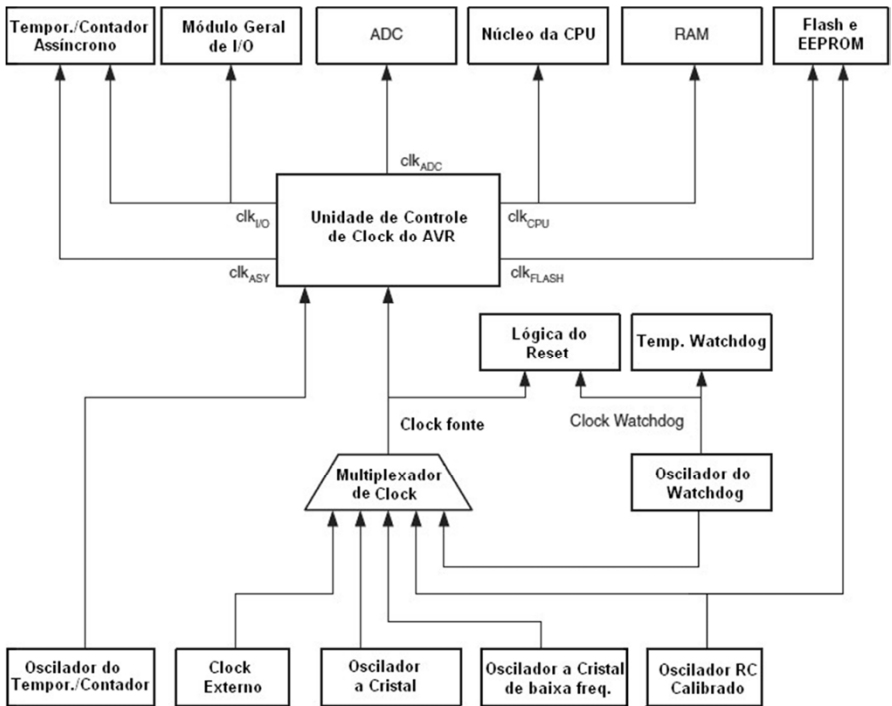


Fig. 2.10- Diagrama esquemático do sistema de *clock* do AVR e sua distribuição.

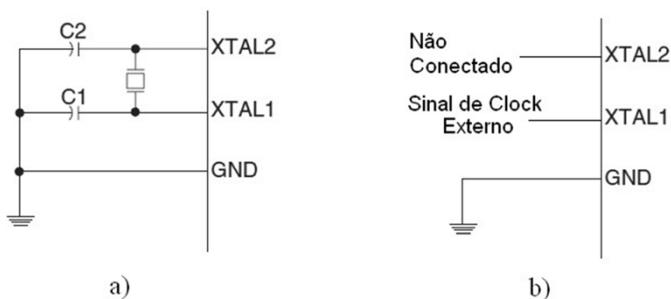


Fig. 2.11- Opções de *clock* externo para o AVR: a) cristal e b) sinal externo.

Quando não é necessário o emprego de um cristal ou ressonador cerâmico externo, é possível utilizar o oscilador interno, permitindo o uso dos pinos XTAL1 e XTAL2 do ATmega328 para outras funções. O oscilador interno pode ser programado para operar com a frequência máxima de 8 MHz (o valor *default* é 1 MHz). Apesar de ter sua frequência dependente da tensão e da temperatura, a frequência do oscilador interno pode ser precisamente calibrada pelo usuário.

2.4 O RESET

A inicialização é fundamental para o trabalho do microcontrolador. Ela é feita durante a energização do circuito ou quando se deseja inicializar os registradores da CPU. Durante o *reset* (inicialização), todos os registradores de entrada e saída são ajustados para os seus valores *default* (padrão) e o programa começa a ser executado a partir do vetor de *reset* (endereço 0 da memória de programa, ou do endereço do *boot loader*). O diagrama do circuito de *reset* é apresentado na fig. 2.12.

As portas de I/O são imediatamente inicializadas quando uma fonte de *reset* é ativa, isso não exige qualquer sinal de *clock*. Após o *reset* ficar inativo, é efetuado um atraso interno automático (configurável) mantendo o *reset* por um pequeno período de tempo. Assim, a tensão de alimentação

pode alcançar um nível estável antes do microcontrolador começar a operar.

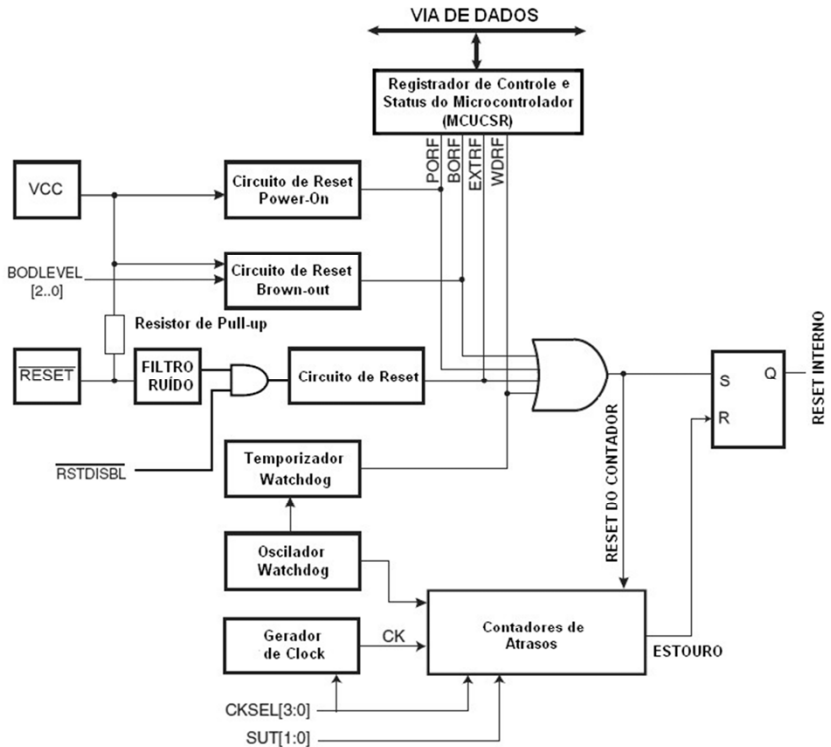


Fig. 2.12- Diagrama do circuito de *reset* do AVR.

O ATmega possui 4 fontes de *reset*:

- Power-on Reset: ocorre na energização enquanto a fonte de alimentação estiver abaixo da tensão limiar de *power-on reset* (V_{POT}).
- Reset externo: ocorre quando o pino de *reset* é aterrado (0 V) por um determinado periodo de tempo.
- Watchdog Reset: ocorre quando o *watchdog* está habilitado e o seu contador atinge o valor limite.

- Brown-out Reset: ocorre quando a tensão de alimentação cair abaixo do valor definido para o *brown-out reset* (V_{BOT}) e o seu detector estiver habilitado.

Recomenda-se conectar o circuito da fig. 2.13 ao pino de *reset*. Os valores usuais são de 10 k Ω para o resistor e de 100 nF para o capacitor⁹. Para economia de componentes, o pino de *reset* pode ser ligado diretamente ao VCC. Todavia, o *reset* físico é indicado quando se deseja inicializar manualmente o microcontrolador. Assim, no circuito da fig. 2.13 pode-se empregar um botão entre o pino de *reset* e o terra.

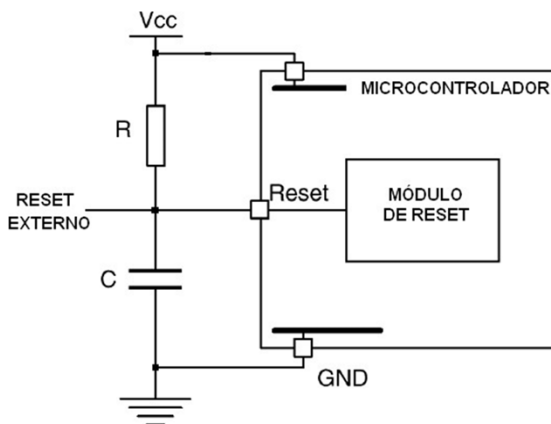


Fig. 2.13- Circuito para ligação ao pino de *reset*.

⁹ Consultar o *application note* do fabricante, AVR042: AVR Hardware Design Considerations.

2.5 GERENCIAMENTO DE ENERGIA E O MODO SLEEP

Para a economia de energia, necessária com o uso de baterias, o modo *sleep* (soneca) permite o desligamento de partes internas do chip não utilizadas do microcontrolador. Esse modo é tão importante que existe um código de instrução para ele: o SLEEP. O AVR possui 6 modos possíveis de *sleep*:

- Idle: a atividade da CPU é suspensa, mas a SPI, a USART, o comparador analógico, o ADC, a interface serial à 2 fios, os temporizadores/contadores, o *watchdog timer* e o sistema de interrupção continuam operando. Basicamente, o *clock* da CPU e da memória *flash* são suspensos.
- Redução de Ruído para o ADC: a CPU é parada, mas continuam operando o ADC, as interrupções externas, igualdade de endereço da interface serial à 2 fios, o temporizador/contador 2 e o *watchdog timer* (se habilitado). Esse modo é empregado para reduzir o ruído para o ADC e garantir sua resolução.
- Power-down: o funcionamento do oscilador externo é suspenso, mas continuam operando a interface serial à 2 fios, as interrupções externas e o *watchdog timer* (se habilitado). Basicamente, interrompe todos os *clocks* gerados.
- Power-save: igual ao modo *power-down*, com exceção que o contador/temporizador 2 continua trabalhando assincronamente.
- Standby: é idêntico ao modo *power-down*, com exceção que o oscilador é mantido funcionando (válido para oscilador externo a cristal ou ressonador cerâmico). O microcontrolador ‘desperta’ do *sleep* em 6 ciclos de *clock*.
- Extended Standby: idêntico ao modo *power-save*, com exceção que o oscilador é mantido funcionando. Leva 6 ciclos de *clock* para o microcontrolador ‘despertar’.

De acordo com os requisitos do projeto, um ou outro modo de economia de energia deverá ser escolhido; quanto maior o número de fontes de *clock* desligadas, menor o consumo de corrente. É importante consultar o manual do fabricante para compreender como se obter o máximo desempenho. Na tab. 2.4, são apresentadas quais fontes são desligadas e quais sinais ‘despertam’ o microcontrolador.

Tab. 2.4 – Fontes desligadas e sinais de ‘despertar’ para os diferentes modos *sleep*.

	Sinais de <i>clock</i> ativos					Osciladores		Fontes de <i>wake-up</i> (para ‘despertar’)							
Modos Sleep	Clock CPU	Clock FLASH	Clock IO	Clock ADC	Clock ASY	Fonte principal de <i>clock</i> habilitada	Oscilador do Temporizador habilitado	INT1, INT0 e mudança nos pinos	Casamento de endereço da interface serial à 2 fios	Temporizador 2	SPM/EEPROM prontos	ADC	Watchdog Timer	Outras I/O	BOD ⁴ desabilitado por software
<i>Idle</i>			X	X	X	X	X ²	X ³	X	X	X	X	X	X	
Redução de Ruído para o ADC				X	X	X	X ²	X ³	X	X ²	X	X	X		
<i>Power-Down</i>								X ³	X				X		X
<i>Power-Save</i>					X		X ²	X ³	X	X			X		X
<i>Standby¹</i>						X		X ³	X				X		X
<i>Extended Standby</i>					X ²	X	X ²	X ³	X	X			X		X

- Notas:
- 1 – Recomendado somente quando se emprega cristal externo ou ressonador.
 - 2 – Se o Temporizador/Contador 2 estiver no modo assíncrono.
 - 3 – Somente para as interrupções INT0 e INT1 por nível.
 - 4 – BOD somente em dispositivos com *picoPower* (ATmega48PA/88PA/168PA/328P).