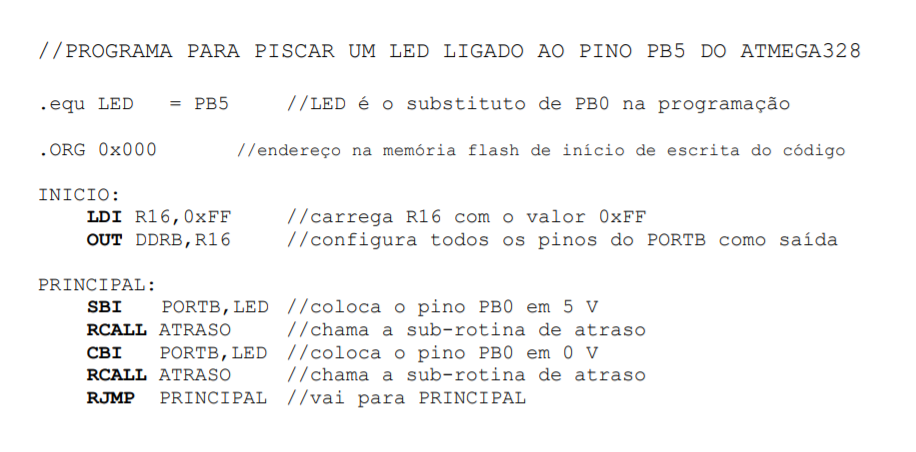
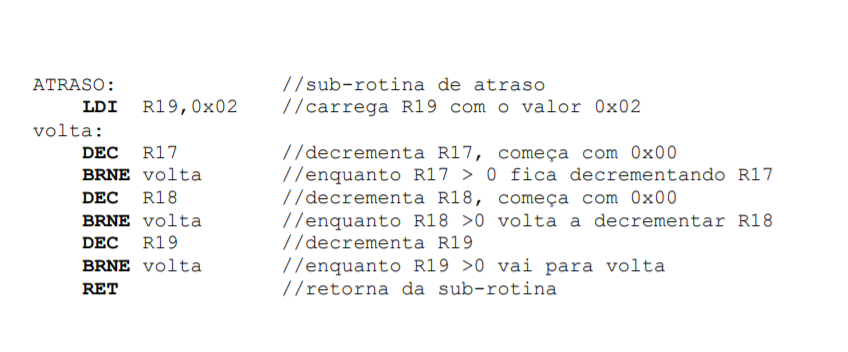
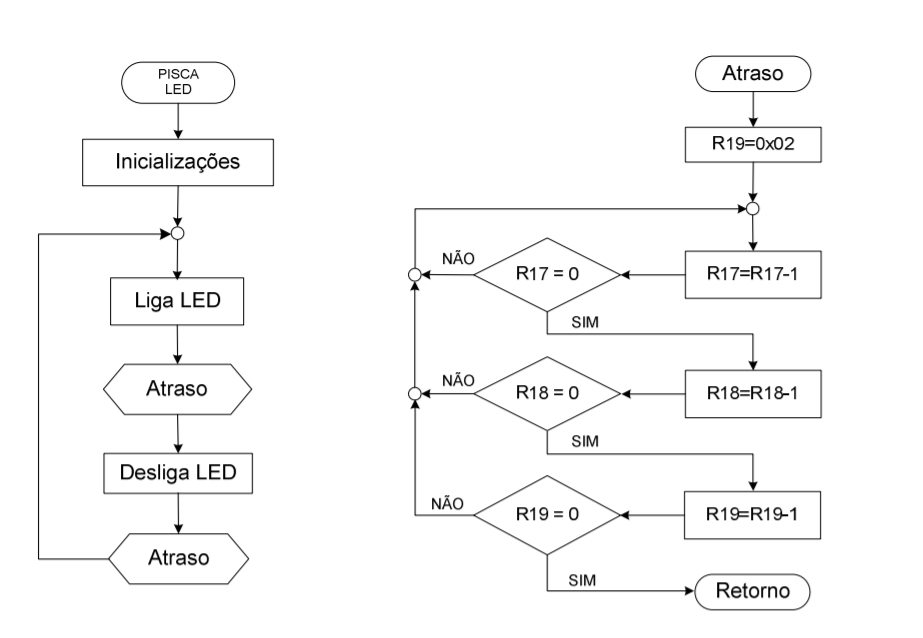


O oscilador interno pode ser programado para operar com a frequência máxima de 8 MHz (o valor default é 1 MHz).

O Atmega tem uma função de sub-rotina de atraso, RCALL ATRASO, podemos utiliza-la para fazer os estados de wait. No caso desse programa o atraso é utilizado para manter o led aceso ou apagado antes de ir para o próximo estado.







PortD, todos os pinos são saídas de 8 bits e a representação é Hexadecimal, exemplo: 0xA8 = 1010 1000

O ATmega328 possui 3 conjuntos de pinos de entrada e saída (I/Os): PORTB, PORTC e PORTD; respectivamente, pinos PB7 .. PB0, PC6 .. PC0 e PD7 .. PD0, todos com a função Lê – Modifica – Escreve. Isso significa que a direção de um pino pode ser alterada sem mudar a direção de qualquer outro pino do mesmo PORT (instruções SBI e CBI19)

SBI = ativa um bit (coloca em 1). CBI = limpa um bit (coloca em 0).

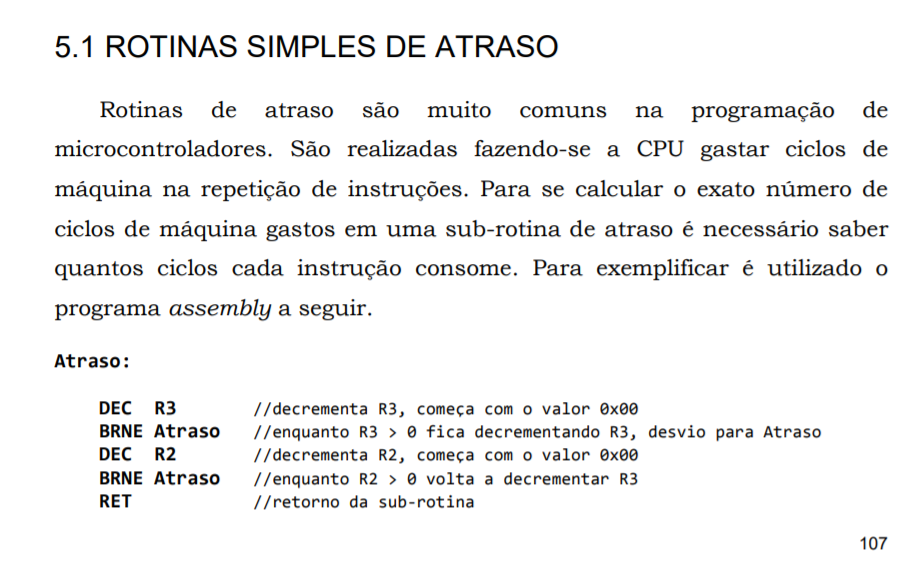
Os registradores responsáveis pelos pinos de I/O são:

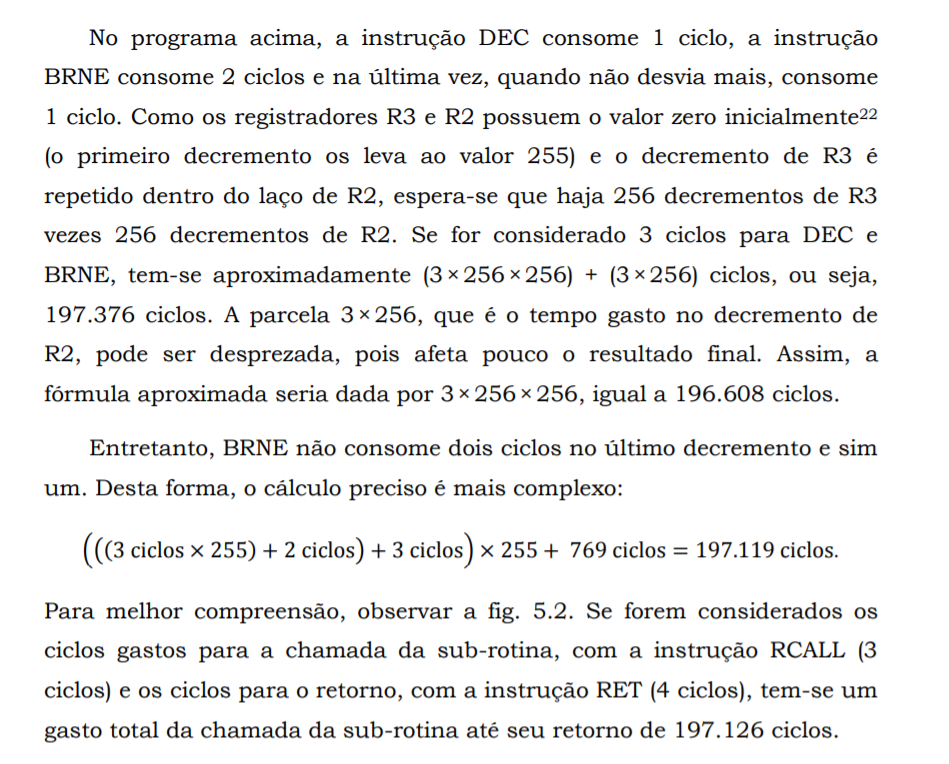
PORTx: registrador de dados, usado para escrever nos pinos do PORTx.

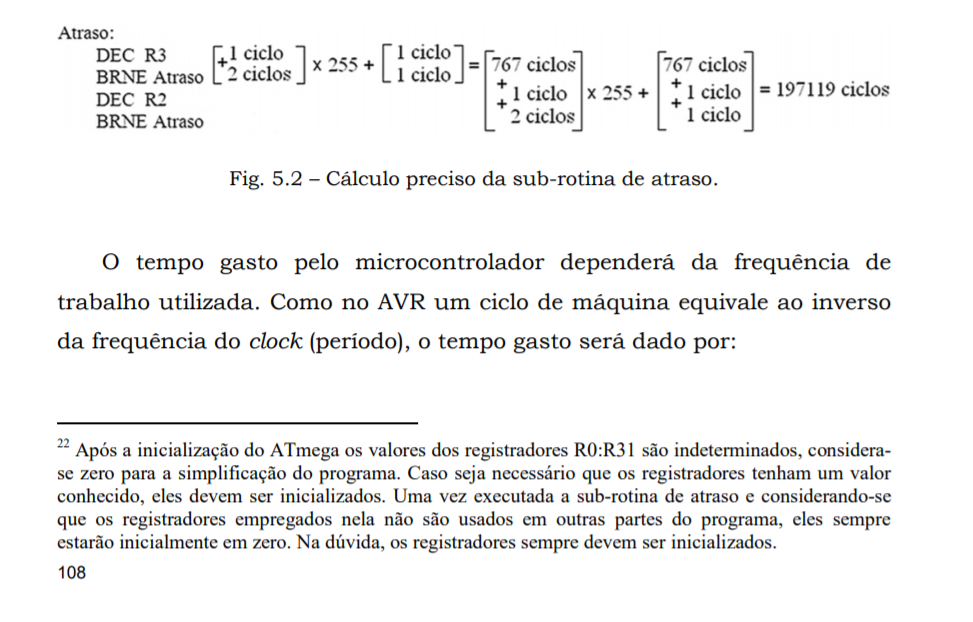
DDRx: registrador de direção, usado para definir se os pinos do PORTx são entrada ou saída.

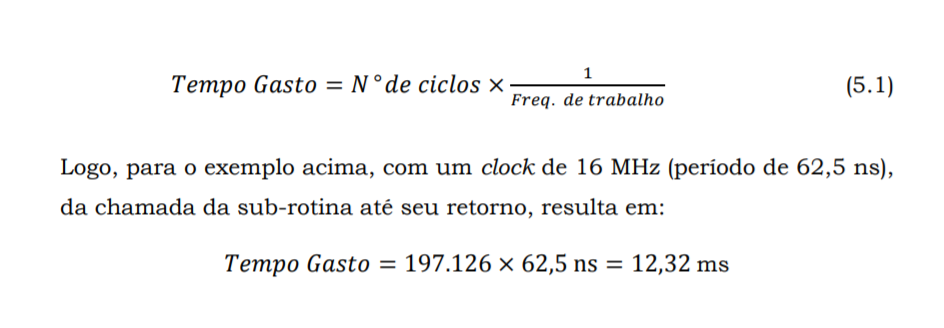
PINx: registrador de entrada, usado para ler o conteúdo dos pinos do PORTx.

Em resumo, para o uso de um pino de I/O, deve-se primeiro definir se ele será entrada ou saída escrevendo-se no registrador DDRx. Então, a escrita no registrador PORTx alterará o estado lógico do pino se ele for saída, ou poderá habilitar o pull-up interno, se ele for entrada21. Os estados lógicos dos pinos do PORT são lidos do registrador PINx. É importante notar que para a leitura do PINx logo após uma escrita no PORTx ou DDRx, deve ser gasto pelo menos um ciclo de máquina para a sincronização dos dados pelo microcontrolador.









PAGINA 107