

CAPÍTULO 1

Introdução aos Microcontroladores

CAPÍTULO 1.....	1
1 INTRODUÇÃO.....	2
2 HISTÓRIA	2
3 MICROCONTROLADORES VERSUS MICROPROCESSADORES.....	5
4.1 UNIDADE DE MEMÓRIA	5
4.2 UNIDADE CENTRAL DE PROCESSAMENTO	6
4.3 BUS.....	7
4.4 UNIDADE DE ENTRADA/SAÍDA.....	8
4.5 COMUNICAÇÃO SÉRIE	8
4.6 UNIDADE DE TEMPORIZAÇÃO.....	10
4.7 WATCHDOG	11
4.8 CONVERSOR ANALÓGICO - DIGITAL	12
4.9 PROGRAMA	14

1 Introdução

As circunstâncias que se nos deparam hoje no campo dos microcontroladores têm os seus primórdios no desenvolvimento da tecnologia dos circuitos integrados. Este desenvolvimento tornou possível armazenar centenas de milhares de transístores num único chip. Isso constituiu um pré-requisito para a produção de microprocessadores e, os primeiros computadores foram construídos adicionando periféricos externos tais como memória, linhas de entrada e saída, temporizadores e outros. Um crescente aumento do nível de integração, permitiu o aparecimento de circuitos integrados contendo simultaneamente processador e periféricos. Foi assim que o primeiro chip contendo um microcomputador e que mais tarde haveria de ser designado por microcontrolador, apareceu.

2 História

É no ano de 1969 que uma equipa de engenheiros japoneses pertencentes à companhia BUSICOM chega aos Estados Unidos com a encomenda de alguns circuitos integrados para calculadoras a serem implementados segundo os seus projectos. A proposta foi entregue à INTEL e Marcian Hoff foi o responsável pela sua concretização. Como ele tinha tido experiência de trabalho com um computador (PC) PDP8, lembrou-se de apresentar uma solução substancialmente diferente em vez da construção sugerida. Esta solução pressupunha que a função do circuito integrado seria determinada por um programa nele armazenado. Isso significava que a configuração deveria ser mais simples, mas também era preciso muito mais memória que no caso do projecto proposto pelos engenheiros japoneses. Depois de algum tempo, embora os engenheiros japoneses tenham tentado encontrar uma solução mais fácil, a ideia de Marcian venceu e o primeiro microprocessador nasceu. Ao transformar esta ideia num produto concreto, Frederico Faggin foi de uma grande utilidade para a INTEL. Ele transferiu-se para a INTEL e, em somente 9 meses, teve sucesso na criação de um produto real a partir da sua primeira concepção. Em 1971, a INTEL adquiriu os direitos sobre a venda deste bloco integral. Primeiro eles compraram a licença à companhia BUSICOM que não tinha a mínima percepção do tesouro que possuía. Neste mesmo ano, apareceu no mercado um microprocessador designado por 4004. Este foi o primeiro microprocessador de 4 bits e tinha a velocidade de 6 000 operações por segundo. Não muito tempo depois, a

companhia Americana CTC pediu à INTEL e à Texas Instruments um microprocessador de 8 bits para usar em terminais. Mesmo apesar de a CTC acabar por desistir desta ideia, tanto a Intel como a Texas Instruments continuaram a trabalhar no microprocessador e, em Abril de 1972, os primeiros microprocessadores de 8 bits apareceram no mercado com o nome de 8008. Este podia endereçar 16KB de memória, possuía 45 instruções e tinha a velocidade de 300 000 operações por segundo. Esse microprocessador foi o pioneiro de todos os microprocessadores actuais. A Intel continuou com o desenvolvimento do produto e, em Abril de 1974 pôs cá fora um processador de 8 bits com o nome de 8080 com a capacidade de endereçar 64KB de memória, com 75 instruções e com preços a começarem em \$360.

Uma outra companhia Americana, a Motorola, apercebeu-se rapidamente do que estava a acontecer e, assim, pôs no mercado um novo microprocessador de 8 bits, o 6800. O construtor chefe foi Chuck Peddle e além do microprocessador propriamente dito, a Motorola foi a primeira companhia a fabricar outros periféricos como os 6820 e 6850. Nesta altura, muitas companhias já se tinham apercebido da enorme importância dos microprocessadores e começaram a introduzir os seus próprios desenvolvimentos. Chuck Peddle deixa a Motorola para entrar para a MOS Technology e continua a trabalhar intensivamente no desenvolvimento dos microprocessadores.

Em 1975, na exposição WESCON nos Estados Unidos, ocorreu um acontecimento crítico na história dos microprocessadores. A MOS Technology anunciou que ia pôr no mercado microprocessadores 6501 e 6502 ao preço de \$25 cada e que podia satisfazer de imediato todas as encomendas. Isto pareceu tão sensacional que muitos pensaram tratar-se de uma espécie de vigarice, considerando que os competidores vendiam o 8080 e o 6800 a \$179 cada. Para responder a este competidor, tanto a Intel como a Motorola baixaram os seus preços por microprocessador para \$69,95 logo no primeiro dia da exposição. Rapidamente a Motorola pôs uma acção em tribunal contra a MOS Technology e contra Chuck Peddle por violação dos direitos de autor por copiarem ao copiarem o 6800. A MOS Technology deixou de fabricar o 6501, mas continuou com o 6502. O 6502 é um microprocessador de 8 bits com 56 instruções e uma capacidade de endereçamento de 64KB de memória. Devido ao seu baixo custo, o 6502 torna-se muito popular e, assim, é instalado em computadores como KIM-1, Apple I, Apple II, Atari, Comodore, Acorn, Oric, Galeb, Orais, Ultra e muitos outros. Cedo aparecem vários fabricantes do 6502 (Rockwell, Sznertek, GTE, NCR, Ricoh e Comodore adquiriram a

MOS Technology) que, no auge da sua prosperidade, chegou a vender microprocessadores à razão de 15 milhões por ano !

Contudo, os outros não baixaram os braços. Frederico Faggin deixa a Intel e funda a Zilog Inc.

Em 1976, a Zilog anuncia o Z80. Durante a concepção deste microprocessador, Faggin toma uma decisão crítica. Sabendo que tinha sido já desenvolvida uma enorme quantidade de programas para o 8080, Faggin conclui que muitos vão permanecer fieis a este microprocessador por causa das grandes despesas que adviriam das alterações a todos estes programas. Assim, ele decide que o novo microprocessador deve ser compatível com o 8080, ou seja, deve ser capaz de executar todos os programas que já tenham sido escritos para o 8080. Além destas características, outras características adicionais foram introduzidas, de tal modo que o Z80 se tornou um microprocessador muito potente no seu tempo. Ele podia endereçar directamente 64KB de memória, tinha 176 instruções, um grande número de registos, uma opção para refrescamento de memória RAM dinâmica, uma única alimentação, maior velocidade de funcionamento, etc. O Z80 tornou-se um grande sucesso e toda a gente se transferiu do 8080 para o Z80.

Pode dizer-se que o Z80 se constituiu sem sombra de dúvida como o microprocessador de 8 bits com maior sucesso no seu tempo. Além da Zilog, outros novos fabricantes como Mostek, NEC, SHARP e SGS apareceram. O Z80 foi o coração de muitos computadores como o Spectrum, Partner, TRS703, Z-3 e Galaxy, que foram aqui usados.

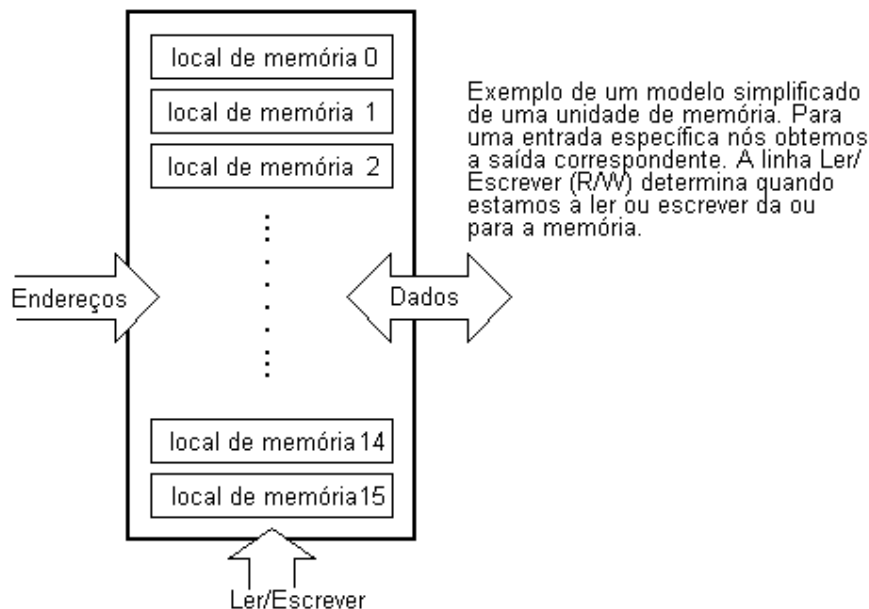
Em 1976, a Intel apareceu com uma versão melhorada do microprocessador de 8 bits e designada por 8085. Contudo, o Z80 era tão superior a este que, bem depressa, a Intel perdeu a batalha. Ainda que mais alguns microprocessadores tenham aparecido no mercado (6809, 2650, SC/MP etc.), já tudo estava então decidido. Já não havia mais grandes melhorias a introduzir pelos fabricantes que fundamentassem a troca por um novo microprocessador, assim, o 6502 e o Z80, acompanhados pelo 6800, mantiveram-se como os mais representativos microprocessadores de 8 bits desse tempo.

3 Microcontroladores versus Microprocessadores

Um microcontrolador difere de um microprocessador em vários aspectos. Primeiro e o mais importante, é a sua funcionalidade. Para que um microprocessador possa ser usado, outros componentes devem-lhe ser adicionados, tais como memória e componentes para receber e enviar dados. Em resumo, isso significa que o microprocessador é o verdadeiro coração do computador. Por outro lado, o microcontrolador foi projectado para ter tudo num só. nenhuns outros componentes externos são necessários nas aplicações, uma vez que todos os periféricos necessários já estão contidos nele. Assim, nós poupamos tempo e espaço na construção dos dispositivos.

4.1 Unidade de Memória

A memória é a parte do microcontrolador cuja função é guardar dados. A maneira mais fácil de explicar é descrevê-la como uma grande prateleira cheia de gavetas. Se supusermos que marcamos as gavetas de modo a elas não se confundirem umas com as outras, então o seu conteúdo será facilmente acessível. Basta saber a designação da gaveta e o seu conteúdo será conhecido.



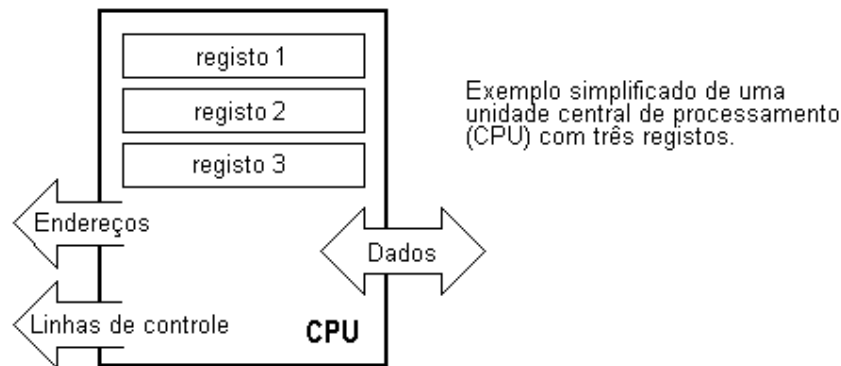
Os componentes de memória são exactamente a mesma coisa. Para um determinado endereço, nós obtemos o conteúdo desse endereço. Dois novos conceitos foram apresentados: endereçamento e memória. A memória é o conjunto de todos os locais de

memória (gavetas) e endereçamento nada mais é que seleccionar um deles. Isto significa que precisamos de seleccionar o endereço desejado (gaveta) e esperar que o conteúdo desse endereço nos seja apresentado (abrir a gaveta). Além de ler de um local da memória (ler o conteúdo da gaveta), também é possível escrever num endereço da memória (introduzir um conteúdo na gaveta). Isto é feito utilizando uma linha adicional chamada linha de controle. Nós iremos designar esta linha por R/W (read/write) - ler/escrever. A linha de controle é usada do seguinte modo: se $r/w=1$, é executada uma operação de leitura, caso contrário é executada uma operação de escrita no endereço de memória.

A memória é o primeiro elemento, mas precisamos de mais alguns para que o nosso microcontrolador possa trabalhar.

4.2 Unidade Central de Processamento

Vamos agora adicionar mais 3 locais de memória a um bloco específico para que possamos ter as capacidades de multiplicar, dividir, subtrair e mover o seus conteúdos de um local de memória para outro. A parte que vamos acrescentar é chamada "central processing unit" (CPU) ou Unidade Central de Processamento. Os locais de memória nela contidos chamam-se registos.



Os registos são, portanto, locais de memória cujo papel é ajudar a executar várias operações matemáticas ou quaisquer outras operações com dados, quaisquer que sejam os locais em que estes se encontrem.

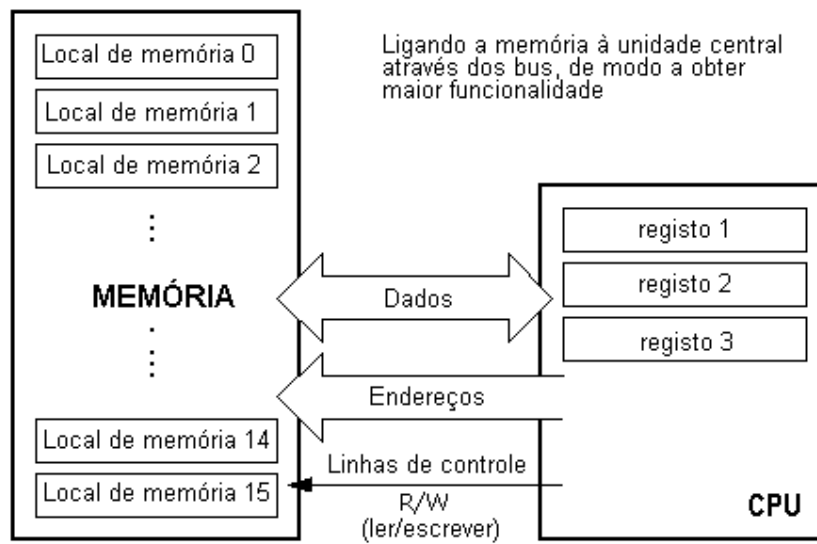
Vamos olhar para a situação actual. Nós temos duas entidades independentes (memória e CPU) que estão interligadas, deste modo, qualquer troca de dados é retardada bem como a funcionalidade do sistema é diminuída. Se, por exemplo, nós desejarmos adicionar os conteúdos de dois locais de memória e tornar a guardar o resultado na memória, nós necessitamos de uma ligação entre a memória e o CPU. Dito mais simplesmente, nós

precisamos de obter um "caminho" através do qual os dados possam passar de um bloco para outro.

4.3 Bus

Este "caminho" designa-se por "bus" . Fisicamente ele corresponde a um grupo de 8, 16 ou mais fios.

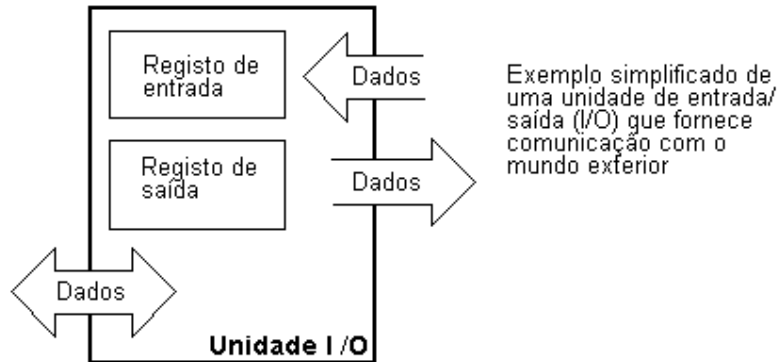
Existem dois tipos de bus: bus de dados e de endereço. O número de linhas do primeiro depende da quantidade de memória que desejamos endereçar e o número de linhas do outro depende da largura da palavra de dados, no nosso caso é igual a oito. O primeiro bus serve para transmitir endereços do CPU para a memória e o segundo para ligar todos os blocos dentro do microcontrolador.



Neste momento, a funcionalidade já aumentou mas um novo problema apareceu: nós temos uma unidade capaz de trabalhar sozinha, mas que não possui nenhum contacto com o mundo exterior, ou seja, conosco! De modo a remover esta deficiência, vamos adicionar um bloco que contém várias localizações de memória e que, de um lado, está ligado ao bus de dados e do outro às linhas de saída do microcontrolador que coincidem com pinos do circuito integrado e que, portanto, nós podemos ver com os nossos próprios olhos.

4.4 Unidade de entrada/saída

Estas localizações que acabamos de adicionar, chamam-se "portos". Existem vários tipos de portos: de entrada, de saída e de entrada/saída. Quando trabalhamos com portos primeiro de tudo é necessário escolher o porto com que queremos trabalhar e, em seguida, enviar ou receber dados para ou desse porto.

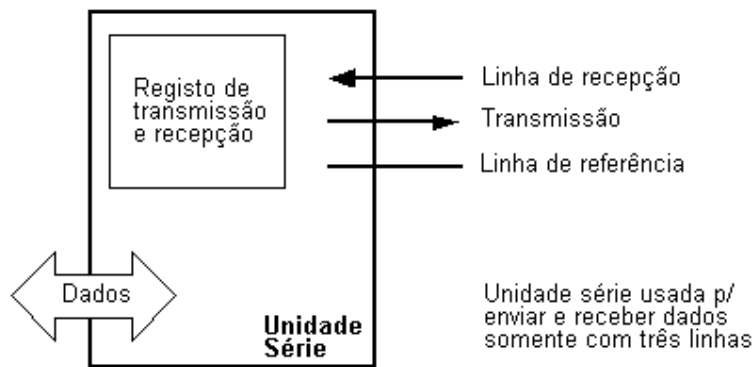


Quando se está a trabalhar com ele, o porto funciona como um local de memória. Qualquer coisa de que se está a ler ou em que se está a escrever e que é possível identificar facilmente nos pinos do microcontrolador.

4.5 Comunicação série

Anteriormente, acrescentámos à unidade já existente a possibilidade de comunicar com o mundo exterior. Contudo, esta maneira de comunicar tem os seus inconvenientes. Um dos inconvenientes básicos é o número de linhas que é necessário usarmos para transferir dados. E se for necessário transferi-los a uma distância de vários quilómetros? O número de linhas vezes o número de quilómetros não atesta a economia do projecto. Isto levamos a ter que reduzir o número de linhas de modo a que a funcionalidade se mantenha. Suponha que estamos a trabalhar apenas com três linhas e que uma linha é usada para enviar dados, outra para os receber e a terceira é usada como linha de referência tanto do lado de entrada como do lado da saída. Para que isto trabalhe nós precisamos de definir as regras para a troca de dados. A este conjunto de regras chama-se protocolo. Este protocolo deve ser definido com antecedência de modo que não haja mal entendidos entre as partes que estão a comunicar entre si. Por exemplo, se um homem está a falar

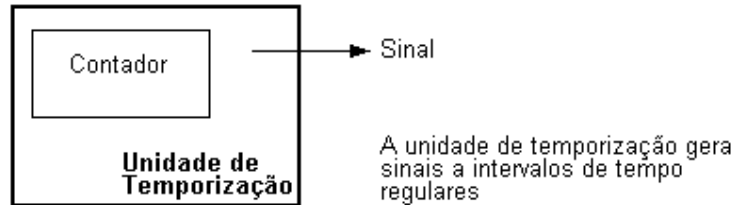
em francês e o outro em inglês, é altamente improvável que efectivamente e rapidamente, ambos se entendam. Vamos supor que temos o seguinte protocolo. A unidade lógica "1" é colocada na linha de transmissão até que a transferência se inicie. Assim que isto acontece, a linha passa para nível lógico '0' durante um certo período de tempo (que vamos designar por T), assim, do lado da recepção ficamos a saber que existem dados para receber e, o mecanismo de recepção, vai activar-se. Regressemos agora ao lado da emissão e comecemos a pôr zeros e uns lógicos na linha de transmissão correspondentes aos bits, primeiro o menos significativo e finalmente o mais significativo. Vamos esperar que cada bit permaneça na linha durante um período de tempo igual a T, e, finalmente, depois do oitavo bit, vamos pôr novamente na linha o nível lógico "1", o que assinala a transmissão de um dado. O protocolo que acabamos de descrever é designado na literatura profissional por NRZ (Não Retorno a Zero).



Como nós temos linhas separadas para receber e enviar, é possível receber e enviar dados (informação) simultaneamente. O bloco que possibilita este tipo de comunicação é designado por bloco de comunicação série. Ao contrário da transmissão em paralelo, aqui os dados movem-se bit após bit em série, daqui provém o nome de comunicação série. Depois de receber dados nós precisamos de os ler e guardar na memória, no caso da transmissão de dados o processo é inverso. Os dados vêm da memória através do bus para o local de transmissão e dali para a unidade de recepção de acordo com o protocolo.

4.6 Unidade de temporização

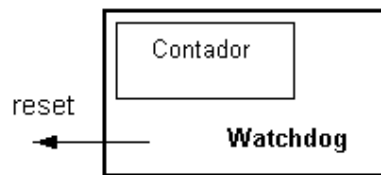
Agora que já temos a unidade de comunicação série implementada, nós podemos receber, enviar e processar dados.



Contudo, para sermos capazes de utilizar isto na indústria precisamos ainda de mais alguns blocos. Um deles é o bloco de temporização que nos interessa bastante porque pode dar-nos informações acerca da hora, duração, protocolo, etc. A unidade básica do temporizador é um contador que é na realidade um registo cujo conteúdo aumenta de uma unidade num intervalo de tempo fixo, assim, anotando o seu valor durante os instantes de tempo $T1$ e $T2$ e calculando a sua diferença, nós ficamos a saber a quantidade de tempo decorrida. Esta é uma parte muito importante do microcontrolador, cujo domínio vai requerer muita da nossa atenção.

4.7 Watchdog

Uma outra coisa que nos vai interessar é a fluência da execução do programa pelo microcontrolador durante a sua utilização. Suponha que como resultado de qualquer interferência (que ocorre frequentemente num ambiente industrial), o nosso microcontrolador pára de executar o programa ou, ainda pior, desata a trabalhar incorrectamente.



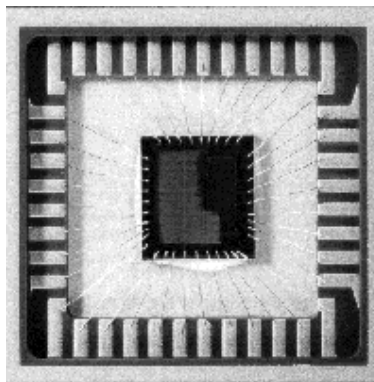
Claro que, quando isto acontece com um computador, nós simplesmente carregamos no botão de reset e continuamos a trabalhar. Contudo, no caso do microcontrolador nós não podemos resolver o nosso problema deste modo, porque não temos botão. Para ultrapassar este obstáculo, precisamos de introduzir no nosso modelo um novo bloco chamado watchdog (cão de guarda). Este bloco é de facto outro contador que está continuamente a contar e que o nosso programa põe a zero sempre que é executado correctamente. No caso de o programa "encravar", o zero não vai ser escrito e o contador, por si só, encarregar-se-á de fazer o reset do microcontrolador quando alcançar o seu valor máximo. Isto vai fazer com que o programa corra de novo e desta vez correctamente. Este é um elemento importante para que qualquer programa se execute fiavelmente, sem precisar da intervenção do ser humano.

4.8 Conversor analógico - digital

Como os sinais dos periféricos são substancialmente diferentes daqueles que o microcontrolador pode entender (zero e um), eles devem ser convertidos num formato que possa ser compreendido pelo microcontrolador. Esta tarefa é executada por intermédio de um bloco destinado à conversão analógica-digital ou com um conversor A/D. Este bloco vai ser responsável pela conversão de uma informação de valor analógico para um número binário e pelo seu trajecto através do bloco do CPU, de modo a que este o possa processar de imediato.

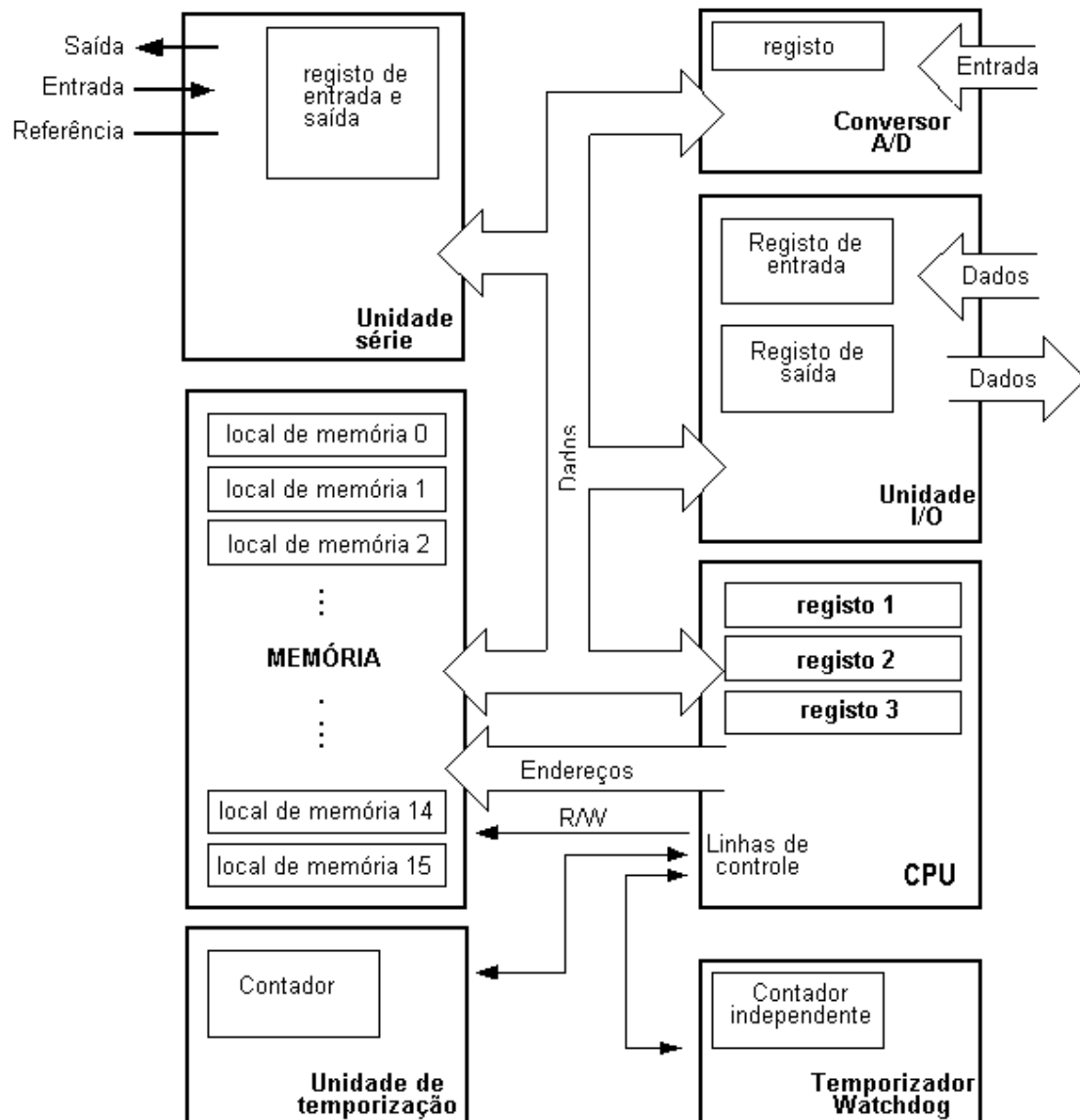


Neste momento, a configuração do microcontrolador está já terminada, tudo o que falta é introduzi-la dentro de um aparelho electrónico que poderá aceder aos blocos internos através dos pinos deste componente. A figura a seguir, ilustra o aspecto interno de um microcontrolador.



Configuração física do interior de um microcontrolador

As linhas mais finas que partem do centro em direcção à periferia do microcontrolador correspondem aos fios que interligam os blocos interiores aos pinos do envólucro do microcontrolador. O gráfico que se segue representa a parte principal de um microcontrolador.



Esquema de um micro controlador com os seus elementos básicos e ligações internas.

Numa aplicação real, um microcontrolador, por si só, não é suficiente. Além dele, nós necessitamos do programa que vai ser executado e de mais alguns elementos que constituirão um interface lógico para outros elementos (que vamos discutir em capítulos mais à frente).

4.9 Programa

Escrever um programa é uma parte especial do trabalho com microcontroladores e é designado por "programação". Vamos tentar escrever um pequeno programa numa linguagem que seremos nós a criar e que toda a gente será capaz de compreender.

INICIO

REGISTO1=LOCAL_DE_MEMORIA_A

REGISTO2=LOCAL_DE_MEMORIA_B

PORTO_A=REGISTO1+REGISTO2

FIM

O programa adiciona os conteúdos de dois locais de memória e coloca a soma destes conteúdos no porto A. A primeira linha do programa manda mover o conteúdo do local de memória "A" para um dos registos da unidade central de processamento. Como necessitamos também de outra parcela, vamos colocar o outro conteúdo noutra registo da unidade central de processamento (CPU). A instrução seguinte pede ao CPU para adicionar os conteúdos dos dois registos e enviar o resultado obtido para o porto A, de modo a que o resultado desta adição seja visível para o mundo exterior. Para um problema mais complexo, naturalmente o programa que o resolve será maior.

A tarefa de programação pode ser executada em várias linguagens tais como o Assembler, C e Basic que são as linguagens normalmente mais usadas. O Assembler pertence ao grupo das linguagens de baixo nível que implicam um trabalho de programação lento, mas que oferece os melhores resultados quando se pretende poupar espaço de memória e aumentar a velocidade de execução do programa. Como se trata da linguagem mais frequentemente usada na programação de microcontroladores, ela será discutida num capítulo mais adiantado. Os programas na linguagem C são mais fáceis de se escrever e compreender, mas, também, são mais lentos a serem executados que os programas assembler. Basic é a mais fácil de todas para se aprender e as suas instruções são semelhantes à maneira de um ser humano se exprimir, mas tal como a linguagem C, é também de execução mais lenta que o assembler. Em qualquer caso, antes que escolha

entre uma destas linguagens, precisa de examinar cuidadosamente os requisitos de velocidade de execução, de espaço de memória a ocupar e o tempo que vai demorar a fazer o programa em assembly.

Depois de o programa estar escrito, nós necessitamos de introduzir o microcontrolador num dispositivo e pô-lo a trabalhar. Para que isto aconteça, nós precisamos de adicionar mais alguns componentes externos. Primeiro temos que dar vida ao microcontrolador fornecendo-lhe a tensão (a tensão eléctrica é necessária para que qualquer instrumento electrónico funcione) e o oscilador cujo papel é análogo ao do coração que bate no ser humano. A execução das instruções do programa é regulada pelas pulsações do oscilador. Logo que lhe é aplicada a tensão, o microcontrolador executa uma verificação dele próprio, vai para o princípio do programa e começa a executá-lo. O modo como o dispositivo vai trabalhar depende de muitos parâmetros, os mais importantes dos quais são a competência da pessoa que desenvolve o hardware e do programador que, com o seu programa, deve tirar o máximo do dispositivo.