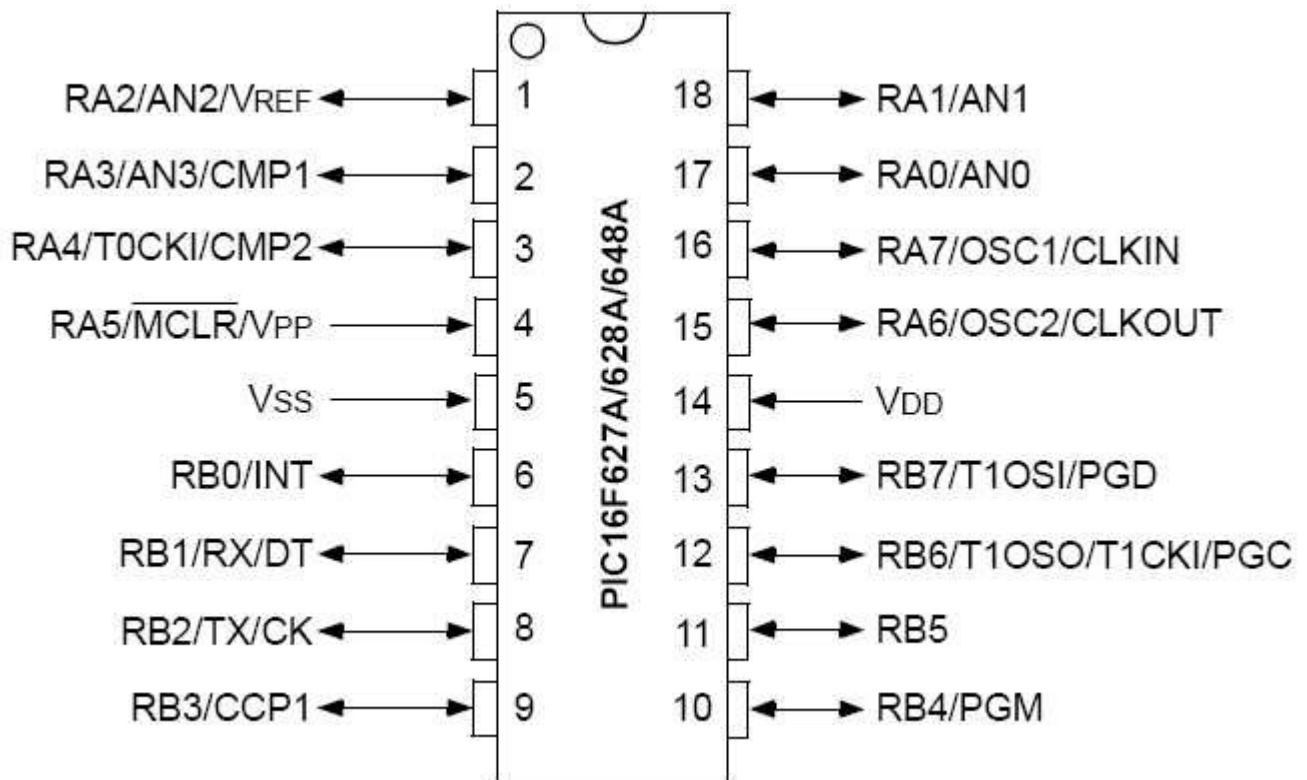


PIC16F628a

O PIC16F628a é um microcontrolador fabricado pela Microchip Technology (www.microchip.com), com as seguintes características:

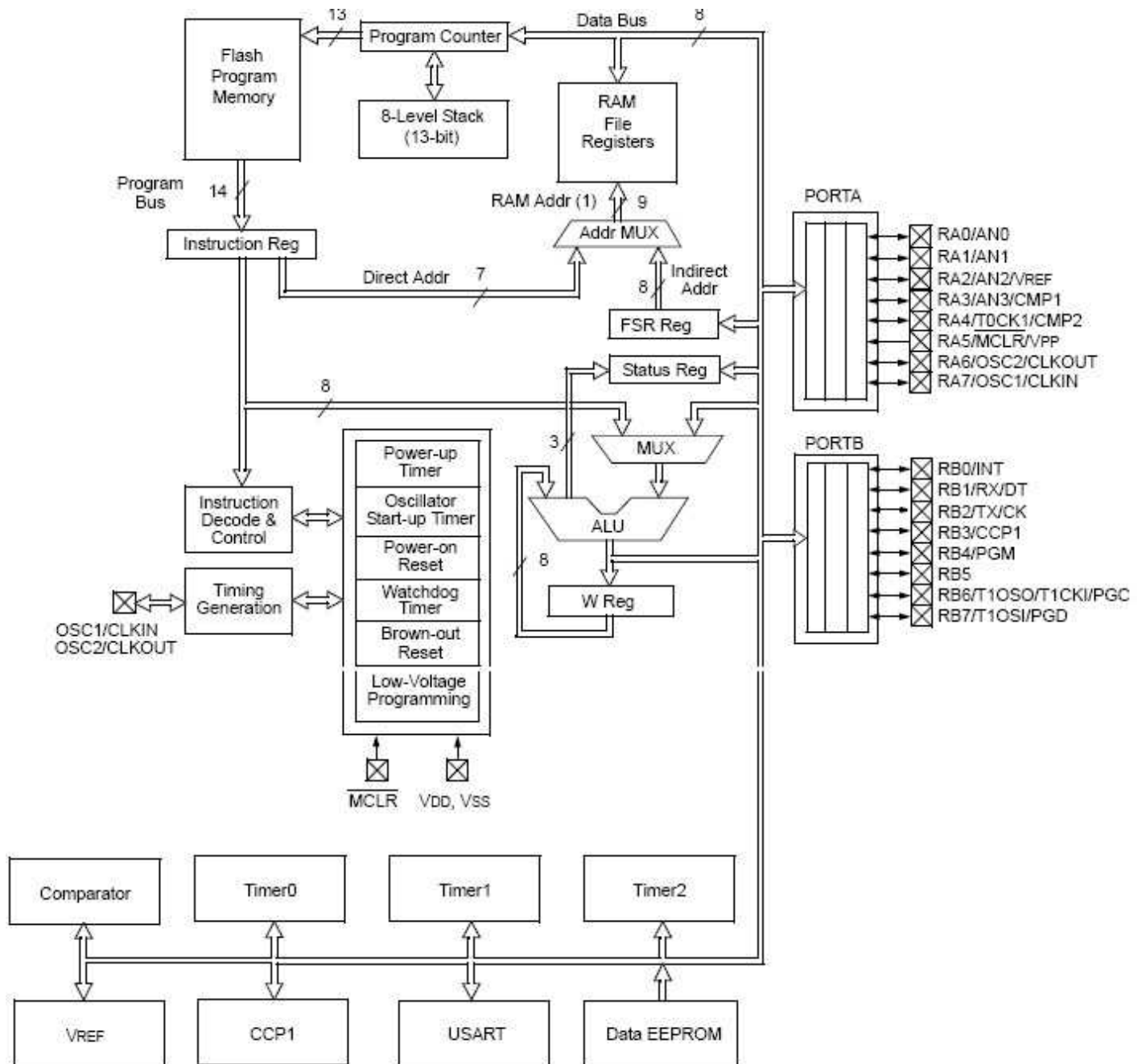
- composto de 18 pinos;
- possui somente 35 instruções no seu microcódigo;
- sinal de clock de frequência até 20 MHz;
- memória de programa do tipo Flash de 2048 words (1 word = 32 bits);
- 224 bytes de memória RAM para dados;
- 128 bytes de memória EEPROM para dados;
- instruções de 14 bits com 200ns de tempo de execução;
- dados de 8 bits por endereço de memória;
- 15 registradores especiais;
- 16 pinos os quais podem ser configurados como entrada e/ou saída;
- outras características especiais como programação *in-circuit* serial, proteção por código, *watchdog timer* (temporizador cão de guarda), módulo CCP, comparador interno, USART,....

Pinagem do PIC16F628a:



Os pinos de RA0 à RA7 e de RB0 à RB7 podem ser configurados como entradas ou saídas digitais. A alimentação se dá no pino V_{dd} , ligado normalmente em 5V com faixa de tolerância de 2 à 6V e o pino V_{ss} é a referência de terra. O pino OSC1/CLKIN é utilizado para sinal de clock produzido por cristal ou um circuito externo e o pino OSC2/CLKOUT para sinal de clock por cristal (utilizado em conjunto com OSC1/CLKIN). O pino MCLR é uma entrada de sinal de reset em nível

baixo (zero). O PIC16F628a é composto pelos subsistemas digitais conforme diagrama de blocos abaixo:



Arquitetura Harvard

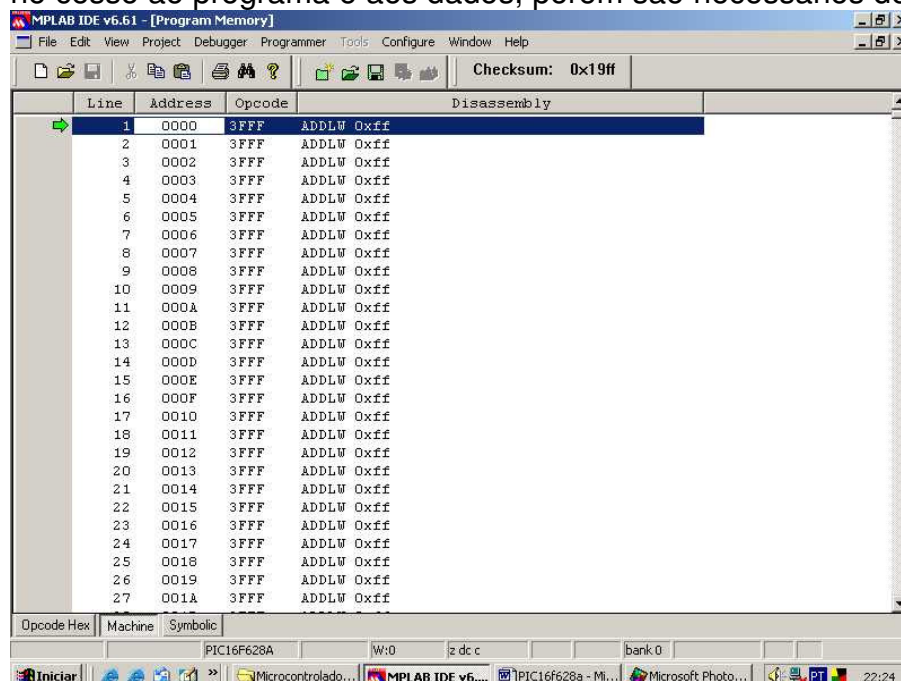
O PIC16F628a possui uma arquitetura denominada Harvard para acesso aos dados e às instruções do programa, a qual é caracterizada por utilizar um *bus* (via de acesso) para comunicação dos dados e outro *bus* distinto para comunicação com o programa. A maioria dos microcomputadores e microcontroladores no mercado utilizam um único *bus*, tanto para dados quanto para instruções do programa. Nota-se a diferença no uso desta arquitetura na visualização da memória do microcontrolador PIC: existe uma memória para o programa e outra para os dados

a serem utilizados. A figura a seguir mostra a memória RAM de um microcomputador PC, em que dados e instruções do programa estão na mesma memória, denominada memória principal (geralmente a RAM do computador). As letras abaixo, nas últimas linhas, mostram valores dos registradores especiais, como o DS (*Data Segment*) o qual indica o endereço inicial dos dados na memória e o CS (*Code Segment*) o qual indica o endereço inicial da memória no qual as instruções do programa corrente se iniciam.

```

0xCD5:01C0 25 4F 20 6E A3 6D 65 72-6F 20 64 65 20 73 82 72 %0 n.mero de s.r
0xCD5:01D0 69 65 20 64 6F 20 76 6F-6C 75 6D 65 20 82 20 25 ie do volume . %
0xCD5:01E0 31 2D 25 32 0D 0A 35 4E-6F 6D 65 20 64 65 20 61 1-%2..5Nome de a
0xCD5:01F0 72 71 75 69 76 6F 20 64-75 70 6C 69 63 61 64 6F rquivo duplicado
-a
0xCD5:0200 20 6F 75 20 61 72 71 75-69 76 6F 20 6E C6 6F 20 ou arquivo n.o
0xCD5:0210 65 6E 63 6F 6E 74 72 61-64 6F 0D 0A 25 43 61 6D encontrado..%Cam
0xCD5:0220 69 6E 68 6F 20 6F 75 20-6E 6F 6D 65 20 64 65 20 inho ou nome de
0xCD5:0230 61 72 71 75 69 76 6F 20-69 6E 76 A0 6C 69 64 6F arquivo inv.lido
0xCD5:0240 0D 0A 1D 45 73 70 61 87-6F 20 64 65 20 61 6D 62 ...Espa.o de amb
0xCD5:0250 69 65 6E 74 65 20 65 73-67 6F 74 61 64 6F 0D 0A iente esgotado..
0xCD5:0260 1C 45 72 72 6F 20 6E 61-20 63 72 69 61 87 C6 6F .Erro na cria..o
0xCD5:0270 20 64 6F 20 61 72 71 75-69 76 6F 0D 0A 1F 41 72 do arquivo...Ar
-a
0xCD5:0280 71 75 69 76 6F 20 64 65-20 6C 6F 74 65 20 6E C6 quivo de lote n.
0xCD5:0290 6F 20 65 78 69 73 74 65-6E 74 65 0D 0A 28 0D 0A o existente...(
0xCD5:02A0 49 6E 73 69 72 61 20 6F-20 64 69 73 63 6F 20 63 Insira o disco c
0xCD5:02B0 6F 6D 20 6F 20 61 72 71-75 69 76 6F 20 64 65 20 om o arquivo de
0xCD5:02C0 6C 6F 74 65 0D 0A 25 43-6F 6D 61 6E 64 6F 20 6F lote..%Comando o
0xCD5:02D0 75 20 6E 6F 6D 65 20 64-65 20 61 72 71 75 69 76 u nome de archiv
0xCD5:02E0 6F 20 69 6E 76 A0 6C 69-64 6F 0D 0A 10 41 63 65 o inv.lido...Ace
0xCD5:02F0 73 73 6F 20 6E 65 67 61-64 6F 20 0D 0A 2C 43 6F sso negado ...Co
-a
0xCD5:0300 6E 74 65 A3 64 6F 20 64-6F 20 64 65 73 74 69 6E nte.do do destin
0xCD5:0310 6F 20 70 65 72 64 69 64-6F 20 61 6E 74 65 73 20 o perdido antes
0xCD5:0320 64 61 20 63 A2 70 69 61-0D 0A 34 4E 6F 6D 65 20 da c.pia..4Nome
0xCD5:0330 64 65 20 61 72 71 75 69-76 6F 20 69 6E 76 A0 6C de arquivo inv.l
0xCD5:0340 69 64 6F 20 6F 75 20 61-72 71 75 69 76 6F 20 6E ido ou arquivo n
0xCD5:0350 C6 6F 20 65 6E 63 6F 6E-74 72 61 64 6F 0D 0A 1A .o encontrado...
0xCD5:0360 25 31 20 61 72 71 75 69-76 6F 28 73 29 20 63 6F %1 arquivo(s) co
0xCD5:0370 70 69 61 64 6F 28 73 29-0D 0A 0E 25 31 20 61 72 piado(s)...%1 ar
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0CD5 ES=0CD5 SS=0CD5 CS=0CD5 IP=0100 NU UP EI PL NZ NA PO NC
0CD5:0100 A06C69 MOV AL,1696C1 DS:696C=00
  
```

A figura abaixo mostra o *software* MPLab IDE com as janelas de visualização da memória Flash que contém o programa (janela à esquerda) e da memória RAM a qual contém os dados e registradores especiais (janela à direita) do PIC16F628a. Este tipo de arquitetura permite uma velocidade maior no acesso ao programa e aos dados, porém são necessários dois *bus*.



Instruções RISC

Na tabela abaixo observa-se o conjunto de instruções do PIC16F628a, com os mnemônicos, os operandos de cada instrução, um resumo descritivo, o número de ciclos do clock exigido, o código em linguagem de máquina, os bits do registrador de status que são afetados e anotações sobre cada instrução. O número de instruções é limitado, sendo denominado arquitetura RISC - *Reduced Instruction Set* (Grupo Reduzido de Instruções). O número de instruções é pequeno, porém cada instrução é formada por 14 bits que codificam uma operação e os operandos ou os dados sujeitos à respectiva operação. A tabela abaixo mostra o microcódigo para os microcontroladores da linha PIC, com a instrução *assembly* na primeira coluna, os operandos possíveis para a respectiva instrução, a descrição, o número de ciclos do clock que a instrução exige para ser executada e o código em linguagem de máquina.

MNEMÔNICOS	OPERANDOS	DESCRIÇÃO	CICLOS DO CLOCK	OPCODE DE 14 BITS
OPERAÇÕES DE REGISTRADORES ORIENTADAS A BYTE				
ADDWF	f, d	Soma o conteúdo do registrador “W” e de “f”	1	00 0111 dfff ffff
ANDWF	f, d	Operação lógica “E” entre o conteúdo de “W” e “f”	1	00 0101 dfff ffff
CLRF	f	Limpa o conteúdo de “f”	1	00 0001 lfff ffff
CLRW	-f,	Limpa o conteúdo do registrador “W”	1	00 0001 0xxx xxxx
COMF	d	Complemento do conteúdo de “f”	1	00 1001 dfff ffff
DECF	f, d	Decrementa o valor armazenado em “f”	1	00 0011 dfff ffff
DECFSZ	f, d	Decrementa o valor armazenado em “f”, pula a próxima instrução se for igual a “0”	1 (2)	00 1011 dfff ffff
INCF	f, d	Incrementa o valor armazenado em “f”	1	00 1010 dfff ffff
INCFSZ	f, d	Incrementa o valor armazenado em “f”, pula a próxima instrução se for igual a “0”	1 (2)	00 1111 dfff ffff
IORWF	f, d	Operação lógica “OU INCLUSIVO” do conteúdo do registrador “W” com o conteúdo de “f”	1	00 0100 dfff ffff
MOVF	f, d	Move o conteúdo de “f” para o registrador “W”	1	00 1000 dfff ffff
MOVWF	f	Move o conteúdo do registrador “W” para “f”	1	00 0000 lfff ffff
NOP	-f,	Sem operação	1	00 0000 0xx0 0000

RLF	d	Rotaciona os bits armazenados em “f” para a esquerda com Carry	1	00 1101 dfff ffff
RRF	f, d	Rotaciona os bits armazenados em “f” para a direita com Carry	1	00 1100 dfff ffff
SUBWF	f, d	Subtração do conteúdo em “W” de “f”	1	00 0010 dfff ffff
SWAPF	f, d	Troca o primeiro nibble (4 bits) com o último nibble armazenados em “f”	1	00 1110 dfff ffff
XORWF	f, d	Operação lógica “OU EXCLUSIVO” do conteúdo de “W” com o conteúdo de “f”	1	00 0110 dfff ffff
OPERAÇÕES DE REGISTRADORES ORIENTADAS A BIT				
BCF	f, b	Muda para “0” o valor do bit “b” de “f”	1	01 00bb bfff ffff
BSF	f, b	Muda para “1” o valor do bit “b” de “f”	1	01 01bb bfff ffff
BTFSC	f, b	Verifica o valor do bit “b” do registrador “f” e pula a próxima instrução se for igual a “0”	1 (2)	01 10bb bfff ffff
BTFSS	f, b	Verifica o valor do bit “b” do registrador “f” e pula a próxima instrução se for igual a “1”	1 (2)	01 11bb bfff ffff
OPERAÇÕES DE CONTROLE E LITERAIS				
ADDLW	k	Adiciona um número “k” ao conteúdo do registrador “W”	1	11 111x kkkk kkkk
ANDLW	k	Operação lógica “E” de um número “k” com “W”	1	11 1001 kkkk kkkk
CALL	k	Chamada de uma subrotina indicada por “k”	2	10 0kkk kkkk kkkk
CLRWDT		Limpa o temporizador cão de guarda	1	00 0000 0110 0100
GOTO	k	Vai ao endereço dado por “k”	2	10 1kkk kkkk kkkk
IORLW	k	Operação OU INCLUSIVO de um número “k” com “W”	1	11 1000 kkkk kkkk
MOVLW	k	Move um número de valor “k” para “W”	1	11 00xx kkkk kkkk
RETFIE		Retorno de uma interrupção	2	00 0000 0000 1001
RETLW	k	Retorna com um valor “k” no registrador “W”	2	11 01xx kkkk kkkk
RETURN		Retorno de uma subrotina ao programa principal	2	00 0000 0000 1000
SLEEP		Entra no modo standby	1	00 0000 0110 0011

SUBLW	k	Subtrai o conteúdo do registrador “W” de um número “k”	1	11 110x kkkk kkkk
XORLW	k	Operação lógica “OU EXCLUSIVO” de um número “k” com o conteúdo do registrador “W”	1	11 1010 kkkk kkkk

Obs.: Nos operandos, as letras f, d, b, k indicam:

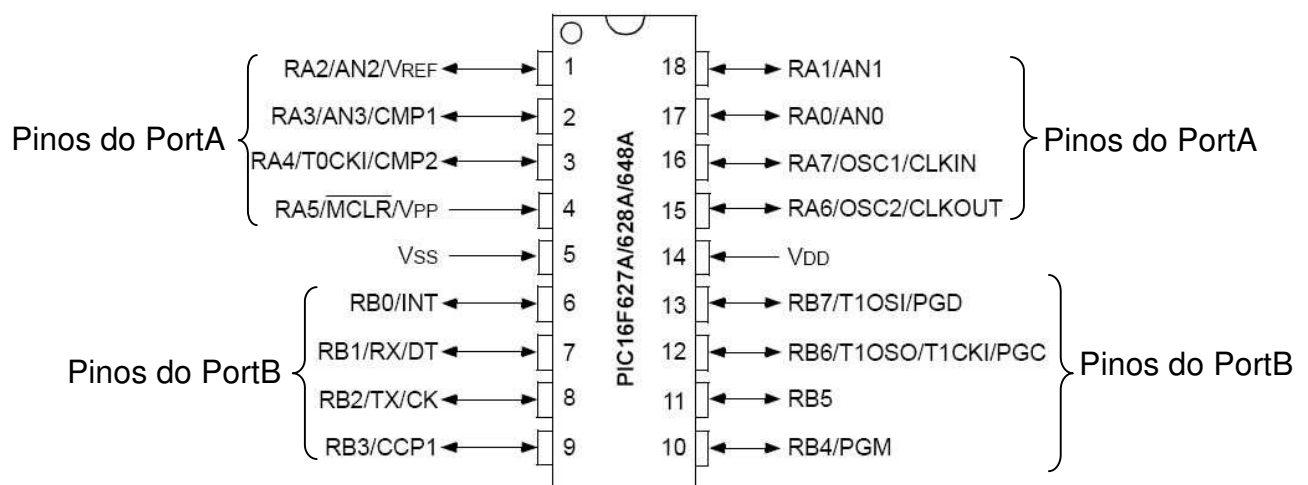
- f: registrador na memória RAM, como o PORTA, TRISB, STATUS, ou uma posição qualquer;
- d: direção para a qual pode se designada a função;
- b: o número de um bit dentro de um registrador. Para o PIC16F628a, são de 0 a 8, porque os registradores possuem 8 bits;
- k: um literal, que envolve letras (ASCII), números no formato decimal, hexadecimal ou binário, até o limite especificado pelo tamanho do registrador.

“Ports” de entrada e saída

Os microcontroladores da linha PIC possuem pinos que podem ser configurados como entrada ou saída digital. Cada conjunto de oito destes pinos é denominado Port, geralmente classificado por uma letra do alfabeto (PortA, PortB, PortC,...)

“Ports” de entrada e saída

Os microcontroladores da linha PIC possuem pinos que podem ser configurados como entrada ou saída digital. Cada conjunto de oito destes pinos é denominado Port, geralmente classificado por uma letra do alfabeto (PortA, PortB, PortC,...). Para isto, existem implementados na memória RAM do PIC registradores especiais para configurar e ler/modificar o valor destes pinos. No PIC16F628a, dois destes registradores, o TRISA e o TRISB, são responsáveis pela configuração destes pinos para funcionamento como entrada ou saída digital. A abreviatura TRIS para estes registradores provem de *tri-state* (três estados) porque alguns pinos podem funcionar de três modos diferentes: como entrada digital, saída digital ou com outra função específica.



Por exemplo: os pinos RB6 e RB7 do PIC16F628a podem funcionar como entrada digital, saída digital ou terminal de comunicação serial para programar e ler o programa presente na memória do PIC. Outros modelos de microcontroladores da linha PIC possuem pinos que podem funcionar como E/S digital e entrada analógica, ou porta de comunicação de dados,.... Como cada registrador em um PIC pode armazenar no máximo oito bits, os registradores TRISA e TRISB podem configurar no máximo oito pinos cada um, com o valor de cada bit nestes registradores correspondendo à configuração de um pino. O PIC16F628a possui treze pinos de E/S digital, então o registrador TRISB é responsável pela configuração de 8 pinos (RB0 a RB7) e o registrador TRISA para o restante (RA0 a RA4, e três bits não são utilizados neste registrador), como demonstrado nas tabelas abaixo:

	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
TRISA								

	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
TRISB								

Obs.: Os bits marcados com um “X” não são utilizados, porque o PIC16F628a possui somente 13 pinos para E/S digital. A tentativa de alteração do valor destes bits não influencia o funcionamento do microcontrolador.

Cada registrador acima configura um pino como entrada digital se o bit respectivo possuir valor “1”; ou saída digital, se o bit respectivo possuir o valor “0”. Os exemplos abaixo mostram os valores que devem estar presentes nos registradores TRISA e TRISB dada a respectiva configuração dos pinos:

- pinos RA0 a RA7 configurados como entradas digitais e pinos RB0 a RB7 como saídas digitais:

TRISA	1	1	1	1	1	1	1	1
-------	---	---	---	---	---	---	---	---

TRISB	0	0	0	0	0	0	0	0
-------	---	---	---	---	---	---	---	---

- pinos RA0, RA2, RA4 e RA6 como entradas digitais e o restante como saídas:

TRISA	0	1	0	1	0	1	0	1
-------	---	---	---	---	---	---	---	---

TRISB	0	0	0	0	0	0	0	0
-------	---	---	---	---	---	---	---	---

- pino RB4 como saída digital e o restante como entrada:

TRISA	1	1	1	1	1	1	1	1
-------	---	---	---	---	---	---	---	---

TRISB	1	1	1	0	1	1	1	1
-------	---	---	---	---	---	---	---	---

- pinos RB0 e RB7 como entradas digitais e o restante como saídas digitais:

TRISA	1	0	1	0	1	0	1	0
-------	---	---	---	---	---	---	---	---

TRISB	0	1	0	1	0	1	0	1
-------	---	---	---	---	---	---	---	---

- pinos de RA0 a RA7 como saídas digitais e o restante como entradas digitais:

TRISA	0	0	0	0	0	0	0	0
-------	---	---	---	---	---	---	---	---

TRISB	1	1	1	1	1	1	1	1
-------	---	---	---	---	---	---	---	---

- pinos RA0, RA1, RA2, RB0, RB1, RB2, RB3 e RB4 como entradas digitais e o restante como saídas:

TRISA				0	0	1	1	1
-------	--	--	--	---	---	---	---	---

TRISB	0	0	0	1	1	1	1	1
-------	---	---	---	---	---	---	---	---

A visualização do valor nos registradores pode ser feita no software MPLAB pela janela da memória RAM ou pela janela dos registradores de funções especiais, que exibe diretamente a denominação dos registradores especiais e seus valores em quatro formatos: decimal, ASCII, hexadecimal e binário. A memória RAM exibe o conteúdo dos registradores especiais localizados na mesma, pelo seu endereço, e das posições de memória restantes. Alguns registradores especiais não se localizam na memória RAM, mas estão implementados no próprio circuito da CPU do microcontrolador PIC, como o *Work Register* – Registrador de Trabalho ou registrador W. O endereços na memória RAM dos registradores TRISA e TRISB são 85H e 86H.

Geralmente o início da maioria dos programas para o PIC contém instruções para modificar os valores de TRISA e TRISB, ou seja, para configurar quais pinos funcionarão como entrada digital ou saída digital.

