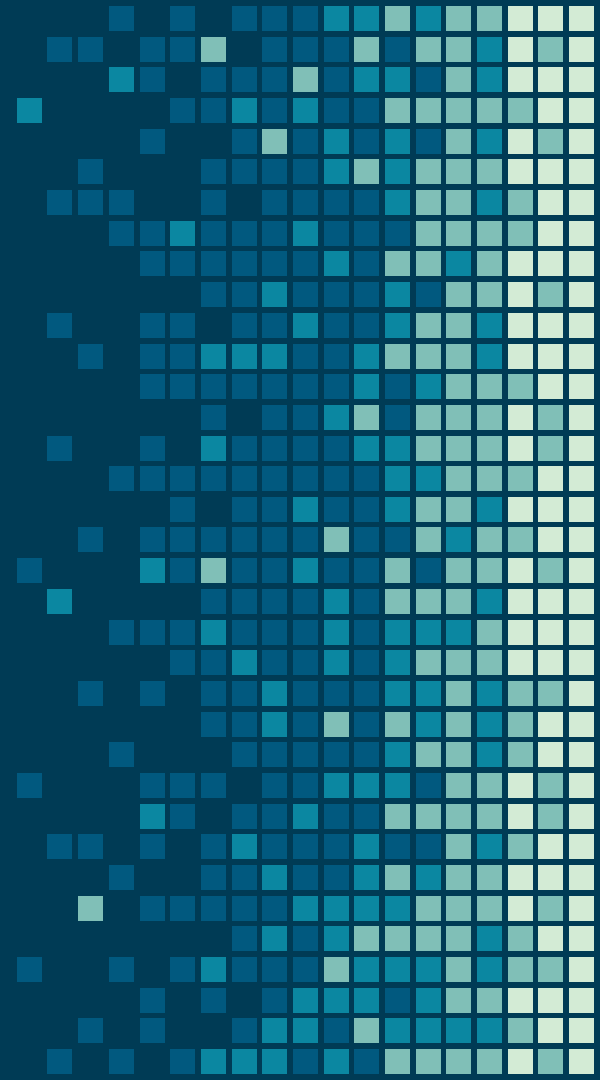


# ASSEMBLER



# Conteúdo

Conjunto de instruções

Conjunro reduzido de instruções  
Trinta e cinco.

Diretivas Assembler

Descrição da instrução

# 1. Conjunto de instruções



# Conjunto de instruções

**W: work (registrador de trabalho ou acumulador)**

**F: file (registradores especiais e/ou de uso geral | posição de memória)**

**L: literal (constante, número qualquer)** Utilizado como L nas instruções e k nos argumentos

**D: destination (local onde o resultado de uma operação será armazenado).**

**B: bit (bits dentro dos registradores, dentro de um byte, flags)**

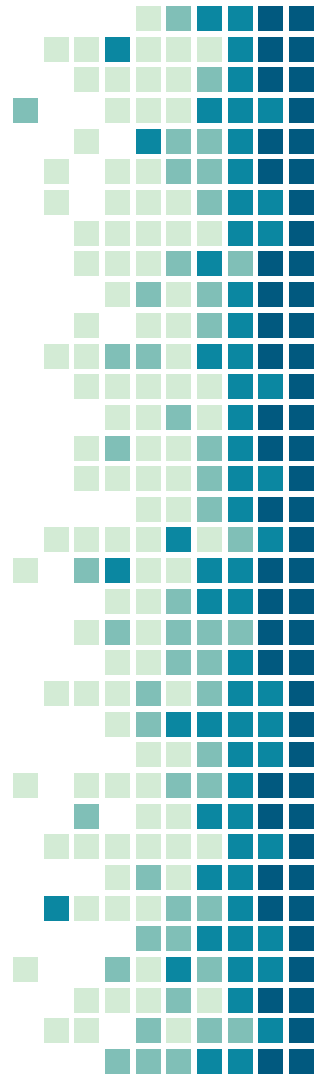
**T: test (utilizado para teste de bits)**

**S: skip (pulo, salto, desvio condicional)**

**S: set (setar, tornar nível alto)**

**C: clear (limpar, tornar nível baixo)**

**Z: zero (testar se é equivalente a zero)**

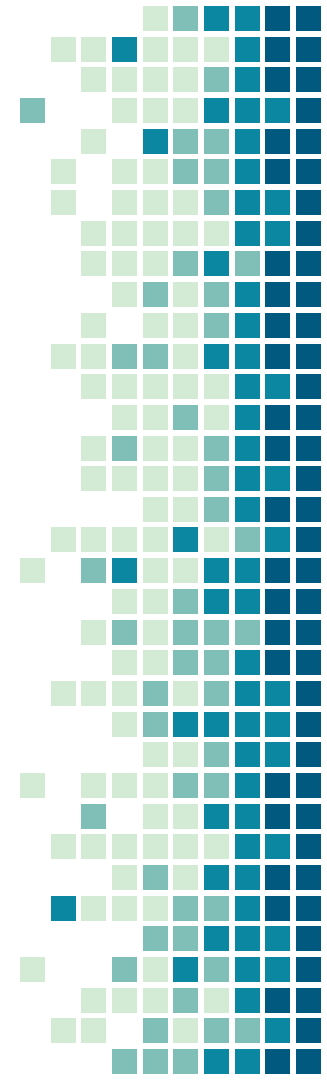


## Representação Numérica

Decimal:                    D'??'    ou    .??  
Ex: D'20' ou .20

Hexadecimal:            H'??' , 0X??    ou    ??H  
Ex: H'2F' ou 2FH    0X2F

Binário:                    B'?????????'  
Ex: B'01101101'



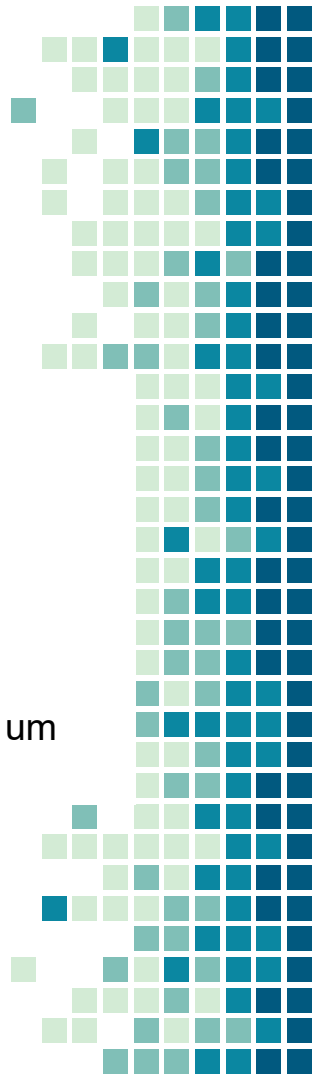
## Mnemônicos

Uma instrução do PIC é normalmente da seguinte forma:

**INST** **OP1**, **OP2**

Onde **INST** é o OP-CODE da instrução e **OP1** e **OP2** são os operandos. Uma instrução pode ter zero, um ou dois operandos.

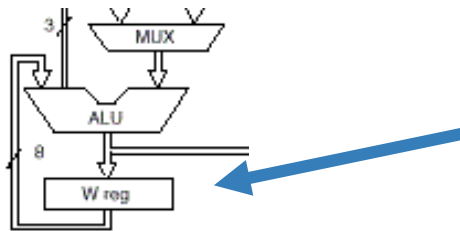
Um operando pode ser um Literal (k), um registrador (f), bit (b onde  $0 \leq b \leq 7$ ) ou um destino (d), onde d=0 work e d=1 file;



## Mnemônicos

O nome das instruções do PIC são construídos de acordo com certa lógica. Entender essa lógica pode facilitar muito o trabalho do programador, pois com esse entendimento é mais fácil decorar o set de instruções do PIC e os seus operandos, e diminuir eventuais consultas a documentação na hora de ler um código. Os termos abaixo são usados na construção dos nomes de instrução:

**(W)ork** – Refere-se ao **registrador temporário W**. Observando o esquema interno do PIC é possível perceber que esse registrador está localizado na [saída da ULA](#). Quase todas as operações sobre bytes, de alguma forma, envolvem esse registrador. As instruções que manipulam esse registrador diretamente tem o W na sua formação.



## Mnemônicos

Ex.: Para fazer uma atribuição do tipo:

**A=25**

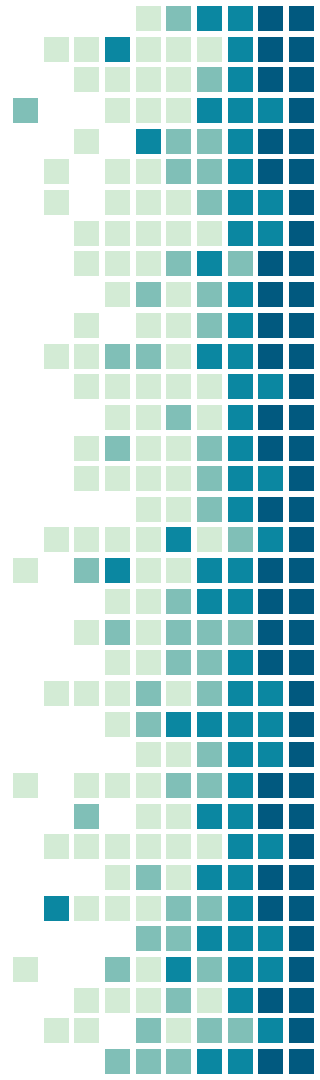
É necessário carregar W com o valor do literal 25 e depois carregar o conteúdo do registrador W na posição representada por A.

A `equ` H'007'

...

<code>MOVLW d'25'</code>	;Move o (L)iteral (25 no caso) para (W)ork
<code>MOVWF A</code>	; move (W)ork para (F)ile (A no caso)

Observe que as referencias a W, são implícitas, ou seja, não aparecem nos operandos.





## Mnemônicos

**(L)iteral** – Refere-se a um **valor numérico**. Pode ser escrito na forma binária, decimal ou hexadecimal. As instruções que operam com literais apresentam o L na sua formação.

**(F)ile** – Refere-se a uma **posição de memória** (File Register). As instruções que operam diretamente com registrador apresentam um F na sua formação.

**(B)it** – Refere-se a um bit específico do registrador F, as instruções que operam diretamente com bit tem um B na sua formação.

**(S)et/(C)lear** – Refere-se a ação de por em nível lógico um (set) ou zero (clear) um determinado registrador (neste caso o pic só permite zerar um registrador específico) ou bit específico do registrador.

As instruções que setam um determinado bit **apresentam S na sua formação**.

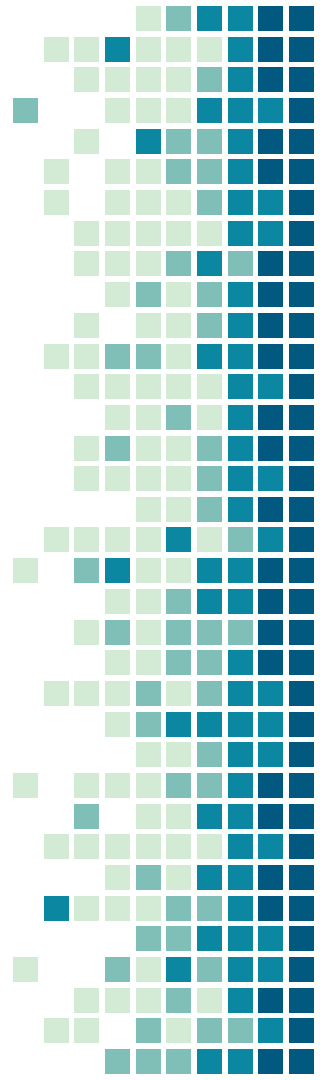
São exemplos:

**BCF** 0x5h,3

; Bit Clear File – zera Bit 3 de 0x5h

**BSF** 0x5h,3

; Bit Set File- Faz o bit 3 de 0x5h igual a 1



## Mnemônicos

**(T)est** – Refere-se ao ato de testar uma determinada condição que está associada a um bit específico do registrador F. **As instruções que fazem testes tem o T na sua formação.**

**(S)kip** – Refere-se ao ato de promover algum desvio no fluxo sequencia de instrução com base em alguma condição. **A instruções que realizam saltos tem um S na sua formação.**

**OBS.:** Para desvios incondicionais normais existem as instruções GOTO e CALL.

**(Z)ero** – Refere-se ao teste se o resultado de uma operação especifica é zero, normalmente verificando o registrador de STATUS (03h,2). As instruções que fazem esse tipo de teste apresentam Z na sua formação. Essas instruções normalmente trabalham com bytes.

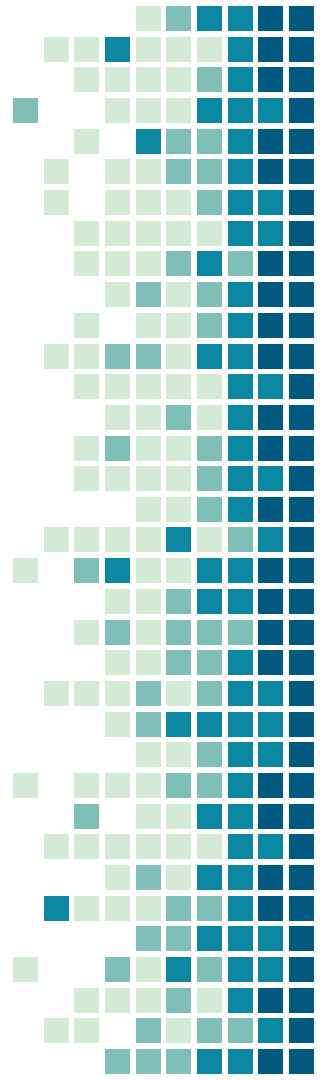
Exemplos:

**BTFSC** 0x3h,2

;Bit Test File Skip Clear – se bit 2 de 3h igual a zero  
;salta a próxima instrução

**DECFSZ** 0x7h

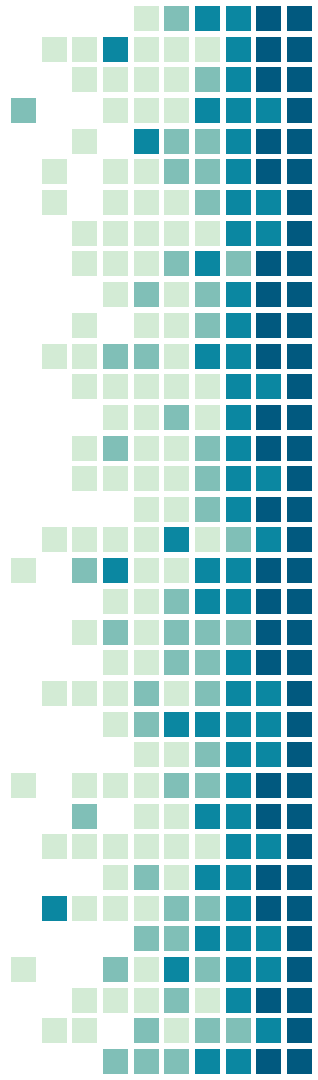
;Decrementa (DEC) File Skip Zero – se o resultado do  
;decremento for zero salta a próxima instrução



## Mnemônicos

Combinando os termos acima com um dos termos seguir com os termos acima formam-se as instruções do PIC que trabalham com bytes:

- ADD** Adiciona dois operandos que podem ser W, F ou L. O resultado sempre será armazenado em W ou F
- AND** Faz o E lógico entre dois operandos que podem ser W, F ou L. O resultado sempre será armazenado em W ou F
- CLR** Torna zero Work ou um determinado registrador F
- COM** Complemento de registrador F
- DEC** Decrementa
- INC** Incrementa
- IOR** OU inclusivo
- MOV** Move um dado ou literal
- RL** Rotaciona para a esquerda (x2)
- RR** Rotaciona para a direita (x2)
- SUB** Subtração
- SWAP** Troca interna entre o bits mais ou menos significativos
- XOR** OU exclusivo

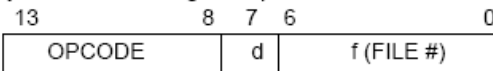


## PIC 16F628

O microcontrolador PIC16F628A apresenta em seu nível ISA 35 instruções. A maioria das instruções são executadas em apenas um ciclo de máquina, lembrando que o um ciclo de máquina, corresponde ao clock de entrada dividido por 4. O Datasheet do PIC16F628A, divide o conjunto de instruções do PIC em três tipos:

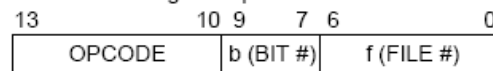
- **Orientadas a bit (Bit-Oriented):** Operam com bits de um determinado registrador;
- **Orientadas a byte (Byte-Oriented):** Operam com registradores completos (bytes);
- **Literal ou controle (Literal and control):** Instruções que usam literais (números ou endereços) como operando.

### Byte-oriented file register operations



d = 0 for destination W  
d = 1 for destination f  
f = 7-bit file register address

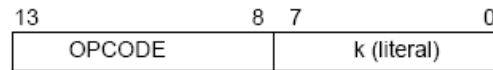
### Bit-oriented file register operations



b = 3-bit bit address  
f = 7-bit file register address

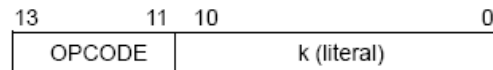
### Literal and control operations

#### General



k = 8-bit immediate value

#### CALL and GOTO instructions only

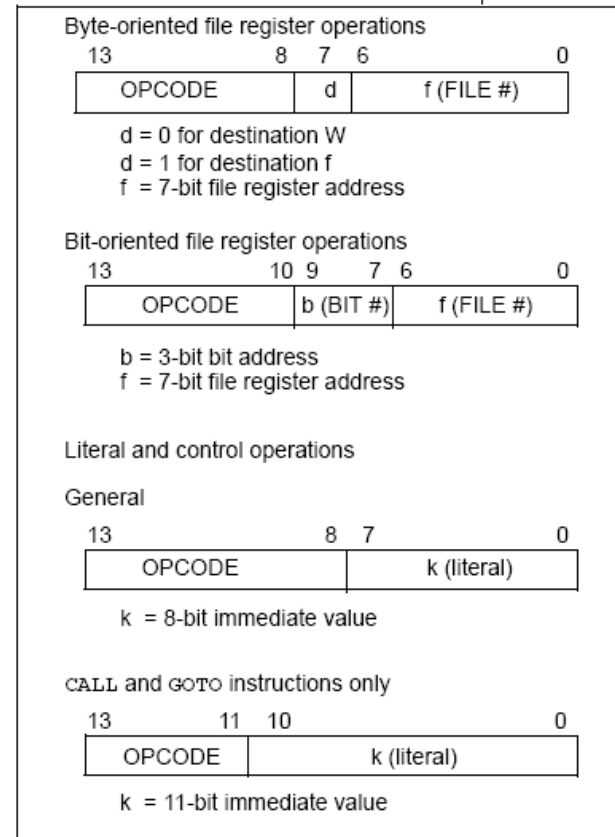


k = 11-bit immediate value

## Formato do conjunto de instruções

Cada instrução do PIC16F628A tem o tamanho de 14 bits, onde alguns destes bits são usados para especificar o **OPCODE** (nome dado ao "comando" assembly), e de zero até dois operandos. Por conveniência, neste material a ultima categoria foi dividida em literal e controle.

O formato das instruções pode variar de acordo com o tipo, abaixo o formato da instrução do PIC para cada tipo. A figura abaixo mostra os formatos de instrução do PIC16F628A.



## Mneumônicos

Instruções do PIC16F628A, com os seus respectivos operandos

## Orientadas a bit

			13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mnemônico	ciclos	Descrição	OPCODE				BIT			REGISTRADOR						
							(0-7)			(00h - 127h)						
<a href="#">BCF f,b</a>	1	Limpa (zera) o bit b do registrador f	0	1	0	0	B	B	B	F	F	F	F	F	F	F
<a href="#">BSF f,b</a>	1	Seta (igual a 1) o bit b do registrador f	0	1	0	1	B	B	B	F	F	F	F	F	F	F
<a href="#">BTFSC f,b</a>	1(2)	Testa o Bit b de f, pula se zero(clear)	0	1	1	0	B	B	B	F	F	F	F	F	F	F
<a href="#">BTFSS f,b</a>	1(2)	Testa o Bit b de f, pula se um(set)	0	1	1	1	B	B	B	F	F	F	F	F	F	F

### Legenda:

X = não importa o valor do bit

k = o bit pertence a um literal

f = O bit pertence a um endereço de registrador

d = onde o resultado será armazenado W(d=0) ou F(d=1)

Instruções do PIC16F628A, com os seus respectivos operandos

## Orientadas a byte

			13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mnemônico	ciclos	Descrição	OPCODE						D	REGISTRADOR						
										(00h - 127h)						
NOP	1	Gasta um ciclo	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CLRWDT	1	Zera o timer <i>watch dog</i>	0	0	0	0	0	0	0	1	1	0	0	1	0	0
<a href="#">MOVWF f</a>	1	Copia o conteúdo de w em f	0	0	0	0	0	0	1	F	F	F	F	F	F	F
<a href="#">CRLW</a>	1	Zera o conteúdo do registrador W	0	0	0	0	0	1	0	X	X	X	X	X	X	X
CLRF f	1	Zera o conteúdo de f	0	0	0	0	0	1	1	F	F	F	F	F	F	F
SUBWF f,d	1	subtrai W de f e armazena em d( $d \leq f - W$ )	0	0	0	0	1	0	D	F	F	F	F	F	F	F
DECF f,d	1	decrementa f e armazena em d( $d \leq f - 1$ )	0	0	0	0	1	1	D	F	F	F	F	F	F	F
IORWF f,d	1	OU normal (Inclusivo) de W com F ( $d \leq f \text{ OR } W$ )	0	0	0	1	0	0	D	F	F	F	F	F	F	F

### Legenda:

X = não importa o valor do bit

k = o bit pertence a um literal

f = O bit pertence a um endereço de registrador

d = onde o resultado será armazenado W(d=0) ou F(d=1)

# Mneumônicos

Instruções do PIC16F628A, com os seus respectivos operandos

## Orientadas a byte

			13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mnemônico	ciclos	Descrição	OPCODE							D	REGISTRADOR (00h - 127h)					
ANDWF f,d	1	E entre W e F ( $d \leq f \text{ AND } W$ )	0	0	0	1	0	1	D	F	F	F	F	F	F	F
XORWF f,d	1	OU exclusivo entre W e F ( $d \leq f \text{ XOR } W$ )	0	0	0	1	1	0	D	F	F	F	F	F	F	F
ADDWF f,d	1	Adiciona W com F ( $d \leq f + W$ )	0	0	0	1	1	1	D	F	F	F	F	F	F	F
<a href="#">MOVF f,d</a>	1	Move F ( $d \leq f$ )	0	0	1	0	0	0	D	F	F	F	F	F	F	F
COMF f,d	1	Complemento de f ( $d \leq \text{NOT } f$ )	0	0	1	0	0	1	D	F	F	F	F	F	F	F
INCF f,d	1	incrementa f e armazena em d ( $d \leq f - 1$ )	0	0	1	0	1	0	D	F	F	F	F	F	F	F
<a href="#">DECFSZ f,d</a>	1(2)	Decrementa f ( $d \leq f - 1$ ) e salta se zero	0	0	1	0	1	1	D	F	F	F	F	F	F	F
<a href="#">RRF f,d</a>	1	Rotaciona F para direita com carry out	0	0	1	1	0	0	D	F	F	F	F	F	F	F
<a href="#">RLF f,d</a>	1	Rotaciona F para esquerda com carry out	0	0	1	1	0	1	D	F	F	F	F	F	F	F
<a href="#">SWAPF f,d</a>	1	Troca os nibbles mais e menos significativos de f	0	0	1	1	1	0	D	F	F	F	F	F	F	F
<a href="#">INCFSZ f,d</a>	1(2)	incrementa f ( $d \leq f - 1$ ) e salta se zero	0	0	1	1	1	1	D	F	F	F	F	F	F	F



# Mneumônicos

Instruções do PIC16F628A, com os seus respectivos operandos

**De controle (quando efetuam desvios gastam 2 ciclos)**

			13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mnemônico	ciclos	Descrição	OPCODE						K literal							
<a href="#">RETURN</a>	2	Retorna de uma subrotina chamada por CALL	0	0	0	0	0	0	0	0	0	0	1	0	0	0
<a href="#">RTFIE</a>	2	Retorna de uma interrupção	0	0	0	0	0	0	0	0	0	0	1	0	0	1
SLEEP	0	Põe o controlador em stand-by	0	0	0	0	0	0	0	1	1	0	0	0	1	1
Mnemônico	ciclos	Descrição	OPCODE			K posição de memória										
<a href="#">CALL k</a>	2	Salva PC+1 na pilha e faz PC=k (salta a execução para o endereço k)	1	1	0	K	K	K	K	K	K	K	K	K	K	K
<a href="#">GOTO k</a>	2	Pula para o endereço k (11 bits) usa 2 ciclos	1	0	1	K	K	K	K	K	K	K	K	K	K	K

## Legenda:

X = não importa o valor do bit

k = o bit pertence a um literal

f = O bit pertence a um endereço de registrador

d = onde o resultado será armazenado W(d=0) ou F(d=1)

# Mneumônicos

Instruções do PIC16F628A, com os seus respectivos operandos

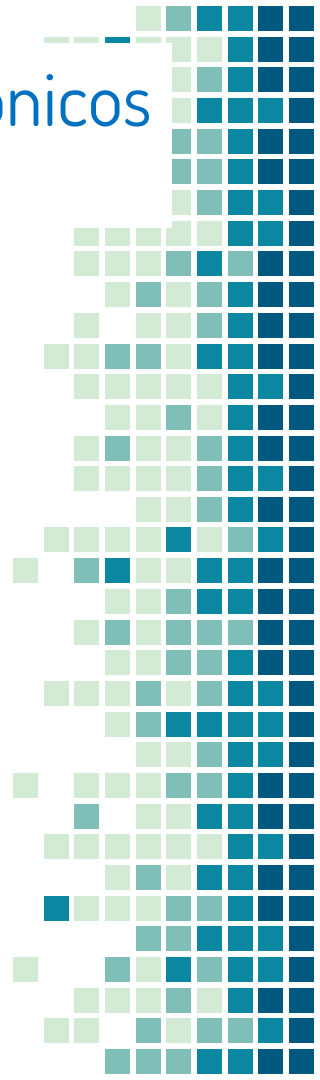
**De controle (quando efetuam desvios gastam 2 ciclos)**

			13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mnemônico	ciclos	Descrição	OPCODE						K literal							
<a href="#">MOVLW k</a>	1	Move literal to W (W = k)	1	1	0	0	X	X	K	K	K	K	K	K	K	K
<a href="#">RETLW K</a>	1	faz W=k e retorna de uma subrotina	1	1	0	1	X	X	K	K	K	K	K	K	K	K
IORLW k	1	W = k OR W	1	1	1	0	0	0	K	K	K	K	K	K	K	K
ANDLW k	1	W = k AND W	1	1	1	0	0	1	K	K	K	K	K	K	K	K
XORLW k	1	W = k AND W	1	1	1	0	1	0	K	K	K	K	K	K	K	K
SUBLW k	1	W = k - W	1	1	1	1	0	X	K	K	K	K	K	K	K	K
ADDLW k	1	W = k + W	1	1	1	1	1	X	K	K	K	K	K	K	K	K

# Conjunto de instruções

Mnemonic, Operands	Description	Cycles	14-Bit Opcode		Status Affected	Notes
			MSb	LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff	Z	1,2
CLRF	f	Clear f	1	00 0001 1fff ffff	Z	2
CLRW	-	Clear W	1	00 0001 0xxxx xxxxx	Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff	Z	1,2
DECF	f, d	Decrement f	1	00 0011 dfff ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1 (2)	00 1011 dfff ffff		1,2,3
INCF	f, d	Increment f	1	00 1010 dfff ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1 (2)	00 1111 dfff ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff	Z	1,2
MOVF	f, d	Move f	1	00 1000 dfff ffff	Z	1,2
MOVWF	f	Move W to f	1	00 0000 1fff ffff		
NOP	-	No Operation	1	00 0000 0xx0 0000		
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01 00bb bfff ffff		1,2
BSF	f, b	Bit Set f	1	01 01bb bfff ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01 10bb bfff ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01 11bb bfff ffff		3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add literal and W	1	11 111x kkkk kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11 1001 kkkk kkkk	Z	
CALL	k	Call subroutine	2	10 0kkk kkkk kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00 0000 0110 0100	$\overline{TO}, \overline{PD}$	
GOTO	k	Go to address	2	10 1kkk kkkk kkkk		
IORLW	k	Inclusive OR literal with W	1	11 1000 kkkk kkkk	Z	
MOVLW	k	Move literal to W	1	11 00xx kkkk kkkk		
RETFIE	-	Return from interrupt	2	00 0000 0000 1001		
RETLW	k	Return with literal in W	2	11 01xx kkkk kkkk		
RETURN	-	Return from Subroutine	2	00 0000 0000 1000		
SLEEP	-	Go into standby mode	1	00 0000 0110 0011	$\overline{TO}, \overline{PD}$	
SUBLW	k	Subtract W from literal	1	11 110x kkkk kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11 1010 kkkk kkkk	Z	

## Mneumônicos





# Bibliografia

[www.microchip.com/](http://www.microchip.com/)

# Créditos

<http://www.boscojr.com/programacao/PIC/pic4.html>

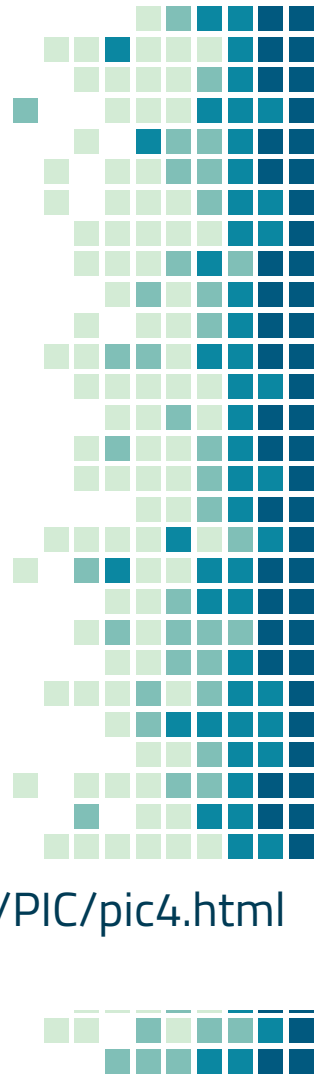
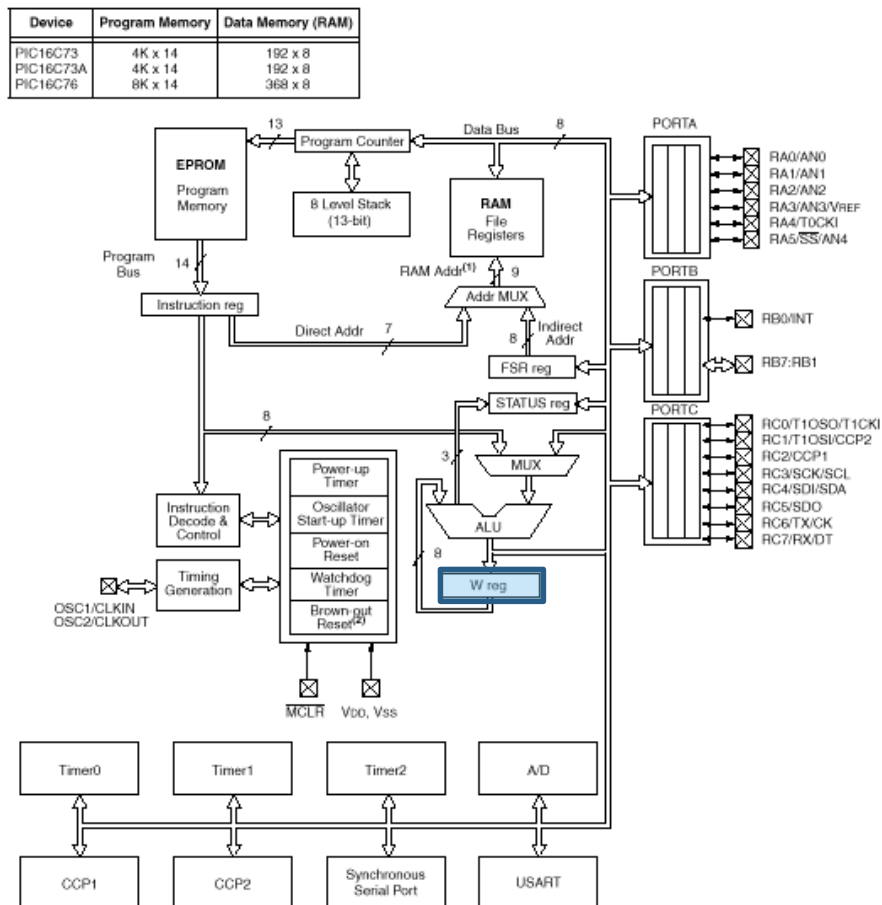


FIGURE 3-2: PIC16C73/73A/76 BLOCK DIAGRAM



[Voltar para Mnemônicos](#)



# Prof.:

Alberto Willian Mascarenhas

[awmascarenhas@gmail.com](mailto:awmascarenhas@gmail.com)

<https://sites.google.com/site/awmascarenhas/>



# OBRIGADO!

Perguntas ?