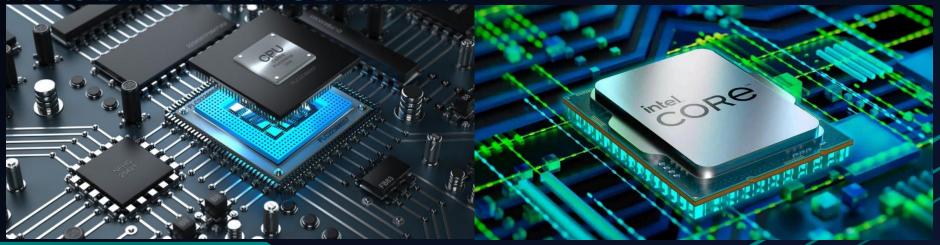


LENGUAJE ENSAMBLADOR

M. ENG. EVERT DE LOS RÍOS TRUJILLO PROGRAMA DE INGENIERÍA ELECTRÓNICA FACULTAD DE INGENIERÍA



OBJETIVOS:

 Conocer el repertorio de instrucciones en Ensamblador.

 Comprender el funcionamiento de las principales instrucciones.

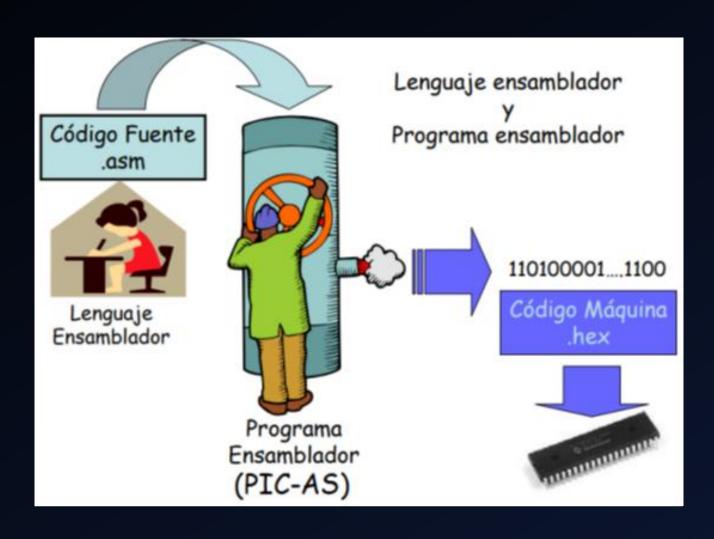
 Escribir el código básico en lenguaje Ensamblador.



LENGUAJE ENSAMBLADOR:

- El único lenguaje que entienden los microcontroladores es el código máquina formado por ceros y unos del sistema binario.
- El lenguaje ensamblador expresa las instrucciones de una forma más natural al hombre a la vez que muy cercana al microcontrolador, ya que cada una de esas instrucciones se corresponde con otra en código máquina.
- El lenguaje ensamblador trabaja con nemónicos, que son grupos de caracteres alfanuméricos que simbolizan las órdenes o tareas a realizar.
- La traducción de los nemónicos a código máquina entendible por el microcontrolador la lleva a cabo un programa ensamblador.
- El programa escrito en lenguaje ensamblador se denomina código fuente (*.asm o *.s). El programa ensamblador proporciona a partir de este fichero el correspondiente código máquina, que suele tener la extensión *.hex.

LENGUAJE ENSAMBLADOR:



PARÁMETROS DE LAS INSTRUCCIONES:

SIMBOLO	REFERENCIA	MODO	EJEMPLO
k	Literal	constante	movlw k
f	Posición de RAM	variable	clrf f
d	Destino	d=0 wreg d=1 f	movf f,d
b	Posición del bit	b=(0-7)	bsf f,b

Radix	Format		
Binary	Digits 0 and 1 followed by B		
Octal	Digits 0 to 7 followed by 0, Q, o or q		
Decimal	Digits 0 to 9 followed by D, d or nothing		
Hexadecimal	Digits 0 to 9, A to F preceded by 0x or followed by H or h		

TRANSFERENCIA DE DATOS					
INSTRUCCIÓN	ACCIÓN	EJEMPLO			
movlw k	W= k	movlw 25			
movwf f	f= W	movwf PORTB			
movf f,d	d=f; (W, f)	movf PORTC,W			

ORIENTADAS A BITS					
INSTRUCCIÓN	ACCIÓN	EJEMPLO			
bsf f,b	f(b)= 1	bsf PORTB,RB5			
bcf f,b	f(b)= 0	bcf PORTB,RB0			

ARITMÉTICAS Y LÓGICAS						
INSTRUCCIÓN	ACCIÓN	EJEMPLO				
addlw k	W= k+W	addlw 45				
addwf f,d	d=f+W; $d(f,W)$	addwf R1,f				
sublw k	W= k-W	sublw 21				
subwf f,d	d=f-W; $d(f,W)$	subwf R1,f				
andlw k	W= k&W	andlw 0x4B				
andwf f,d	d= f&W ; d(f,W)	andwf R1,f				

CONTROL DE FLUJO (SALTOS)				
INSTRUCCIÓN	ACCIÓN	EJEMPLO		
goto k	Salto incondicional a la dirección k.	goto ciclo		
call k	Salto incondicional a sub-rutina.	call rutina		
return	Salto incondicional, vuelta de sub-rutina.	return		
btfss f,b	Salto condicional de una posición si f(b)=1.	btfss PORTC,RC2		
btfsc f,b	Salto condicional de una posición si f(b)=0.	btfsc PORTC,RC4		
decfsz f	f=f-1 y Salto condicional de una posición si f=0.			

REPERTORIO DE INSTRUCCIONES:

		ription Cycles 14-Bit Opcode		Status	Notes			
		5 5	MSb	ß.		LSb	Affected	(S) (S)
BYTE-ORIENTED FILE REGISTER OPERATIONS								
f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
f, d	AND W with f	1	00				Z	1,2
f	Clear f	1	00	0001	lfff	ffff	Z	2
	Clear W	1	00	0001	0xxx	xxxx	Z	
f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
f. d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
	Move f	1	00	1000		200 (1800) and (1800)	Z	1,2
f	Move W to f	1	00	0000				(8)
		1	00					
		The state of the s	15.3		70,000	(UTUTAL)	С	1,2
			00			PERSONAL PROPERTY.	06.546	1,2
			00				155 T-155	1,2
			48.0				0,00,2	1,2
		1	72.753			PARTY PARTY	7	1,2
		2 2	20.0):::::::::::::::::::::::::::::::::::::	77.7.7		7	
		4	01	001-1-	1.555			4.2
200			1000	200000000000000000000000000000000000000		67 727 77		1,2 1,2
		17770000000	22233					3
205 2000			22.20					3
		1 (2)	01	1155	biii	IIII	(2)	3
AD CO	NIROL OPERATIONS	210					200	200
k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
k	AND literal with W		11	1001	kkkk	kkkk	Z	
k	Call subroutine	2	10	0kkk	kkkk	kkkk	ex 35 55	
	Clear Watchdog Timer	1	00	0000	0110	0100	TO,PD	
k	Go to address	2	10	1kkk	kkkk	kkkk	300	
k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
k	Move literal to W	1	11	00xx	kkkk	kkkk	XX877	
200	Return from interrupt	2	00	0000	0000	1001		
k	Return with literal in W		11	01xx				
100	Return from Subroutine	2	00	0000	0000	1000		
100		1	00	0000			TO.PD	
			6.000					
k		1	2000				The control of the co	
	f, d f, d f, d f, d f, d f, d f, d f, d	f, d Add W and f f, d AND W with f Clear f Clear W f, d Complement f f, d Decrement f f, d Decrement f f, d Increment f f, d Increment f, Skip if 0 f, d Increment f, Skip if 0 f, d Inclusive OR W with f f, d Move f f Move W to f No Operation f, d Rotate Left f through Carry f, d Rotate Right f through Carry f, d Swap nibbles in f f, d Exclusive OR W with f ED FILE REGISTER OPERATIONS f, b Bit Clear f f, b Bit Test f, Skip if Clear f, b Bit Test f, Skip if Set ID CONTROL OPERATIONS k Add literal and W k AND literal with W k Call subroutine Clear Watchdog Timer k Go to address k Inclusive OR literal with W k Move literal to W Return from interrupt k Return with literal in W Return from Subroutine Go into standby mode k Subtract W from literal	f, d	Tree File Register Operations	NTED FILE REGISTER OPERATIONS	NTED FILE REGISTER OPERATIONS	NTED FILE REGISTER OPERATIONS	NTED FILE REGISTER OPERATIONS

- Iote 1: When an I/O register is modified as a function of itself (e.g., MOVF_PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
 - If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module
 - If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

DIRECTIVAS:

DIRECTIVA	INDICACIÓN	EJEMPLO
ORG X	Le informa al programa ensamblador en que posición de memoria de programa debe ir ubicada la siguiente línea de código.	ORG 05H
Banksel f	Selecciona el banco de memoria RAM en que se encuentra f	Banksel TRISA
GLOBAL id	Asigna una variable de nombre id en una posición fija de memoria RAM.	GLOBAL unidad

COMENTARIOS

