



MICROINSTRUCCIONES.

La unidad de control es el “**cerebro**” del microprocesador. Esta unidad realiza la decodificación de los códigos de operación y de los códigos de función para poder identificar la instrucción que se va a ejecutar y su tipo (Tipo I, Tipo R y Tipo J). Una vez identificada la instrucción, la unidad de control activa o no cada una de las señales de control de cada unidad funcional (ALU, Archivo de registros, Memoria de datos, Memoria de programa, Pila) del microprocesador. De esta manera se forma un código que genera la unidad de control al que llamamos **microinstrucción**. Por lo tanto, podemos definir a **una microinstrucción como cada uno de los códigos que genera la unidad de control para activar las señales de control de cada unidad funcional del procesador. Estos códigos se generan en cada ciclo de reloj para poder ejecutar una instrucción del ensamblador.**

Para entender como se realiza la ejecución de las instrucciones a través del procesador y como se generan las microinstrucciones, analicemos los ejemplos siguientes.

Ejemplo 1. Programa que genera un contador. Observe el código mostrado en la tabla 1.

Instrucciones	UP	DW	WPC	SDMP	SR2	SWD	SHE	DIR	WR	LF	SEXT	SOP1	SOP2	ALUOP	SDMD	WD	SR
LI R0, #1	0	0	0	0	0	0	0	0	1	0	0	0	0	0000	0	0	0
LI R1, #7	0	0	0	0	0	0	0	0	1	0	0	0	0	0000	0	0	0
CICLO: ADD R1, R1, R0	0	0	0	0	0	1	0	0	1	1	0	0	0	0011	0	0	1
SWI R1, 5	0	0	0	0	1	0	0	0	0	0	0	0	0	0000	1	1	0
B CICLO	0	0	1	0	0	0	0	0	0	0	0	0	0	0000	0	0	0

Tabla 1 Microinstrucciones del programa ejemplo 1.

Durante la ejecución de cada instrucción TODAS las señales de control de cada bloque del microprocesador se encuentran por defecto con cero. Lo primero que se realiza es tomar una instrucción de la memoria de programa, eso quiere decir, que se lee la localidad 0 de la memoria y se saca el primer dato de 25 bits guardado ahí. Después la unidad de control decodifica la instrucción para saber que tipo de instrucción acaba de leer de la memoria de programa. Posteriormente la unidad de control activa las correspondientes señales de control del procesador para ejecutar la instrucción.

La ejecución del programa comienza después de un reset o clear. Esto provoca que el Stack Pointer (SP) de la pila comience en cero al igual que todos contadores de programa y registros del procesador. Con el SP en cero el contador que maneja las direcciones de la memoria de programa es PC0. En este ejemplo en particular la primera instrucción a ejecutar es LI R0, #1. Esta instrucción sale de la memoria de programa y el código de operación colocado en los bits 24...20 se va directamente a la unidad de control. El código del registro R0 colocado en los bits 19...16 se coloca en el bus de entrada WRITE REGISTER del archivo de registros. El número 1

AUTOR: VICTOR HUGO GARCÍA ORTEGA



colocado en los bits 15...0 se coloca en el bus de entrada WRITE DATA del archivo de registros. La instrucción se ejecuta cuando la unidad de control activa la señal WR del archivo de registros, esto provoca que en el siguiente flanco de subida de la señal de reloj se cargue el número 1 en el registro R0 y que el valor del PC0 se incremente en uno. La siguiente instrucción LI R1, #7 se ejecuta de la misma manera.

Después de la ejecución de las instrucciones de carga, la siguiente instrucción a ejecutar se encuentra en la dirección 2 de memoria. Esta instrucción es ADD R1, R1, R0. Esta instrucción sale de la memoria de programa y el código de operación colocado en los bits 24...20 se va directamente a la unidad de control. Como se trata de una instrucción tipo R el código de función colocado en los bits 3...0 también se va a la unidad de control. El código del registro R1 colocado en los bits 19...16 se coloca en el bus de entrada WRITE REGISTER del archivo de registros. El código del registro R1 colocado en los bits 15...12 se coloca en el bus de entrada READ REGISTER 1 del archivo de registros. Esto provoca que el número 7 almacenado en R1 pase directamente a la ALU. El código del registro R0 colocado en los bits 11...8 se coloca en el bus de entrada READ REGISTER 2 del archivo de registros. Esto provoca que el número 1 almacenado en R0 pase directamente a la ALU. La instrucción se ejecuta cuando la unidad de control coloca el código "0011" en el bus ALUOP, activa la señal WR del archivo de registros, la señal SR, la señal SWD y la señal LF del registro de banderas **(La señal LF se activa con todas las instrucciones que modifican las banderas de la ALU)**. Esto provoca que se realice la suma de los operandos de la ALU y que el resultado de la suma se coloque en el bus de entrada WRITE REGISTER del archivo de registros. Entonces, el resultado de la suma se almacena en el registro R1 en el siguiente flanco de subida de la señal de reloj y que el valor del PC0 se incremente a tres.

Posteriormente, la siguiente instrucción a ejecutar es SWI R1, 5. Esta instrucción sale de la memoria de programa y el código de operación colocado en los bits 24...20 se va directamente a la unidad de control. El código del registro R1 colocado en los bits 19...16 se coloca en el bus de entrada READ REGISTER 2 del archivo de registros. Esto provoca que el número 8 almacenado en R1 se coloque en el bus Di15...Di0 de la memoria de datos. El número 5 colocado en los bits 15...0 se coloca en el bus de direcciones A15...A0 de la memoria de datos. La instrucción se ejecuta cuando la unidad de control activa la señal WD de la memoria de datos y la señal SDMD. Esto provoca que se escriba el número 8 en la dirección de memoria 5 en el siguiente flanco de subida de la señal de reloj y que el valor del PC0 se incremente a cuatro.

Finalmente, la última instrucción a ejecutar es B CICLO. Esta instrucción sale de la memoria de programa y el código de operación colocado en los bits 24...20 se va directamente a la unidad de control. El código colocado en los bits 19...16 no se utiliza en esta instrucción. El número 5 colocado en los bits 15...0 se coloca en el bus de entrada D15...D0 de la pila. La instrucción se ejecuta cuando la unidad de control activa la señal WPC de la pila. Esto provoca que se escriba la dirección 2 en PC0 y se de el salto a la etiqueta CICLO.



Ejemplo 2. Programa que obtiene el promedio de dos números. Observe el código mostrado en la tabla 2.

Instrucciones	UP	DW	WPC	SDMP	SR2	SWD	SHE	DIR	WR	LF	SEXT	SOP1	SOP2	ALUOP	SDMD	WD	SR
LI R0, #34	0	0	0	0	0	0	0	0	1	0	0	0	0	0000	0	0	0
LI R1, #30	0	0	0	0	0	0	0	0	1	0	0	0	0	0000	0	0	0
ADD R2, R0, R1	0	0	0	0	0	1	0	0	1	1	0	0	0	0011	0	0	1
SRL R2, R2, #1	0	0	0	0	0	0	1	0	1	0	0	0	0	0000	0	0	0
SWI R2, 64	0	0	0	0	1	0	0	0	0	0	0	0	0	0000	1	1	0
CICLO: NOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	0	0	0
B CICLO	0	0	1	0	0	0	0	0	0	0	0	0	0	0000	0	0	0

Tabla 2: Microinstrucciones del ejemplo 2

Ejemplo 3. Programa que obtiene los primeros 12 términos de la serie de Fibonacci. Observe el código mostrado en la tabla 3.

Instrucciones	UP	DW	WPC	SDMP	SR2	SWD	SHE	DIR	WR	LF	SEXT	SOP1	SOP2	ALUOP	SDMD	WD	SR
LI R0, #0	0	0	0	0	0	0	0	0	1	0	0	0	0	0000	0	0	0
LI R1, #1	0	0	0	0	0	0	0	0	1	0	0	0	0	0000	0	0	0
LI R2, #0	0	0	0	0	0	0	0	0	1	0	0	0	0	0000	0	0	0
LI R3, #10	0	0	0	0	0	0	0	0	1	0	0	0	0	0000	0	0	0
SERIE: ADD R4, R0, R1	0	0	0	0	0	1	0	0	1	1	0	0	0	0011	0	0	1
SWI R4, 72	0	0	0	0	1	0	0	0	0	0	0	0	0	0000	1	1	0
ADDI R0, R1, #0	0	0	0	0	0	1	0	0	1	1	0	0	1	0011	0	0	1
ADDI R1, R4, #0	0	0	0	0	0	1	0	0	1	1	0	0	1	0011	0	0	1
ADDI R2, R2, #1	0	0	0	0	0	1	0	0	1	1	0	0	1	0011	0	0	1
BNEI R2, R3, SERIE (semiciclo positivo)	0	0	0	0	1	0	0	0	0	1	0	0	0	0111	0	0	0
Si la condición se cumple (semiciclo negativo)	0	0	1	1	0	0	0	0	0	0	0	1	1	0011	0	0	1
CICLO: NOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	0	0	0
B CICLO	0	0	1	0	0	0	0	0	0	0	0	0	0	0000	0	0	0

Tabla 3 Microinstrucciones del programa ejemplo 3.



Instituto Politécnico Nacional
Escuela Superior de Cómputo



INSTRUCCIONES DE CARGA Y ALMACENAMIENTO									
Instr.	Ejemplo	Significado	Código de operación						Microinstrucción
LI	LI Rd, #Slit16	Rd = Slit16	01	Rd	Slit16				WR
LWI	LWI Rd, lit16	Rd = Mem[lit16]	02	Rd	lit16				
LW	LW Rd, lit12(Rt)	Rd = Mem[Rt+lit12]	23	Rd	Rt	lit12			
SWI	SWI Rd, lit16	Mem[lit16] = Rd	03	Rd	lit16				SR2 SDMD WD
SW	SW Rd, lit12(Rt)	Mem[Rt+lit12] = Rd	04	Rd	Rt	lit12			
INSTRUCCIONES ARITMÉTICAS									
ADD	ADD Rd,Rt,Rs	Rd = Rt+Rs	00	Rd	Rt	Rs	S/U	00	SWD WR LF SR ALUOP=0011
SUB	SUB Rd,Rt,Rs	Rd = Rt-Rs	00	Rd	Rt	Rs	S/U	01	SWD WR LF SR ALUOP=0111
ADDI	ADDI Rd,Rt,#Slit12	Rd = Rt+Slit12	05	Rd	Rt	Slit12			SWD WR LF SR ALUOP=0011 SOP2
SUBI	SUBI Rd,Rt,#Slit12	Rd = Rt-Slit12	06	Rd	Rt	Slit12			SWD WR LF SR ALUOP=0111 SOP2
INSTRUCCIONES LÓGICAS									
AND	AND Rd,Rt,R	Rd=Rt&Rs	00	Rd	Rt	Rs	S/U	02	
OR	OR Rd,Rt,Rs	Rd=Rt Rs	00	Rd	Rt	Rs	S/U	03	
XOR	XOR Rd,Rt,Rs	Rd=Rt ^ Rs	00	Rd	Rt	Rs	S/U	04	
NAND	NAND Rd,Rt,Rs	Rd=~(Rt & Rs)	00	Rd	Rt	Rs	S/U	05	
NOR	NOR Rd,Rt,Rs	Rd=~(Rt Rs)	00	Rd	Rt	Rs	S/U	06	
XNOR	NOR Rd,Rt,Rs	Rd=~(Rt ^ Rs)	00	Rd	Rt	Rs	S/U	07	
NOT	NOT Rd, Rs	Rd = ~Rs	00	Rd	Rs	Rs	S/U	08	
ANDI	ANDI Rd,Rt,#lit12	Rd=Rt & lit12	07	Rd	Rt	lit12			
ORI	ORI Rd,Rt,#lit12	Rd=Rt lit12	08	Rd	Rt	lit12			
XORI	XORI Rd,Rt,#lit12	Rd=Rt ^ lit12	09	Rd	Rt	lit12			
NANDI	NANDI Rd,Rt,#lit12	Rd=~(Rt & lit12)	10	Rd	Rt	lit12			

AUTOR: VICTOR HUGO GARCÍA ORTEGA



Instituto Politécnico Nacional
Escuela Superior de Cómputo



NORI	NORI Rd,Rt,#lit12	$Rd = \sim(Rt \mid \text{lit12})$	11	Rd	Rt	lit12	
XNORI	XNORI Rd,Rt,#lit12	$Rd = \sim(Rt \wedge \text{lit12})$	12	Rd	Rt	lit12	
INSTRUCCIONES DE CORRIMIENTO							
SLL	SLL Rd,Rt,#lit4	$Rd = Rt << \text{lit4}$	00	Rd	Rt	S/U	lit4 09 SHE DIR WR
SRL	SRL Rd,Rt,#lit4	$Rd = Rt >> \text{lit4}$	00	Rd	Rt	S/U	lit4 10 SHE WR
INSTRUCCIONES DE SALTOS CONDICIONALES E INCONDICIONALES							
BEQI	BEQI Rd,Rt,Slit12	If($Rd == Rt$) goto Slit12 $PC = PC + \text{Slit12}$	13	Rd	Rt	Slit12	SR2 LF ALUOP=0111 WPC SDMP SOP1 SOP2 SR ALUOP=0011 LF
BNEI	BNEI Rd,Rt,Slit12	If($Rd \neq Rt$) goto Slit12 $PC = PC + \text{Slit12}$	14	Rd	Rt	Slit12	SR2 LF ALUOP=0111 WPC SDMP SOP1 SOP2 SR ALUOP=0011 LF
BLTI	BLTI Rd,Rt,Slit12	If($Rd < Rt$) goto Slit12 $PC = PC + \text{Slit12}$	15	Rd	Rt	Slit12	SR2 LF ALUOP=0111 WPC SDMP SOP1 SOP2 SR ALUOP=0011 LF
BLETI	BLETI Rd,Rt,Slit12	If($Rd \leq Rt$) goto Slit12 $PC = PC + \text{Slit12}$	16	Rd	Rt	Slit12	SR2 LF ALUOP=0111 WPC SDMP SOP1 SOP2 SR ALUOP=0011 LF
BGTI	BGTI Rd,Rt,Slit12	If($Rd > Rt$) goto Slit12 $PC = PC + \text{Slit12}$	17	Rd	Rt	Slit12	SR2 LF ALUOP=0111 WPC SDMP SOP1 SOP2 SR ALUOP=0011 LF
BGETI	BGETI Rd,Rt,Slit12	If($Rd \geq Rt$) goto Slit12 $PC = PC + \text{Slit12}$	18	Rd	Rt	Slit12	SR2 LF ALUOP=0111 WPC SDMP SOP1 SOP2 SR ALUOP=0011 LF
B	B lit16	$PC = \text{lit16}$	19	S/U	lit16		WPC
INSTRUCCIONES DE MANEJO DE SUBROUTINAS							
CALL	CALL #lit16	$PC(n+1) = \text{lit16}$	20	S/U	lit16		
RET	RET	$PC = PC(n-1)$	21	S/U	S/U	S/U	S/U
OTRAS INSTRUCCIONES							
NOP	NOP		22	S/U	S/U	S/U	S/U