

1. Explique razonadamente (no basta copiar lo que está escrito en la tabla)

- Las diferencias entre las instrucciones ST y STS

La diferencia es que utilizan distinto modo de direccionamiento para el operando destino. En ambos casos, el destino es la memoria de datos. En el caso de ST indica la dirección a escribir mediante un registro: Y o Z, es decir, utiliza direccionamiento indirecto. En el caso de STS indica la dirección a escribir de forma explícita, es decir, usa direccionamiento directo a memoria (también llamado absoluto).

- Las diferencias entre las instrucciones RET y STOP

RET es retorno de subrutina, indica el final de una subrutina, es un salto incondicional, cuando se ejecuta el contador de programa ha de recuperar el valor almacenado en la pila y continuar la ejecución del programa que llamó a la subrutina. STOP indica el final del programa, esto quiere decir que ya no hay que ejecutar más instrucciones.

- Las diferencias entre las instrucciones SUB y CP

Ambas realizan una resta en la ALU y modifican el valor de SREG con arreglo al resultado de la resta. La diferencia entre ellas está en que SUB almacena el resultado de la resta en el registro destino mientras que en el caso de CP la resta no queda almacenada en ningún registro.

2. Proporcione la descomposición en microoperaciones de la instrucción CALL dir

	<u>Unidad de datos:</u>	<u>Unidad de control:</u>
Ciclo de búsqueda:	1. $IR \leftarrow \text{MEMCOD}(PC)$, $PC \leftarrow PC+1$,	WIR,IPC
Ciclo de ejecución:	1. $MAR \leftarrow SP$, $SP \leftarrow SP-1$,	WMAR,RSP,DSP
	2. $\text{MEMDAT}(MAR) \leftarrow PC$, $AC \leftarrow IR7:0$	RPC,WMEM,WAC,INM,TB
	3. $PC \leftarrow AC$	WPC,RAC

3. Considere el siguiente programa del CS3:

a) Obtenga su código máquina.

	<u>Posición</u>	<u>Contenido</u>
LDI R7, 0	0	11111 111 00000000
CPI R5, 0	1	11011 101 00000000
BREQ almac	2	00110 000 00001000
CPI R6, 0	3	11011 110 00000000
sigue: BRZS almac	4	00110 000 00001000
ADD R7, R5	5	01000 111 00000 101
SUBI R6, 1	6	11010 110 00000001
JMP sigue	7	00111 000 00000100
almac: STS 0xA2,R7	8	00010 111 10100010
STOP	9	10111 000000000000

b) Suponga que inicialmente $R5=\$10$, $R6=5$ y $R7=\$32$, diga qué valor tienen estos tres registros tras la ejecución completa del programa.

Si analizamos lo que hace el programa podemos ver que realiza la operación producto de los registros $R5$ y $R6$ y almacena el resultado en $R7$, ya que suma $R5$ a $R7$ tantas veces como indica $R6$. En cada iteración, además, decrementa $R6$.

Por tanto, tras la ejecución del programa, podemos asegurar que $R7=80$ ($\$50$ en hexadecimal), $R5=\$10$ (16 en decimal) y $R6=0$

c) Diga el valor de PC, IR, SP y MAR al finalizar el programa.

PC = 10 ya que la instrucción STOP está en la dirección 9, pero el PC se incrementa al hacer la búsqueda y ya está apuntando a la siguiente instrucción antes de que la unidad de control decodifique a STOP.

IR = $\$B800$ (en binario: 10111 0000000000) pues contiene la última instrucción ejecutada que es STOP

SP = $\$FF$ (valor inicial de SP) ya que no se ha usado la pila.

MAR = $\$A2$ pues la última vez que se accede a la memoria de datos se hace a esta dirección

d) Calcule cuántos ciclos de reloj se tarda en ejecutar el siguiente trozo de programa:

ADD R7, R5 → 3 ciclos (1 ciclo de búsqueda + 2 de ejecución)

SUBI R6, 1 → 3 ciclos (1 ciclo de búsqueda + 2 de ejecución)

JMP sigue → 3 ciclos (1 ciclo de búsqueda + 2 de ejecución)

total : 9 ciclos de reloj

4. Se desea añadir al CS3 una nueva instrucción, STSI, cuya sintaxis es la siguiente:

STSI dir,dato3.

Dicha instrucción almacena un dato sin signo de 3 bits en la dirección *dir* de MEMDAT.

a) Asigne a la nueva instrucción el código de operación y el formato de código máquina que considere oportunos. Como ejemplo, escriba el código máquina de la instrucción STSI $\$E2,3$.

b) Describa, si son necesarias, las modificaciones que se deben realizar en la unidad de datos para poderla implementar.

c) Obtenga la secuencia de microoperaciones de STSI en la nueva unidad de datos (transferencias RT y señales a activar).

d) Escriba un fragmento de programa que, haciendo uso de STSI, inicialice la memoria de datos con los valores 2, 7 y 0 a partir de la dirección $\$F0$.