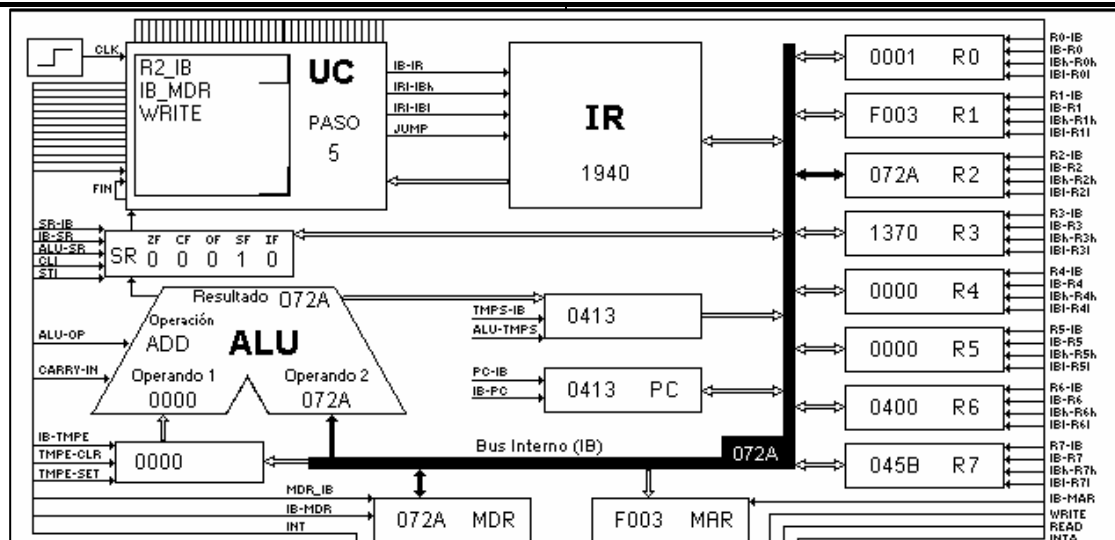


<div> <div>A</div> <div> Instrucciones generales para la realización de este examen La respuesta debe escribirse en el hueco existente a continuación de cada pregunta con letra clara. Cada respuesta correcta suma un punto. Cada respuesta incorrecta, ilegible o vacía no suma ni resta. El total de puntos se dividirá entre el total de preguntas y se multiplicará por 10 para obtener la nota del examen. El siguiente programa para la CPU teórica dibuja un asterisco en pantalla y lo mueve hacia a la izquierda cuando se pulsa la tecla ‘o’ y hacia la derecha cuando se pulsa la ‘p’. Para ello utiliza tres elementos: <ul style="list-style-type: none"> La variable pos: Indica la posición actual donde está el asterisco. El procedimiento mueve_asterisco: Recibe por la pila un valor que indica el incremento que hay sumarle a pos para volver a pintar el asterisco. Antes de pintarlo, lo borra de la posición anterior. La rutina de servicio rut_teclado: Comprueba qué teclas se han pulsado. Si se pulsa una ‘o’, llama a mueve_asterisco con parámetro -1; si se pulsa una ‘p’, lo llama con parámetro 1; si es otra tecla no hace nada. </div> </div> <div> <pre> ORIGEN 400h INICIO main .PILA XXXXh ; Eliminado intencionadamente .DATOS pos VALOR 0F000h .CODIGO PROCEDIMIENTO mueve_asterisco PUSH R6 MOV R6, R7 PUSH R0 PUSH R1 PUSH R2 ; Recuperar en R0 el parámetro ; -- 1 -- MOVL R6, BYTEBAJO DIRECCION pos MOVH R6, BYTEALTO DIRECCION pos MOV R1, [R6] ; R1 = pos actual ; Escribir en blanco en la pos actual </pre> </div>	<pre> MOVL R2, ' ' MOVH R2, 7 MOV [R1], R2 ; Modificar pos y escribir el asterisco ADD R1, R1, R0 MOVL R2, '*' MOVH R2, 7 MOV [R1], R2 MOV [R6], R1 ; Actualizar pos POP R2 POP R1 POP R0 POP R6 RET FINP PROCEDIMIENTO rut_teclado PUSH R0 PUSH R1 PUSH R2 PUSH R3 PUSH R6 MOVL R1, 1h ; R1 = reg. control tecl. MOVH R1, 0FEh MOVL R2, 0 ; Hacer R2 = máscara MOVH R2, 1 bucle: ; Hacerlo mientras queden teclas MOV R3, [R1] AND R3, R3, R2 BRZ no_hay_tecla DEC R1 ; R1 = reg. de datos MOV R3, [R1] ; R3 = tecla pulsada INC R1 ; R1 = reg. de control ; Comprobar la tecla pulsada MOVL R6, 'o' MOVH R6, 12h ; scan de la 'o' COMP R6, R3 ; --2-- Saltar si es necesario XOR R6, R6, R6 DEC R6 ; Poner un -1 en R6 PUSH R6 CALL mueve_asterisco INC R7 JMP bucle no_es_o: </pre>	<pre> MOVL R6, 'p' MOVH R6, 13h ; scan de la 'p' COMP R6, R3 BRNZ bucle MOVL R6, 1 MOVH R6, 0 PUSH R6 CALL mueve_asterisco INC R7 JMP bucle no_hay_tecla: POP R6 POP R3 POP R2 POP R1 POP R0 ; --3-- FINP main: XOR R0, R0, R0 MOVL R1, BYTEBAJO DIRECCION rut_teclado MOVH R1, BYTEALTO DIRECCION rut_teclado MOV [R0], R1 STI JMP -1 FIN </pre> <div> La figura de la página siguiente muestra un momento de la ejecución del programa. <ul style="list-style-type: none"> Después de haber generado una interrupción y de recibir de la CPU la señal INTA, ¿cuál es el primer número que la interfaz del teclado envía por el bus de datos? <div>0000h</div> <ul style="list-style-type: none"> ¿Cuál es el código ASCII del asterisco? Contestar en hexadecimal. <div>2Ah</div> <ul style="list-style-type: none"> ¿Qué instrucción o instrucciones falta/n en -- 2 --? <div>BRNZ no_es_o</div> </div>
--	--	---



- ¿Qué instrucción o instrucciones falta/n en -- **1** --?

INC R6
INC R6
MOV R0, [R6]

- Sabiendo que no se ha pulsado ninguna vez la tecla ‘o’, ¿cuántas veces como mínimo se habrá pulsado la tecla ‘p’ antes de llegar al instante mostrado en la figura?

3

- ¿Qué instrucción o instrucciones falta/n en -- **3** --?

IRET

- ¿Qué tamaño mínimo debe tener la pila? **Contestar en decimal.**

13

- Si en el paso 1 de la ejecución de la primera instrucción del programa se pulsase una tecla, ¿qué habría en IR en el primer paso de la primera instrucción que se ejecutase de **rut_teclado**? **Contestar en hexadecimal.**

A800h

- Se pretende implementar un sumador elemental mediante un PLA.

- ¿Cuál es el número **mínimo** de puertas AND y OR necesario para confeccionar el PLA? Nota: Las puertas pueden tener cuantas entradas necesiten.

AND: 7 OR: 2

Se desea diseñar un dispositivo de memoria que cubra la primera cuarta parte del espacio de direcciones de la CPU teórica. El dispositivo tiene un decodificador 2:4.

- ¿De cuántos bancos se compone el dispositivo?

4

- Sabiendo que el ancho de palabra de los chips es la mitad del ancho de palabra de la CPU, ¿cuál es la organización de los chips? (Ejemplo de formato de respuesta: 6Kx9).

4Kx8

Se dispone de una unidad de control microprogramada para la CPU teórica que se encarga de generar las palabras de control. Las palabras de control se interpretan como se indica en la figura (se muestran sólo los 14 bits menos significativos).

	LEER	JUMP	IB-MAR	IB-PC	IB-TMPE	IB-IR	MDR-IB	PC-IB	CarryIn	TMPS-IB	ADD	TMPE-CLR	ALU_TMPS	FIN
...														

IR = 1100 0000 1111 1001

- Para el valor del Registro de Instrucción que se muestra, ¿cuáles serían las palabras de control que generaría la unidad de control? Codificar las palabras de control en **HEXADECIMAL**.

4	0240
5	100A
6	0411
7	

En la CPU teórica se sabe que el siguiente fragmento de código ha tardado 3.2 microsegundos en ejecutarse y que la instrucción INC R6 sólo ha producido acarreo la segunda vez que se ejecutó:

```
atrás: INC R6
      BRC alli
      JMP atrás
alli:  XOR R6, R6, R6
```

— ¿Cuál es la frecuencia, en MHz, del reloj de la CPU?

10 MHz

Se ha creado una nueva instrucción para la CPU teórica que permite realizar un salto si el valor del registro R0 es cero. El mnemónico de la nueva instrucción es:

BR_R0Z Inm_8

Durante las señales de control de la instrucción, para saber si el registro R0 contiene el valor cero, se realiza una operación de suma en la ALU y se comprueba si el bit de cero está activo o no.

Sabiendo que la CPU tiene un reloj de 5 MHz y que la ejecución de esta instrucción dentro de un programa duró 1,2 microsegundos, completar la siguiente tabla con las señales de control que se generaron durante la ejecución de la instrucción (el paso 5 ya se suministra completo y se supone que la CPU ha sido modificada para realizarlo):

4	R0-IB; TMPE_CLR; ADD; ALU-SR
5	(comprobar si ZF=0)
6	FIN
7	
8	

Se sabe que las entradas de la ALU de 16 bits vista en clase tienen los siguientes valores:

- En el **operando A** se encuentra la combinación de bits que representa el uno en formato exceso a Z con 16 bits y exceso central.

- En el **operando B** se encuentran los 16 bits más altos de la codificación del número $2^{-127}(1+2^{-2})$ en formato IEEE-754.

- **CarryIn** = 0, **Resta** = 1, **Op1** = 1 y **Op0** = 1.

Se recuerda que las ALUs elementales que componen la ALU de 16 bits tienen conectadas las salidas de una puerta AND, de una puerta OR, de una puerta XOR y el resultado de un sumador elemental, a las entradas e_0 , e_1 , e_2 y e_3 del multiplexador, respectivamente.

– ¿Cuál es el resultado obtenido en la salida? Expresar el resultado con **cuatro dígitos hexadecimales**.

7F B1h

En la misma ALU del ejercicio anterior se introducen ahora los siguientes valores:

- En el **operando A** el número positivo más grande que se puede codificar en complemento a 2.

- En el **operando B** el número negativo más grande que se puede codificar en exceso a Z central.

- **CarryIn** = 0, **Resta** = 0, **Op1** = 1 y **Op0** = 1.

– ¿Cuál es el resultado obtenido en la salida y cuál será el valor de los bits de estado? Expresar el resultado en **DECIMAL, interpretado en complemento a 2**.

Resultado (decimal)= -2 ZF= 0 OF= 1 SF= 1 CF= 0

— ¿Cuáles de las siguientes afirmaciones son **CIERTAS**? (Puedes responder "ninguna" si así lo consideras.)

- El sistema UNICODE de representación de caracteres, al utilizar conjuntos distintos para distintas lenguas, permite la asignación del mismo código a distintos caracteres, siempre y cuando pertenezcan a conjuntos diferentes.
- Las CPUs con unidades de control microprogramadas reservan una parte de su espacio de direccionamiento en la memoria del sistema para almacenar la memoria de microprograma.
- Una CPU con unidad de control cableada ejecuta un programa a la misma velocidad que otra CPU con unidad de control microprogramada siempre y cuando ambas tengan la misma frecuencia de reloj y ejecuten los pasos de control en el mismo tiempo.
- La entrada/salida basada en muestreo periódico obliga a la CPU a derrochar tiempo en la consulta del estado de los periféricos.

C, D

— ¿Cuáles de las siguientes afirmaciones son **CIERTAS**? (Puedes responder "ninguna" si así lo consideras.)

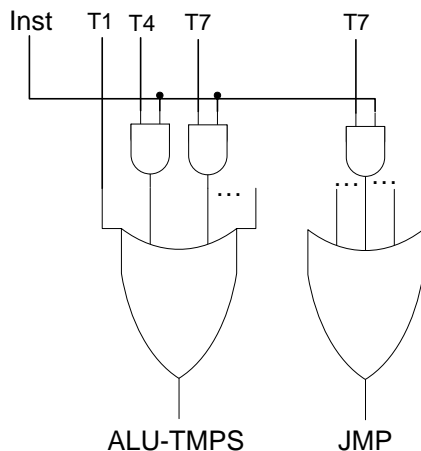
- El registro PC de la CPU de ejemplo se puede usar en las rutinas de interrupción para acceder al registro de estado almacenado en la pila.
- En la CPU de ejemplo, el usuario, a través de un programa, NO puede cargar en el registro PC el valor que desee.
- En la CPU teórica, se pueden usar los registros de propósito general R0, R1, ..., R6 para acceder, dentro de la pila, a los parámetros de un procedimiento.
- La relación capacidad de almacenamiento/velocidad de acceso es mayor en los chips de RAM estática que en los de RAM dinámica.

C

- Se ha diseñado un formato de representación de números reales en coma flotante con las siguientes características: 3 bits para el exponente que se representa en exceso a Z central, y 5 bits para la mantisa que se codifica en signo-magnitud y se normaliza a todo fracción con bit implícito a la derecha de la coma decimal. ¿Cuál sería en la representación en IEEE-754 del número más grande que se puede codificar en este formato diseñado? **Contestar en hexadecimal.**

40 F8 00 00

El siguiente circuito muestra la parte de la UC cableada encargada de activar las señales JUMP y ALU-TMPS para una determinada instrucción.



- ¿Cuál es la señal de control indicada como Inst?

CALL Inm_8

- Completa los caracteres que faltan en el interior de las casillas, teniendo en cuenta los caracteres que ya hay y sus códigos ASCII.

41h	61h	6Ch	6Ah	30h	34h
A			j		4

- Se ejecuta en la CPU teórica la instrucción SUB R0, R1, R2. Se sabe además que, justo antes de la ejecución de la instrucción, los bytes más significativos de R1 y R2 estaban a cero; para R1, el byte menos significativo contenía el resultado de un XOR entre los códigos ASCII de los caracteres 'v' y 'V'; y, para R2, contenía el resultado de un XOR entre los códigos ASCII de los caracteres 'W' y 'w'. ¿Cuál es el valor obtenido en R0 (**en hexadecimal**) y cuál es el valor de los bits del registro de estado tras ejecutar la instrucción?

Resultado: 0h Z: 1 C: 0 O: 0 S: 0

- Se desea ampliar la ALU para que, además de las 4 operaciones habituales (las lógicas AND, OR y XOR y la aritmética de suma -o resta, si está activada dicha línea-), permita efectuar otras 39 operaciones más. ¿Cuántas líneas de selección de operación debería **como mínimo** tener esta ALU? Contestar en **decimal**.

6

Se han utilizado 4 chips de memoria de 2Kx16 para diseñar un dispositivo de memoria. Este dispositivo se ha conectado a una CPU con un bus de direcciones de 14 líneas. Se ha visto que el dispositivo cubre el rango de direcciones 0000h-0FFF.

- Dibujar la puerta AND que debe actuar como circuito de activación del dispositivo. (En el ejemplo de respuesta se muestran tres líneas de dirección, pero puede ser cualquier otro número; indicar su peso y si entran negadas o no a la puerta.)

