

## Ejercicio Examen 1.A - Fundamentos del Software - Noviembre 2013

<b>Apellidos y Nombre:</b>	
<b>e-mail:</b>	

1. Marque **V** (verdadero) o **F** (falso) en las siguientes afirmaciones: **[3 ptos]**

- ☐ En una arquitectura Von Neumann, la memoria se divide en una zona específica de datos y una zona específica de código. De tal forma que ambas zonas son completamente independientes. **F**
- ☐ El registro de instrucción se encarga de ejecutar una instrucción de memoria. **F**
- ☐ El DMA hace uso de interrupciones para su buen funcionamiento. **V**
- ☐ El puntero de pila (SP) almacena una dirección de memoria y tiene que tener el mismo tamaño que el contador de programa (PC). **V**
- ☐ CALL es una instrucción que permite llamar a subrutinas, y para ello almacena cuando se ejecuta la dirección de memoria actual en un registro especial. **F**
- ☐ El valor 100010011010011 expresado en binario es igual al valor 0x44D3 expresado en hexadecimal. **V**

2. Tenemos un computador con arquitectura Von Neumann, cuyo repertorio de instrucciones contiene 128 instrucciones distintas. Su memoria contiene 1800 palabras, de las cuales, las 50 últimas están reservadas para pila. Además de los registros específicos (PC = contador de programa, IR = registro de instrucción, SP = puntero de pila, PSW = palabra de estado), dicho computador posee 2 registros generales (R0 y R1). La palabra de estado posee 4 bits tal y como muestra la siguiente tabla:

<b>RESERVADO PARA USO FUTURO</b> (NO USADO)	<b>DMATRNS</b> activado en caso de que el DMA termine una transferencia	<b>ZERO</b> <b>modificado solamente por</b> <b>operaciones aritméticas:</b> activado en caso de que la última operación fuera 0	<b>NEG</b> <b>modificado solamente por</b> <b>operaciones aritméticas:</b> activado en caso de que la última operación fuera < 0
--	---	---	--

Parte del repertorio de instrucciones se muestra en esta tabla, tenga en cuenta que en esta tabla se encuentran la instrucción más corta y la más larga en número de bits.

CÓDIGO INSTRUCCIÓN	INSTRUCCIÓN	DESCRIPCIÓN	CÓDIGO INSTRUCCIÓN	INSTRUCCIÓN	DESCRIPCIÓN
3	MOV RA, D	Mueve el contenido de la dirección D en RA.	7	PUSH RA	Introduce el contenido de RA en la pila.
4	MOV D, RA	Mueve el contenido de la dirección de memoria D al registro RA.	9	CALL D	Realiza una llamada a la subrutina en la dirección D.
10	POP RA	Devuelve el último elemento de la pila y lo introduce en RA.	15	SUB RA, RB	Operación aritmética: RB - RA e introduce el resultado en RA.

CÓDIGO INSTRUCCIÓN	INSTRUCCIÓN	DESCRIPCIÓN	CÓDIGO INSTRUCCIÓN	INSTRUCCIÓN	DESCRIPCIÓN
29	JNZ D	Salta a la dirección D si el bit ZERO es 0, en otro caso no hace nada	31	RET	Vuelve de una subrutina

1. Indique cuántos bits (como mínimo) ocuparía una palabra en memoria y el registro de instrucción. **[2 ptos]**

Como mínimo:

SP = PC =>  $\log 1800 / \log 2 \Rightarrow 11$  bits

PSW = 4 bits

Código de operación (COP) =>  $\log 128 / \log 2 = 7$  bits

Identificador numérico de registro (IDR) =>  $\log 2 / \log 2 = 1$  bit

dirección = PC = 11 bits

Instrucción = COP + dirección + IDR = 7 + 11 + 1 = 19 bits

IR = WORD = 19 bits

2. Simule 3 ciclos de instrucción de este programa teniendo en cuenta que PC=1000, los registros generales están a 0, PSW = 0x0 y SP = 1006. Tenga en cuenta que la dirección 0 contiene el valor 6 en decimal y la dirección 1 contiene el valor 4. Indique el valor de PSW en cada momento en hexadecimal, el resto de registros puede mostrarlos como más convenga. **[2.5 ptos]**

<b>0999</b> JNZ 1020	<b>1001</b> MOV 1, R2	<b>1003</b> POP R2	<b>1005</b> POP R1
<b>1000</b> MOV 0, R1	<b>1002</b> SUB R2, R1	<b>1004</b> JNZ 1000	<b>1006</b> JNZ 999

R1	R2	PC	RI	SP	PSW
6	0	1001	MOV 0,R1	1006	0x0
6	4	1002	MOV 1, R2	1006	0x0
6	2	1003	SUB R2, R1	1006	0x0
?	2	1004	POP R1	1005	0x0

3. Cree una rutina en la dirección 1400 que tome como parámetros dos valores de la pila y devuelva el contenido de la dirección de memoria 1401. El resultado será devuelto por la pila. Dicha rutina no debe alterar el valor de ninguno de los registros al ejecutarse. **[2.5 ptos]**

1400 MOV R1, 1209 # SALVAMOS LOS REGISTROS PARA LA OPERACIÓN

1401 MOV R2, 1210

1402 POP R1 # OBTENEMOS LOS PARÁMETROS

1403 POP R2

1404 MOV 1041, R2 # ALMACENAMOS LA ZONA DE MEMORIA DADA EN R2

1405 PUSH R2 # ALMACENAMOS RESULTADO EN LA PILA

```
1406 MOV 1210, R1 # RECUPERAMOS REGISTROS
1407 MOV 1211, R2
1408 JMP 1211 # RESERVAMOS ESPACIO PARA DATOS (DA IGUAL LO QUE INTRODUCAMOS)
1409 ---- # RESERVADO DATOS
1410 ---- # RESERVADO DATOS
1411 RET
```