

Ejercicio Examen 1.B - Fundamentos del Software - Noviembre 2013

Apellidos y Nombre:	
e-mail:	

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

- ☐ La ALU lee una instrucción desde un registro general y la ejecuta, modificando algunas veces la palabra de estado (PSW). **F**
- ☐ En una arquitectura de Harvard 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. **V**
- ☐ El uso de interrupciones vectoriales ha quedado obsoleto en los computadores modernos debido a la incorporación del DMA. **F**
- ☐ El puntero de pila (SP) y el registro de instrucción (IR) son siempre registros del mismo tamaño. **F**
- ☐ El valor 111110011010011 expresado en binario es igual al valor 0xF9A3 expresado en hexadecimal. **F**
- ☐ IRET es equivalente a JMP, con la diferencia de que saca de una pila específica la dirección a la que saltar. **V**

2. Tenemos un computador con arquitectura Von Neumann, cuyo repertorio de instrucciones contiene 128 instrucciones distintas. Su memoria contiene 2800 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:

RESERVADO PARA USO FUTURO (NO USADO)	DMATRNS activado en caso de que el DMA termine una transferencia	ZERO modificado solamente por operaciones aritméticas: activado en caso de que la última operación fuera 0	NEG modificado solamente por operaciones aritméticas: 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 2800 / log 2 => 12 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 = 12 bits

Instrucción = COP + dirección + IDR = 7 + 12 + 1 = 20 bits

IR = WORD = 20 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 = 1780. Tenga en cuenta que la dirección 0 contiene el valor 2 en decimal y la dirección 1 contiene también el valor 2. Indique el valor de PSW en cada momento en hexadecimal, el resto de registros puede mostrarlos como más convenga. **[2.5 ptos]**

0999 JNZ 1020	1001 MOV 1, R2	1003 PUSH R2	1005 POP R1
1000 MOV 0, R1	1002 SUB R2, R1	1004 JNZ 1000	1006 POP R1

R1	R2	PC	RI	SP	PSW
2	0	1001	MOV 0,R1	1780	0x0
2	2	1002	MOV 1, R2	1780	0x0
2	0	1003	SUB R2, R1	1780	0x2
2	0	1004	PUSH R2	1781	0x2

3. Cree una rutina en la dirección 1200 que tome como parámetros dos valores de la pila y los reste. El resultado será devuelto por la pila. Dicha rutina no debe alterar el valor de ninguno de los registros al ejecutarse. **[2.5 ptos]**

1200 MOV R1, 1209 # SALVAMOS LOS REGISTROS PARA LA OPERACIÓN

1201 MOV R2, 1210

1202 POP R1 # OBTENEMOS LOS PARÁMETROS

1203 POP R2

1204 SUB R2, R1 # RESTAMOS NÚMEROS Y RESULTADO EN R1

1205 PUSH R2 # ALMACENAMOS RESULTADO EN LA PILA

```
1206 MOV 1210, R1 # RECUPERAMOS REGISTROS
1207 MOV 1211, R2
1208 JMP 1211 # RESERVAMOS ESPACIO PARA DATOS (DA IGUAL LO QUE INTRODUCAMOS)
1209 ---- # RESERVADO DATOS
1210 ---- # RESERVADO DATOS
1211 RET
```