

Examen Final 08-06-2013. Teoría

Apellidos y nombre:

- Indica y explica brevemente los tipos de arquitecturas de los microprocesadores desde el punto de vista del juego de instrucciones **(1 pto.)**
- Indicar el contenido de los registros y la memoria que modifiquen su valor tras ejecutarse el siguiente código en lenguaje ensamblador **(1 pto.)**¹:

```

org $400
move.l -(a0), 4(a1)
move.l (a1)+, d4
move.w 1(a0, d0.w), d3
move.l d0, -(a7)
jsr subrutina
nop
fin      jsr fin

subrutina rts

org 0
dc.l $a000, $400
org $3000
ds.w 2
dc.b $1d, $aa
dc.w $1
dc.l $ad45e18
end

```

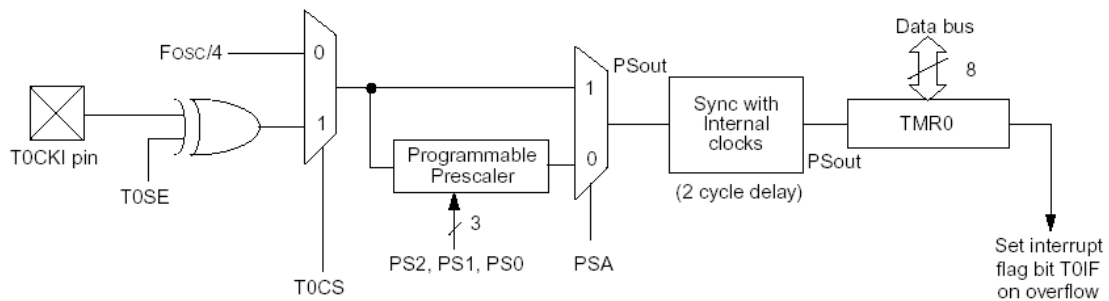
CPU

D0	FFFFFFFF		A0	0000300A	
D1	XXXXXXXX		A1	00003008	
D2	XXXXXXXX		A2	00000000	
D3	XXXXXXXX		A3	00000000	
D4	XXXXXXXX		A4	00000000	
D5	XXXXXXXX		A5	00000000	
D6	XXXXXXXX		A6	00000000	
D7	XXXXXXXX		A7	00000000	
V. inicial		V. final	V. inicial		V. final

Memoria	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00002FF0																
00003000																
00003010																
00001020																

¹ El ejercicio se responderá en la hoja del enunciado.

3. Con respecto al microprocesador MC68000, indica la funcionalidad y sentido (entrada o salida) de los siguientes pines del dispositivo: #DTACK, #BG, #BERR y #UDS-#LDS **(1 pto.)**
4. Describe el mecanismo de interrupciones del microprocesador MC68000 **(1 pto.)**
5. Describe detalladamente cómo es el mecanismo de ejecución segmentada en los microcontroladores PIC de la gama media. **(1,25 ptos.)**
6. Gestión de los puertos de E/S en lenguaje C mediante directivas **(0.75 ptos.)**
7. Explica el funcionamiento del siguiente diagrama de bloques, correspondiente a un bloque funcional de una MCU PIC de la gama media **(1 pto.)**:



Note 1: T0CS, T0SE, PSA, PS2:PS0 (OPTION_REG<5:0>).

Note 2: The prescaler is shared with Watchdog Timer (refer to [Figure 11-6](#) for detailed block diagram).

Examen Final 28-06-2013. Prácticas

Apellidos y nombre:

1. Escribir un programa en el lenguaje ensamblador del microprocesador MC68000, que calcule el doble de un número entero de 16 bits mediante una subrutina. Cualquier comunicación entre el programa principal y la subrutina ha de realizarse a través de la pila **(1 ptos.)**
2. Desarrollar una aplicación en lenguaje ensamblador para la MCU 16F877A, utilizando interrupciones, que proporcione sobre el display del circuito el número de veces que se ha pulsado la tecla RB0. Cuando se sobrepase el 9 el circuito debe representar esta situación de error mediante el carácter 'E' de forma indefinida **(1 pto.)**
3. Desarrollar una aplicación en lenguaje C para la MCU 16F877A, utilizando interrupciones, que proporcione sobre el display del circuito el número de veces que se ha pulsado la tecla RB0. Cuando se sobrepase el 9 el circuito debe representar esta situación de error mediante el carácter 'E' de forma indefinida y escribir sobre el LCD el mensaje 'Error' **(1 pto.)**