

Nombre (en mayúsculas):

Apellidos (en mayúsculas):

PROBLEMA 1 (4 puntos)

Supongamos que se quiere implementar una arquitectura igual a la del CS2010 salvo que poseerá una instrucción nueva llamada “leeSP”. Su sintaxis será la siguiente:

```
leeSP <registro_destino>
```

donde `<registro_destino>` denota un registro de propósito general. Desde el punto de vista del programador, su efecto es el siguiente:

```
<registro_destino> ← SP
```

A modo de ejemplo, supongamos que el contenido inicial de los registros R2 y SP es, respectivamente, 13 y 9. En esas condiciones, el contenido de R2 será 9 tras ejecutarse el siguiente programa:

leeSP R2

Para la implementación se usará la misma unidad de datos que se propuso para el CS2010. Se pide lo siguiente:

- a) (1 punto) Indique que bits del código de operación deberán usarse para codificar el número de registro destino en esta nueva instrucción.
- b) (3 puntos) Indique la secuencia de microoperaciones necesaria para realizar el ciclo de ejecución de la nueva instrucción así como las señales que debe activar la unidad de control rellenando la siguiente tabla. Procure minimizar el número de ciclos de reloj.

[illegible]

Problema 2 (3 puntos)

Escriba una subrutina en ensamblador de AVR que calcule el valor absoluto de un número. Su entrada se codifica en notación complemento a 2 de 16 bits en los registros R1:R0. Su salida se codifica usando la misma notación en los registros R21:R20. Procure minimizar la longitud del programa.

Problema 3 (2 puntos)

Dado el siguiente programa del CS2010, indique en binario o hexadecimal el código máquina de sus instrucciones.

```
                .equ asdf=1
                ADDI R2,1
    pepito:      ADDI R2,$01
                SUBI R1,asdf
                JMP  pepito
```

Problema 4 (1 punto)

Indique el contenido que tendrá el registro R0 de un procesador AVR justo tras ejecutar la siguiente secuencia de instrucciones:

```
CLR R0
CLR R1
INC R1
INC R1
STS 0,R1
INC R1
```