

ALUMNO: _____

Ejercicio 1.

a) Para cada una de las instrucciones siguientes, escriba una instrucción cuyo efecto sea equivalente.

STS 0x0A,R0 → MOV R10,R0 (ya que en la dirección 0x0A está mapeado el registro R10)

STS 0x2A,R20 → OUT 10,R20 (ya que la dirección 0x2A está en la zona de registros I/O)

LDS R29,0x29 → IN R29,9 (ya que la dirección 0x29 está en la zona de registros I/O)

IN R3,PINB → LDS R3,0x23 ya que PINB es un registro I/O que ocupa la dirección 0x23

b) Represente el mapa de memoria de datos del ATmega328P mediante un dibujo e indique el número total y la posición dentro de la memoria de datos, de los registros de propósito general, los registros de E/S y los registros de E/S extendidos. Indique también en dicho dibujo el tamaño de la SRAM y su ubicación dentro de la memoria de datos del ATmega328P.



c) Escriba una instrucción que guarde en R0 el contenido del registro de E/S extendido que está ubicado en la posición más baja de todas (la menor).

LDS R0,0x60

d) Configure el puerto B del microcontrolador como salida e indique las distintas formas en que podemos sacar el dato 0x45 por el puerto.

SER R16 / LDI R16,\$FF

OUT DDRB,R16

LDI R16,0x45

OUT PORTB,R16 / OUT 0x05,R16 / STS 0x25,R16

e) Considere la siguiente situación inicial, donde se indica en hexadecimal el contenido de ciertos registros y direcciones de memoria

Dirección	Contenido
\$300	\$B0
\$301	\$EF
\$302	\$01
\$303	\$00

Registro	Contenido
R0	\$A0
R1	\$F0
YH	\$03
YL	\$01

Indique el valor que tendrá el registro R0 después de ejecutar la instrucción indicada en cada uno de estos casos (independientemente):

MOV R0,R1 → R0 = \$F0
 MOVW R1:R0,Y → R0 = \$01
 LD R0,Y+ → R0 = \$EF
 LD R0,-Y → R0 = \$B0
 LDD R0,Y+2 → R0 = \$00
 PUSH R0 → R0 = \$A0

f) Suponga que un Atmega328P está ejecutando el siguiente fragmento de programa:

```
SBI DDRB,1
BUCLE: SBI PORTB,1
        CBI PORTB,1
        RJMP BUCLE
```

¿Cuál es el efecto de dichas líneas de programa?

Genera una señal cuadrada a través del pin PB1. En primer lugar, configura dicho pin como salida y, posteriormente, en un bucle va fijando su valor a 1 y luego a 0 en cada iteración.

Si tenemos en cuenta la duración de cada instrucción del bucle: sbi (1 ciclo), cbi (1 ciclo) y rjmp (2 ciclos), podemos afirmar también que el tiempo que la señal cuadrada está en 0 es mayor que el tiempo que está en 1. Para igualar ambos tiempos (esto no se pide en el examen) haríamos lo siguiente:

```
sbi ddrb,1
bucle: sbi portb,1
        nop
        nop
        cbi portb,1
        rjmp bucle
```

Ejercicio 2.

En un sistema basado en el Atmega328P se ha conectado un pulsador que no dispone de resistencia de pull-up al pin PC0, un led rojo al pin PC1 y 8 leds verdes a los pines del puerto B. (El pulsador tiene un extremo conectado a tierra y el otro al pin del microcontrolador, los leds están conectados con el cátodo a tierra.)

Se quiere diseñar un programa que recorra dos tablas A y B que contienen, cada una, 9 datos de 8 bits con signo, y genere una tabla C del mismo tamaño, de modo que los elementos de la Tabla C cumplan:

$$\text{TablaC}[i] \leftarrow \text{menor}(\text{TablaA}[i], \text{TablaB}[i])$$

En cada iteración se tomará un dato de la tabla A y otro de la tabla B. El acceso a cada pareja de datos se controlará por el pulsador: cuando se realice una pulsación se accederá a un dato de cada tabla, se compararán entre sí, y se escribirá el menor de los dos en la tabla C, además, dicho dato se mostrará en binario mediante la iluminación de los leds verdes. Con la siguiente pulsación se accederá a la siguiente pareja de datos y así sucesivamente. Al finalizar la operación se iluminará el led rojo y se apagarán todos los leds verdes.

Diseñe un programa que además de realizar la operación que se acaba de describir cumpla con los siguientes requisitos:

- Debe configurar los puertos de E/S adecuadamente
- Debe reservar espacio en la memoria de datos, a partir de la posición \$200, para las tablas (A, B y C). Las tablas estarán una justo a continuación de la otra sin espacio libre entre ellas. Se usarán las etiquetas TablaA, TablaB y TablaC para designar la dirección del primer elemento de cada una de las tablas.
- Debe llamar a la subrutina *inicializar*, que se encargará de cargar con valores las tablas A y B. Esta subrutina no tiene que diseñarla debe limitarse a llamarla una vez al principio del programa.

```
.include "m328pdef.inc"
```

```
.dseg
```

```
.org $200
```

```
TablaA: .byte 9
```

```
TablaB: .byte 9
```

```
TablaC: .byte 9
```

```
.cseg
```

```
.org 0x50
```

```
.def cont=r20 ; lo usaré para contar los datos
```

```
.def apaga=r17 ; lo usaré para apagar los diodos verdes al final
```

```
cbi ddrC,0 ; configuro PC0 como entrada
```

```
sbi portC,0 ; activo pull-up en PC0
```

```
sbi ddrC,1 ; configuro PC1 como salida para el led rojo
```

```

ldi r16,$ff
out ddrb,r16 ; configuro puerto B completo como salida para los leds verdes

call inicializar ; subrutina para inicializar tabla, no hay que escribirla

ldi xl,low(TablaA) ; inicialización de los tres punteros, uno a cada tabla
ldi xh,high(TablaA)
ldi yl,low(TablaB)
ldi yh,high(TablaB)
ldi zl,low(TablaC)
ldi zh,high(TablaC)

ldi cont,9
ldi apaga,0

out portb,apaga ; comienzo apagando los diodos verdes y el rojo
cbi portc,1

wait: sbic pinc,0
      rjmp wait ; espero pulsación

libera: sbis pinc,0
        rjmp libera ; espero fin de pulsación

      ld r0,x+
      ld r1,y+ ; tomo un elto de cada tabla y los comparo
      cp r0,r1
      brlt r0menor ; si r0 es el menor tendré que almacenarlo en TablaC
      st z+,r1 ; si llego aquí r1 es el menor y lo almaceno en TablaC
      out portb,r1 ; lo muestro en los diodos
next: subi cont,1 ; decremento el contador
      brne wait ; si aún no ha llegado a 0 continuo
      out portb,apaga ; si llego aquí, he terminado, apago los diodos verdes y
                        ; enciendo el rojo

      sbi portc,1

fin: rjmp fin

r0menor: st z+,r0
         out portb,r0 ; lo muestro en los diodos
         rjmp next

```