

ALUMNO: _____

Ejercicio 1

Para el ATMEGA328P,

(a) Escriba las instrucciones para:

- mover un dato desde R3 a R20

MOV R20,R3 o LDS R20,0x3 o STS 0x20,R3

- mover un dato desde el puerto PB al registro R20

IN R20,PINB o IN R20,3 o LDS R20,0x23

- mover un dato desde el registro R3 al puerto PD

OUT PORTD,R3 o OUT 0x0b,R3 o STS 0x2b,R3

- mover un dato desde el puerto PB al puerto PD

Usaremos un registro auxiliar

IN R20,PINB

OUT PORTD,R20

O cualquier combinación con las opciones de movimiento de los dos apartados anteriores.

- mover un dato desde la dirección 0 de la memoria SRAM a R20

LDS R20,0x100

- mover un dato desde R20 a la dirección de la memoria SRAM indicada por Z

ST Z,R20

- mover un dato desde la dirección de la memoria SRAM indicada por X a la dirección de la memoria SRAM indicada por Z

Usaremos un registro auxiliar

LD R20,X

ST Z,R20

(b) Suponga que R0 contiene el dato \$A7, y los registros XL y XH contienen los datos \$03 y \$07 respectivamente. Explique cómo cambian los contenidos de R0 y X tras ejecutar la instrucción "LDD R0,X+1". ¿Y tras ejecutar "LD R0,X+"?

Una respuesta válida para LDD R0,X+1, es decir que no se puede hacer ya que LDD solo admite los punteros Y y Z.

No obstante, la intención no era esa por tanto también considero válido que no se haya observado esa restricción y responder lo siguiente:

En el primer caso, se hace $R0 \leftarrow \text{MEMDAT}(\$0704)$ y el puntero X no cambia su valor.

En el segundo caso, se hace $R0 \leftarrow \text{MEMDAT}(\$0703)$ y $X \leftarrow \$0704$

(c) Si el ATMEGA328P tiene un reloj con frecuencia 8Mhz,

- Determine como configurar el temporizador para que se produzca una interrupción cada 0,1 ms.

Dado que en 1s transcurren 8.000.000 de ciclos, en 0,1 ms transcurrirán 800 ciclos, por tanto, podemos considerar modo CTC con VMAX = 800 y ck/1

Habrà que hacer OCR1A <- 800, TCCR1B <- 0b1001, TIMSK1 <- 0b10

- Escriba las líneas de código necesarias para configurar el temporizador como se ha pedido en el subapartado anterior así como para habilitar las interrupciones por alcanzarse el valor máximo de cuenta.

```
LDI R16,high(800)
STS OCR1AH,R16
LDI R16,low(800)
STS OCR1AL,R16
LDI R16, 0b1001
STS TCCR1B,R16
LDI R16,0b10
STS TIMSK1,R16
SEI
```

(d) Escriba una subrutina que inicialice los puertos de entrada/salida para poder conectar dos pulsadores mecánicos en PC1 y en PC0 y también 8 conmutadores en los pines PB7-0. Todos ellos precisan que se añada la correspondiente resistencia de pull-up. La subrutina también debe configurar los pines PD7-0 para conectar 8 diodos luminosos.

```
Init_ES:  CBI DDRC,0    //entrada PC0
          CBI DDRC,1    //entrada PC1
          SBI PORTC,0   //pull up PC0
          SBI PORTC,1   //pull up PC1
          CLR R16
          OUT DDRB,R16  //entradas PB7-0
          LDI R16,0xFF
          OUT PORTB,R16 //pull up PB7-0
          OUT DDRD,R16  //salidas PD7-0
          RET
```

Ejercicio 2

2.1 Escriba una subrutina (denominada *abs*) que calcule el valor absoluto de un dato de 8 bits que representa un número con signo en complemento a 2. La subrutina recibirá el dato en el registro R0 y escribirá el resultado en el registro R1, es decir, su operación será $R1 \leftarrow |R0|$

```
abs: CPI R0,0
     BRLT cambia_signo
     MOV R1,R0
     RET
cambiasigno: NEG R0
            MOV R1,R0
            RET
```

Otra solución

```
abs: SBRC R0,7
     NEG R0
     MOV R1,R0
     RET
```

2.2 Haciendo uso de la subrutina *abs*, escriba un programa que recorra una tabla de 20 bytes situada a partir de la dirección 0x200 de la memoria de datos y obtenga otra tabla con los valores absolutos de dichos números a partir de la dirección 0x230. El programa debe situarse a partir de la dirección 0x60 de la memoria de código.

```
.CSEG
.ORG 0X60
.DEF R16=CONT
LDI XH,high(0x200)
LDI XL,low(0x200)
LDI YH,high(0x230)
LDI YL,low(0x230)
LDI CONT,20
sigue: LD R0,X
      CALL abs
      ST Y,R1
      SUBI CONT,1
      BRNE sigue
fin:   RJMP fin
```