Estructura de Computadores (EdC-ISW-G1) 2018-19
Boletín 4: Programación AVR

Problema 1
Sean A y B dos números sin signo de un byte, almacenados en las direcciones $0100 y $0101 respectivamente. Escriba un fragmento de programa que obtenga la suma de ambos números y almacene el resultado (16 bits) en las dos siguientes posiciones de memoria.

Problema 2
Sean A y B dos números sin signo de un byte, almacenados en las direcciones $0100 y $0101 respectivamente. Escriba un fragmento de programa que obtenga la multiplicación de ambos números y almacene el resultado (16 bits) en las dos siguientes posiciones de memoria.

Problema 3
Escriba un fragmento de programa que sume tres números sin signo de 1 byte, almacenados consecutivamente a partir de la posición $0100 de la SRAM, y guarde el resultado (16bits) a partir de la dirección $0200.

Problema 4
Escriba un fragmento de programa que **sume tres números con signo** de 1 byte, almacenados consecutivamente a partir de la posición $0100 de la SRAM, y guarde el resultado (16bits) a partir de la dirección $0200.

Problema 5
A partir de la dirección $0100 de la SRAM se encuentra almacenado un mensaje de comunicaciones que se ha recibido por el puerto serie. La longitud del mensaje viene indicado en el segundo byte de dicho mensaje (máx. 255 bytes). El último byte del mensaje es una suma de comprobación (CHECKSUM) del mensaje completo. Programe una subrutina que devuelva en R0=$00 si el mensaje recibido es correcto, o R0=$FF si se ha detectado un error (suma de comprobación incorrecta).

Problema 6
Escriba un programa que traslade una tabla de 16 bytes almacenada en la dirección $0100 a la posición $0200.

Problema 7
a) Escriba una subrutina que traslade una tabla de datos de una posición a otra de memoria. El número de datos a trasladar se indica en R16, y los registros X e Y indican las direcciones fuente y destino respectivamente.
b) Resuelva el problema 6 utilizando la subrutina del apartado a.

Problema 8
Escriba un fragmento de programa cargue en R0 el elemento menor de una tabla de 16 números sin signo de 1 byte almacenados a partir de la dirección $0100.

Problema 9
Escriba una subrutina que devuelva en R0 el elemento menor de una tabla de números con signo de 1 byte. La dirección de comienzo de la tabla se indica en el registro X, y el número de datos en R16. Utilice esta subrutina para resolver el problema 8.

Problema 10
Escriba un programa que escriba a partir de la dirección $0100, todos los números pares existentes entre el 0 y el 255.

Problema 11
Una tabla con 100 datos de 1 byte con signo está almacenada a partir de la dirección $0100. Escriba un programa que almacene en R0 el número de datos positivos que hay en dicha tabla, y en R1 el número de datos negativos.

Problema 12
Una tabla con 16 datos de 1 byte está almacenada a partir de la dirección $0100. Escriba un programa que invierta el orden de la misma.

Problema 13

Escriba una subrutina que invierta el orden de datos de una tabla. El número de datos (de 1 byte) se indica en R16, y la dirección de comienzo de la tabla en X.

Problema 14

Se tiene una tabla de 10 números con signo de 1 byte almacenados a partir de la dirección $0200. Escriba un fragmento de programa que ordene la tabla de mayor a menor (algoritmo de la burbuja).

Problema 15

Escriba una subrutina en lenguaje ensamblador del AVR que localice el menor y el mayor de los datos de una tabla almacenada en memoria como bytes sin signo, situados a partir de la posición $0100. La subrutina recibirá en R21 el número de elementos de la tabla y devolverá:

- En R22 el valor del menor de los datos
- En R23 el valor del mayor de los datos
- En R24 la distancia entre el mayor y el menor, es decir, el valor absoluto de la diferencia entre el mayor y el menor.

Problema 16

Una tabla con 50 datos de 1 byte, con signo, está almacenada en la SRAM a partir de la dirección $100. Escriba un programa para el AVR que almacene en R1 el número de datos positivos que hay en dicha tabla, en R2 el número de datos negativos y en R0 el número de datos que sean iguales a cero. Se valorará el uso de directivas.

Problema 17

Para el siguiente fragmento de programa de **AVR,** indique los registros y posiciones de memoria que se ven afectados por el siguiente fragmento de código, sabiendo que el valor inicial del Puntero de Pila es SP=$04FF. Debe explicar, instrucción a instrucción, qué hace cada una de ellas. Asimismo, debe rellenar esta tabla:

	R0	R1	R16	R17	R28	R29	SP	MEM	[MEM]
`LDI R16,15`									
`LDI R17,19`									
`PUSH R16`									
`INC R16`									
`STS $100,R16`									
`STS $105,R17`									
`LDI R29,$01`									
`LDI R28,$00`									
`LD R0,Y+`									
`LDD R1,Y+4`									
`MUL R16,R17`									
`POP R28`									
`MOV R29,R16`									

Problema 18

Escriba una subrutina que permita saber si un dato de 8 bits corresponde a un número BCD de dos dígitos es o no correcto. La subrutina analiza el dato suministrado en el registro R16 y devuelve R0=$00 si el dato corresponde a un número BCD y R0=$FF si el dato no es correcto en código BCD. Ningún registro, salvo R0, debe verse modificado tras la ejecución de la subrutina.

Ejemplos: Si R16 = 0011 1001, devuelve R0=$00 porque es correcto
Si R16 = 0011 1100, devuelve R0=$FF porque no es correcto
Si R16 = 1011 0100, devuelve R0=$FF porque no es correcto

Problema 19

En las direcciones $100 y $101 se encuentran almacenados dos números sin signo de 1 byte. Escriba un programa para AVR que almacene en R0 el valor absoluto de la resta de ambos números.

Problema 20

Escriba una subrutina POTENCIA que calcule 2^n (con n<16). El exponente se pasa a la subrutina en el registro R16. La subrutina devuelve en R1:R0 el resultado.

Problema 21

Indique el contenido final de los registros de propósito general que se modifican al ejecutar el código siguiente:

```
LDI R31,0
LDI R30,$0F
ADIW Z,2
LDI R16,1
MOV R17,R16
INC R17
LD R0,Z+
```

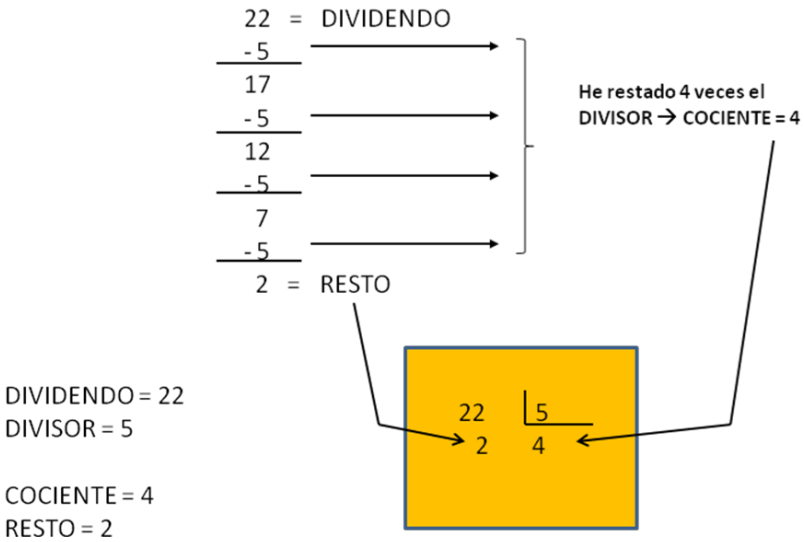
Problema 22

Se conecta un pulsador al pin 6 del puerto B. Hacer un programa para el AVR que establezca el pin 6 del puerto B como entrada. Que active la resistencia de "pull-up" de ese pin y que cuente en un registro R20 cuántas veces se ha pulsado dicho pulsador. IMPORTANTE: Si el pulsador sigue pulsado sólo debe contar una vez.

Problema 23

a) Hacer una subrutina, llamada DIVISION, que divida el DIVIDENDO (que debe estar en R20) entre el DIVISOR (que debe estar en R21). El resultado de la división será el COCIENTE en R0 y el RESTO en R1. Hacer la división por el método de "restas sucesivas" que se explica más abajo.
b) Dada una tabla de 32 datos de 8 bits, situados en la SRAM a partir de la posición $100, calcular, usando la subrutina DIVISION, cuántos de ellos son múltiplos de 5. Guardar el resultado en la dirección $200 de la SRAM.

División por "restas sucesivas"



Problema 24

Se desea realizar el sistema de control de la puerta de un garaje que opere de la siguiente forma:

Señales de entrada del sistema:

LLAVE: "1"cuando el usuario introduce la llave en la cerradura
ABIERTA: "1" cuando la puerta está completamente abierta
CERRADA: "1" cuando la puerta está completamente cerrada
SENSOR: "1" cuando el coche atraviesa la puerta

Señales de salida del sistema:

ABRIR: activa el motor de la puerta para abrirla ("1")
CERRAR: activa el motor de la puerta para cerrarla ("1")

Proceso:

1.- Inicio: el sistema está a la espera de que el usuario introduzca la llave en la cerradura.
2.- Apertura: el sistema activa el motor para abrir la puerta. Hay que mantener activa esta orden hasta que la puerta esté completamente abierta.
3.- Espera: una vez que la puerta está completamente abierta, el sistema espera a que transcurra 1 minuto **(suponga que ya existe una subrutina denominada DELAY que cuenta un minuto, eso es, CALL DELAY genera un retardo de 1 minuto).**
4.- Cierre: transcurrido el minuto de espera, el sistema activa el motor para cerrar la puerta. Hay que mantener activa esta orden hasta que la puerta esté completamente cerrada. Si durante este proceso, el sensor detecta que un coche está atravesando la puerta, deberá interrumpirse el cierre de la misma y volver al proceso de apertura.

Se pide:

a) Realice una asignación de señales a puertos del microcontrolador AVR que realizará el control del sistema, indicando las líneas de código que configuran dichos puertos. (No active resistencias de PULL-UP en las entradas). (1,5 puntos)
b) Suponiendo que ya están configurados los puertos, escriba el código para AVR que realice el control de la puerta del garaje.

Problema 25

Con un microcontrolador con arquitectura AVR, se quiere controlar un puente de lavado de coches. El coche está parado en todo momento y es el puente el que se mueve. En el puente de lavado están instalados los siguientes elementos: salidas de agua a presión, salidas de jabón, salidas de aire caliente a presión, salidas de cera y salidas de control del motor (para indicar sentido de ida o de vuelta).
Con objeto de detectar si ha finalizado el recorrido de "ida y vuelta", existen también dos células fotoeléctricas, CELULA-A (en posición la inicial), y CELULA-B (en el extremo contrario, final del recorrido).
Por otra parte, el usuario puede elegir dos tipos de lavado posibles: con cera Tortuga (lavado largo, 5 pasadas) o sin cera Tortuga (lavado corto, 4 pasadas); para ello, un lector de fichas de lavado existente genera un 1 ó un 0, según el tipo de lavado.

Entradas al microcontrolador AVR:

PULSADOR (necesita activar PULL-UP): orden de comienzo del ciclo de lavado.
TIPO (tipo de lavado): 1 si es lavado largo, 0 si es lavado corto.
CELULA-A (inicio recorrido): 1 si el puente está en la posición inicial, 0 si no.
CELULA-B (final recorrido de ida): 1 si el puente está en final del recorrido, 0 si no.

Salidas de la unidad de control: (todas activas en alto)

AGUA, RODILLOS, JABÓN, CERA, AIRE, MOTOR –IDA, MOTOR-VUELTA

El movimiento del puente se controla con dos señales MOTOR-IDA y MOTOR-VUELTA, dependiendo del tipo de movimiento que queremos que realice

PROCESO DE LAVADO (CON CERA, 5 PASADAS/ SIN CERA, 4 PASADAS)

1) AGUA A PRESIÓN y JABON (ida y vuelta)
2) RODILLOS Y AGUA A PRESIÓN (ida y vuelta)
3) AGUA A PRESIÓN (ida y vuelta)
4) CERA (SÓLO CICLO LARGO) (ida y vuelta)
5) AIRE CALIENTE(ida y vuelta)

Se pide:
Escriba un programa para el microcontrolador AVR que configura adecuadamente los puertos del microcontrolador, así como el código del programa que se necesitaría para controlar el puente de lavado.

Problema 26

A) Se quiere realizar el control del sistema de llenado de un depósito de agua para riego agrícola con un sistema basado en el microcontrolador AVR. Una vez dada la orden de puesta en marcha a través de un pulsador (Xs), cuando el sistema detecte que el nivel de agua del depósito está por debajo de un determinado nivel 1, pone en marcha una motobomba (B1) para sacar agua de un acuífero. La bomba se mantiene en marcha hasta que el agua alcanza el nivel 2 (lógicamente, por encima del nivel 1). (En el depósito están instalados 2 sensores de nivel L1 y L2 que se generan un 1 lógico cuando están sumergidos en agua).
Asigne una distribución adecuada de los pines de E/S del microcontrolador y realice el programa de control correspondiente.

B) Se ha detectado que el sistema anterior falla cuando el sistema de riego saca más caudal de agua del depósito que el que es capaz de reponer la motobomba B1. Para resolver este problema, se ha pensado en instalar una segunda motobomba B2, que entra en funcionamiento sólo cuando el nivel de agua del depósito está por debajo de un nivel mínimo (un tercer sensor Lmín, por debajo de L1, nos daría esa información). En resumen:
   1. Cuando el agua desciende por debajo de L1, entra en funcionamiento B1 hasta alcanzar L2.
   2. Si la bomba B1 no es suficiente y sigue descendiendo el nivel hasta alcanzar Lmín, entraría en funcionamiento la segunda bomba B2; ambas bombas siguen funcionando hasta alcanzar el nivel L2.

Escriba un nuevo programa que contemple esta modificación.

Señales de entrada del sistema:

Xs: pulsador para la orden de comienzo (necesita pull-up)
L1: sensor de nivel 1 (L1=1 cuando está sumergido en agua)
L2: sensor de nivel 2 (L2=1 cuando está sumergido en agua)
Lmín: sensor de nivel mínimo (Lmín=1 cuando está sumergido en agua)

Señales de salida del sistema:

B1: puesta en marcha motobomba 1 (B1=1 para que la bomba B1 funcione)
B2: puesta en marcha motobomba 2 (B2=1 para que la bomba B2 funcione)

Problema 27

Con ayuda del microcontrolador AVR ATmega328p, se desea realizar el sistema de control de una bomba de riego agrícola. El control de encendido/apagado de la bomba se realiza en función de dos parámetros, la humedad del terreno y la luz del día.
Para medir la humedad se tiene un sensor conectado a Puerto B que entrega un dato de 8 bits: los bits PB6-PB0 indican el valor concreto del nivel de humedad (de 0 a127), mientras que el bit PB7 indica que hay un dato válido de humedad disponible para su lectura (PB7=1); si PB7=0, el sensor está procesando un dato y no es fiable su lectura.
Por otra parte, en PC0 se tiene conectado un sensor de luz: PC0=1 día, PC0=0 noche.
Por último, la bomba de riego se activa en el puerto PD0: (PD0=1 bomba en marcha).
La puesta en marcha del sistema se realiza con un pulsador conectado a PC1 (requiere *pull-up).*
Realice un programa de control de riego que configure adecuadamente los puertos y que ponga en marcha la bomba de riego cuando se den las siguientes condiciones:

- Si la humedad es menor que 50 y es de día
- Si la humedad es menor que 70 y es de noche
- Si el dato de humedad no es válido (PB7=0), la bomba de riego debe mantener su estado (encendida o apagada)
- Para cualquier otro caso, la bomba debe estar apagada.