



Grado en Ingeniería Informática - Ingeniería del Software

Estructura de Computadores - Prueba 2 - Junio de 2011

Apellidos, Nombre: _____

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

Problema 1. Realice un programa que calcule la cantidad de números negativos existentes en un vector de N elementos. Utilice adecuadamente las definiciones del fragmento de programa indicado a continuación para la solución

```
EQU DIR_VECTOR=0x100
EQU RESULTADO=0x150
EQU N_ELEMENTOS=0x0A
....
BREAK
```

- (a) Resuelva el problema si los elementos del vector son de tamaño *byte*. (2 puntos)
- (b) Resuelva el problema si los elementos del vector son de tamaño *word*. (2 puntos)
- (c) Indique si existen casos en los que se pudiera producir desbordamiento en el resultado del programa que ha realizado. Explique adecuadamente cada uno de los casos. (1 punto)

Problema 2. Describa la operación que realiza cada fragmento de código **en su conjunto**.

- (a) Fragmento 1: No es válida una respuesta explicando instrucción a instrucción. (1 punto)

```
CLR R0
OUT DDRB,R0
SBIS PINB,4
RJMP UN_SITIO
RJMP OTRO_SITIO
```

SOLUCIÓN:

- (b) Fragmento 2: No es válida una respuesta explicando instrucción a instrucción. (2 puntos)

```
LDI R16,0x88
LDI R17,0x99
LDI R18,0x0A
STS OCR1AH,R16
STS OCR1AL,R17
STS TCCR1B,R18
SBI TIMSK1,1
```

SOLUCIÓN:

- (c) Fragmento 3: No es válida una respuesta explicando instrucción a instrucción. (2 puntos)

```
.CSEG
.ORG 0x0000
JMP PROGRAMA
.ORG 0x0016
JMP HAZ_ALGO
PROGRAMA:
CALL INICIALIZA
DUERME:
SLEEP
RJMP DUERME
HAZ_ALGO:
SBIS PORTB,0
RJMP A
RJMP B
A: SBI PORTB,0
RETI
B: CBI PORTB,0
RETI
```

SOLUCIÓN: ¿que puede ser INICIALIZA?

Solución 1a

```
.include "m328Pdef.inc"

.EQU DIR_VECTOR=0x0100
.EQU RESULTADO=0x0150
.EQU N_ELEMENTOS=0x0A

; Contar el número de negativos

ldi r16,N_ELEMENTOS
clr r0
ldi xh,high(DIR_VECTOR) ; ldi xh,0x01
ldi xl,low(DIR_VECTOR) ; ldi xl,0x00
bucle:
ld r17,x+
rol r17 ; if c==1 es negativo
brcs es_negativo

sigue:
dec r16
brne bucle
sts RESULTADO,r0
break

es_negativo:
inc r0
jmp sigue
```

Solución 1b

El programa sólo cambia en la línea marcada en rojo

```
.include "m328Pdef.inc"

.EQU DIR_VECTOR=0x0100
.EQU RESULTADO=0x0150
.EQU N_ELEMENTOS=0x0A

; Contar el número de negativos

ldi r16,N_ELEMENTOS
clr r0
ldi xh,high(DIR_VECTOR) ; ldi xh,0x01
ldi xl,low(DIR_VECTOR) ; ldi xl,0x00
bucle:
ld r17,x+
rol r17 ; if c==1 es negativo
brcs es_negativo

sigue:
ld r17,x+
dec r16
brne bucle
sts RESULTADO,r0
break

es_negativo:
inc r0
jmp sigue
```



Grado en Ingeniería Informática - Ingeniería del Software

Estructura de Computadores - Prueba 2 - Junio de 2012

Apellidos, Nombre: _____

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

Problema 1. Referente al microcontrolador ATmega328 indique las respuestas que crea adecuadas. Al menos siempre es una válida.

- (a) El registro SP (puntero de pila) se utiliza en: (1 punto)
 - con las operaciones aritméticas.
 - en las llamadas a funciones.
 - en las interrupciones.

- (b) Durante la ejecución de una instrucción con modo de direccionamiento inmediato: (1 punto)
 - El procesador siempre accede a la memoria RAM.
 - Este modo de direccionamiento no accede a memoria RAM.
 - El operando puede estar incluido en el OPCODE de la propia instrucción.

- (c) Durante la ejecución de una instrucción con modo de direccionamiento indirecto: (1 punto)
 - El procesador siempre accede a la memoria RAM.
 - Este modo de direccionamiento no accede a memoria RAM.
 - El operando puede estar incluido en el OPCODE de la propia instrucción.

- (d) El microcontrolador ATmega328 incluye dos memorias: (1 punto)
 - Una es una memoria RAM y otra una memoria FLASH.
 - La memoria FLASH se puede leer desde un programa en ejecución dentro de micro.
 - La memoria RAM ocupa el mapa completo de memoria de datos del microcontrolador.

Problema 2. Indique el resultado final de los registros tras la ejecución de cada uno de los siguientes programas:

| <p>(a) (1 punto)</p> <pre style="margin-left: 40px;">LDI R16,\$10 SBRS R16,4 LSL R16 LSR R16 BREAK</pre> <table border="1" style="margin-left: auto; margin-right: auto; text-align: center;"> <tr><th colspan="2">Valor final en Hexadecimal</th></tr> <tr><td style="width: 50px;">R16</td><td>0x80</td></tr> </table> | Valor final en Hexadecimal | | R16 | 0x80 | <p>(b) (1 punto)</p> <pre style="margin-left: 40px;">SER R24 CLR R25 ADIW R24,1</pre> <table border="1" style="margin-left: auto; margin-right: auto; text-align: center;"> <tr><th colspan="2">V. finales hexadecimal</th></tr> <tr><td style="width: 50px;">R24</td><td>0x00</td></tr> <tr><td>R25</td><td>0x01</td></tr> </table> | V. finales hexadecimal | | R24 | 0x00 | R25 | 0x01 | <p>(c) (1 punto)</p> <pre style="margin-left: 40px;">LDI R22,\$10 LDI R23,\$11 CLR R24 CLR R25 B: ADIW R24,1 CP R24,R22 CPC R25,R23 BRGE SALTO RJMP B SALTO: BREAK</pre> <table border="1" style="margin-left: auto; margin-right: auto; text-align: center;"> <tr><th colspan="2">V. finales hexadecimal</th></tr> <tr><td style="width: 50px;">R24</td><td>0x10</td></tr> <tr><td>R25</td><td>0x11</td></tr> </table> | V. finales hexadecimal | | R24 | 0x10 | R25 | 0x11 |
|--|----------------------------|--|------------|-------------|--|------------------------|--|------------|-------------|------------|-------------|---|------------------------|--|------------|-------------|------------|-------------|
| Valor final en Hexadecimal | | | | | | | | | | | | | | | | | | |
| R16 | 0x80 | | | | | | | | | | | | | | | | | |
| V. finales hexadecimal | | | | | | | | | | | | | | | | | | |
| R24 | 0x00 | | | | | | | | | | | | | | | | | |
| R25 | 0x01 | | | | | | | | | | | | | | | | | |
| V. finales hexadecimal | | | | | | | | | | | | | | | | | | |
| R24 | 0x10 | | | | | | | | | | | | | | | | | |
| R25 | 0x11 | | | | | | | | | | | | | | | | | |

(d) (1 punto)

```
LDI R16,5
LDI R17,6
LDI XH,02
LDI XL,00
ST X+,R16
ST X,R17
BREAK
```

| Contenido de la MEMORIA (HEX) | |
|-------------------------------|------|
| \$0200 | 0x05 |
| \$0201 | 0x06 |
| \$0202 | ???? |
| \$0203 | ???? |

(e) (1 punto)

```
LDI R16,4
LDI XH,02
LDI XL,04
BUC:
ST -X,R16
DEC R16
BRNE BUC
```

| Contenido de la MEMORIA (HEX) | |
|-------------------------------|------|
| \$0200 | 0x01 |
| \$0201 | 0x02 |
| \$0202 | 0x03 |
| \$0203 | 0x04 |

Problema 3. Realice un programa que llame a la subrutina "HAZALGO" si el número de 8 bits almacenado en R12 es negativo, sino debe continuar sin llamar a la subrutina

Escriba el programa aquí

(1 punto)



Grado en Ingeniería Informática - Ingeniería del Software

Estructura de Computadores - Prueba 2 - Junio de 2012

Apellidos, Nombre: _____

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

Problema 1. Referente al microcontrolador ATmega328 indique las respuestas que crea adecuadas. Al menos siempre es una válida.

- (a) El registro SP (puntero de pila) se utiliza en: (1 punto)
 - las operaciones aritméticas.
 - las llamadas a funciones.
 - las interrupciones.

- (b) Durante la ejecución de una instrucción con modo de direccionamiento inmediato: (1 punto)
 - El procesador siempre accede a la memoria RAM.
 - Este modo de direccionamiento no accede a memoria RAM.
 - El operando puede estar incluido en el OPCODE de la propia instrucción.

- (c) Durante la ejecución de una instrucción con modo de direccionamiento indirecto: (1 punto)
 - El procesador siempre accede a la memoria RAM.
 - Este modo de direccionamiento no accede a memoria RAM.
 - El operando puede estar incluido en el OPCODE de la propia instrucción.

- (d) El microcontrolador ATmega328 incluye dos memorias: (1 punto)
 - Una es una memoria RAM y otra una memoria FLASH.
 - La memoria FLASH se puede leer desde un programa en ejecución dentro de micro.
 - La memoria RAM ocupa el mapa completo de memoria de datos del microcontrolador.

Problema 2. Indique el resultado final de los registros tras la ejecución de cada uno de los siguientes programas:

| <p style="text-align: center;">(a) (1 punto)</p> <pre style="text-align: center;">SER R24 CLR R25 ADIW R24,1</pre> <table border="1" style="margin-left: auto; margin-right: auto; text-align: center;"> <thead> <tr><th colspan="2">Valores finales en hexadecimal</th></tr> </thead> <tbody> <tr><td style="width: 50px;">R24</td><td style="width: 50px;">0x00</td></tr> <tr><td>R25</td><td>0x01</td></tr> </tbody> </table> | Valores finales en hexadecimal | | R24 | 0x00 | R25 | 0x01 | <p style="text-align: center;">(b) (1 punto)</p> <pre style="text-align: center;">LDI R22,\$10 LDI R23,\$11 CLR R24 CLR R25 B:ADIW R24,1 CP R24,R22 CPC R25,R23 BRGE UN_SALTO RJMP B SALTO: BREAK</pre> <table border="1" style="margin-left: auto; margin-right: auto; text-align: center;"> <thead> <tr><th colspan="2">V. finales hexadecimal</th></tr> </thead> <tbody> <tr><td style="width: 50px;">R24</td><td style="width: 50px;">0x10</td></tr> <tr><td>R25</td><td>0x11</td></tr> </tbody> </table> | V. finales hexadecimal | | R24 | 0x10 | R25 | 0x11 | <p style="text-align: center;">(c) (1 punto)</p> <pre style="text-align: center;">LDI R16,5 LDI R17,6 LDI XH,02 LDI XL,00 ST X+,R16 ST X,R17 BREAK</pre> <table border="1" style="margin-left: auto; margin-right: auto; text-align: center;"> <thead> <tr><th colspan="2">Contenido de la MEMORIA (HEX)</th></tr> </thead> <tbody> <tr><td style="width: 50px;">\$0200</td><td style="width: 50px;">0x05</td></tr> <tr><td>\$0201</td><td>0x06</td></tr> <tr><td>\$0202</td><td>???</td></tr> <tr><td>\$0203</td><td>???</td></tr> </tbody> </table> | Contenido de la MEMORIA (HEX) | | \$0200 | 0x05 | \$0201 | 0x06 | \$0202 | ??? | \$0203 | ??? |
|--|--------------------------------|--|-----|------|-----|------|---|------------------------|--|-----|------|-----|------|---|-------------------------------|--|--------|------|--------|------|--------|-----|--------|-----|
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | |
| R24 | 0x00 | | | | | | | | | | | | | | | | | | | | | | | |
| R25 | 0x01 | | | | | | | | | | | | | | | | | | | | | | | |
| V. finales hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | |
| R24 | 0x10 | | | | | | | | | | | | | | | | | | | | | | | |
| R25 | 0x11 | | | | | | | | | | | | | | | | | | | | | | | |
| Contenido de la MEMORIA (HEX) | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0200 | 0x05 | | | | | | | | | | | | | | | | | | | | | | | |
| \$0201 | 0x06 | | | | | | | | | | | | | | | | | | | | | | | |
| \$0202 | ??? | | | | | | | | | | | | | | | | | | | | | | | |
| \$0203 | ??? | | | | | | | | | | | | | | | | | | | | | | | |

Problema 3. Realice los siguientes programas:

- (a) En las posiciones de memoria \$200 y \$201 hay un número de 16bits, en las posiciones \$202 y \$203 hay otro número de 16bits. Tanto la posición \$200 como la posición \$202 contienen el bytes MSB del número (parte más significativa). Realice la suma de 16bits y almacenándola a partir de la posición \$204.
- (b) Realice un programa que sume dos números y salte a la etiqueta ERROR si ha ocurrido overflow.
- (c) Realice un programa llame a la subrutina "HAZALGO" si el número de 8 bits almacenado en R12 es negativo, de lo contrario el programa continuará su ejecución.

(a) Escriba el programa aquí (1 punto)

(b) Escriba el programa aquí (1 punto)

(c) Escriba el programa aquí (1 punto)



Grado en Ingeniería Informática - Ingeniería del Software

Estructura de Computadores - Prueba 2 - Junio de 2013

| | | | | |
|--|--|--|--|--|
| | | | | |
|--|--|--|--|--|

Apellidos, Nombre: _____

Problema 1. Describa brevemente, en ámbito de la asignatura:

- (a) Los modos de direccionamiento inmediato, directo e indirecto. (1 punto)
- (b) El uso de la pila en los procesadores. (1 punto)
- (c) La diferencia entre una subrutina y una interrupción. (1 punto)

Problema 2. Escoja una instrucción del microprocesador CS2010 con modo de direccionamiento directo a memoria y especifique ciclo a ciclo su fase de ejecución. No es necesario realizar una carta ASM, pero si debe numerar los ciclos y especificar en cada uno de ellos las señales que se activan. (1 punto)

Problema 3. Indique el valor final de la memoria/registros indicados para cada uno de estos programas

```

CLR R25
SER R24
ADIW R24, 1
STS $100, R25
STS $101, R24
BREAK
  
```

| DIR | CONT |
|--------|------|
| \$0100 | |
| \$0101 | |

(1 punto)

```

LDI R16, 1
CLR R17
B: INC R17
ROL R16
SBR5 R17, 2
JMP B
BREAK
  
```

| REG | VAL |
|-----|-----|
| R16 | |
| R17 | |

(1 punto)

```

LDI R16, $02
CLR R17
LDI XH, $01
LDI XL, $00
B:
ST X+, R16
LSL R16
INC R17
CPI R17, $08
BRNE B
BREAK
  
```

| DIR | CONT |
|--------|------|
| \$0100 | |
| \$0101 | |
| \$0102 | |
| \$0103 | |
| \$0104 | |
| \$0105 | |
| \$0106 | |
| \$0107 | |
| \$0108 | |

(1 punto)

Problema 4. Realice lo siguiente en lenguaje ensamblador del microcontrolador AVR ATmega328P:

- (a) Un programa que reste dos números de 32 bits almacenados a partir de la dirección \$0200 de la memoria RAM. La parte alta de cada número está en la primera posición de memoria de los 4 bytes que forman cada número. Tras la resta si el resultado es cero debe saltar a la etiqueta FIN. (1 punto)
- (b) Una subrutina que reciba en R19 un número de 8bits en complemento a 2 y calcule su valor absoluto, devolviéndolo en el mismo registro R19. (1 punto)
- (c) Usando la subrutina anterior, realice un programa que calcule los valores absolutos de cada elemento de una tabla de números de 1byte almacenadas en memoria a partir de la dirección \$0100. La tabla original debe sobrescribirse con los nuevos valores. (1 punto)



Grado en Ingeniería Informática - Ingeniería del Software

Estructura de Computadores - Prueba 2 - Junio 2014

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

Apellidos, Nombre: _____

Problema 1. Indique el resultado final de los registros y la memoria tras la ejecución de cada uno de los siguientes programas:

| <p><i>(1 punto)</i></p> <pre>LDI R25,\$10 LDI R24,\$D0 ADIW R24,\$30 BREAK</pre> <table border="1" style="width: 100%; margin-top: 20px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="width: 50%; text-align: center;">R24</td><td style="width: 50%;"></td></tr> <tr><td style="text-align: center;">R25</td><td></td></tr> </table> | Valores finales en hexadecimal | | R24 | | R25 | | <p><i>(1 punto)</i></p> <pre>CLR R0 CLR R1 LDI R16,1 CLR R17 LDI R20,\$88 LDI R21,\$99 B:ADD R0,R16 ADC R1,R17 CP R0,R20 CPC R1,R21 BRLO B BREAK</pre> <table border="1" style="width: 100%; margin-top: 20px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="width: 50%; text-align: center;">R0</td><td style="width: 50%;"></td></tr> <tr><td style="text-align: center;">R1</td><td></td></tr> </table> | Valores finales en hexadecimal | | R0 | | R1 | | <p><i>(1 punto)</i></p> <pre>LDI R16,\$C0 LDI XH,\$02 LDI XL,\$00 LDI R18,\$03 B:ST X+,R16 LSL R16 BRCS B BREAK</pre> <table border="1" style="width: 100%; margin-top: 20px;"> <tr><th colspan="2" style="text-align: center;">Contenido de la MEMORIA (HEX)</th></tr> <tr><td style="width: 50%; text-align: center;">\$0200</td><td style="width: 50%;"></td></tr> <tr><td style="text-align: center;">\$0201</td><td></td></tr> <tr><td style="text-align: center;">\$0202</td><td></td></tr> <tr><td style="text-align: center;">\$0203</td><td></td></tr> <tr><td style="text-align: center;">\$0204</td><td></td></tr> </table> | Contenido de la MEMORIA (HEX) | | \$0200 | | \$0201 | | \$0202 | | \$0203 | | \$0204 | | <p><i>(1 punto)</i></p> <pre>LDI XH,\$02 LDI XL,\$00 LDI R18,\$03 B:SBRC R18,4 JMP F ST X+,R18 ROL R18 JMP B F:BREAK</pre> <table border="1" style="width: 100%; margin-top: 20px;"> <tr><th colspan="2" style="text-align: center;">Contenido de la MEMORIA (HEX)</th></tr> <tr><td style="width: 50%; text-align: center;">\$0200</td><td style="width: 50%;"></td></tr> <tr><td style="text-align: center;">\$0201</td><td></td></tr> <tr><td style="text-align: center;">\$0202</td><td></td></tr> <tr><td style="text-align: center;">\$0203</td><td></td></tr> </table> | Contenido de la MEMORIA (HEX) | | \$0200 | | \$0201 | | \$0202 | | \$0203 | |
|--|--------------------------------|--|-----|--|-----|--|--|--------------------------------|--|----|--|----|--|--|-------------------------------|--|--------|--|--------|--|--------|--|--------|--|--------|--|---|-------------------------------|--|--------|--|--------|--|--------|--|--------|--|
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R24 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R25 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Contenido de la MEMORIA (HEX) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0200 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0201 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0202 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0203 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0204 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Contenido de la MEMORIA (HEX) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0200 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0201 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0202 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0203 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Problema 2. En la tabla se muestra un ejemplo de ejecución de una suma de 24bits, concretamente: 13FAB0+D4539A=E84E4A.

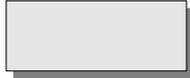
- (a) Realice una subrutina en ensamblador para el microcontrolador ATmega328P que sume dos números de 24bits almacenados en memoria como se indica en la tabla. *(1.5 puntos)*
- (b) Realice un programa que haga uso de la subrutina para sumar 15 números consecutivos en memoria almacenados de la misma forma. El resultado final siempre está a partir de la dirección \$0100. *(1.5 puntos)*

| Dirección | Contenido |
|-----------|-----------|
| \$0100 | \$E8 |
| \$0101 | \$4E |
| \$0102 | \$4A |
| \$0103 | \$13 |
| \$0104 | \$FA |
| \$0105 | \$B0 |
| \$0106 | \$D4 |
| \$0107 | \$53 |
| \$0108 | \$9A |

Problema 3. Escriba un programa en ensamblador que dada una dirección de memoria donde comienza un vector de 50 elementos de tamaño byte, cuente el número de elementos positivos. El resultado se guardará en R0.

- (a) Realice el programa para el CS2010 considerando que el vector comienza en la dirección \$F0 de la memoria. *(1.5 puntos)*
- (b) Realice el programa para el microcontrolador ATmega328P considerando que el vector comienza en la dirección \$01F0. *(1.5 puntos)*

Apellidos, Nombre: _____



Problema 1. Indique el resultado final de los registros ejecución de cada uno de los siguientes programas:

| <p align="center"><i>(0.5 puntos)</i></p> <pre>LDI R16,\$10 LDI R17,\$F0 LDI R18,\$20 LDI R19,\$20 ADD R17,R19 ADC R16,R18 BREAK</pre> <table border="1" style="width:100%; border-collapse: collapse; margin-top: 20px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="text-align: center;">R16</td><td style="text-align: center;">\$31</td></tr> <tr><td style="text-align: center;">R17</td><td style="text-align: center;">\$10</td></tr> </table> | Valores finales en hexadecimal | | R16 | \$31 | R17 | \$10 | <p align="center"><i>(0.5 puntos)</i></p> <pre>CLR R17 LDI R16,8 B: LSL R17 ORI R17,\$01 DEC R16 BRNE B BREAK</pre> <table border="1" style="width:100%; border-collapse: collapse; margin-top: 20px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="text-align: center;">R16</td><td style="text-align: center;">00</td></tr> <tr><td style="text-align: center;">R17</td><td style="text-align: center;">FF</td></tr> </table> | Valores finales en hexadecimal | | R16 | 00 | R17 | FF | <p align="center"><i>(0.5 puntos)</i></p> <pre>CLR R0 LDI R16,\$C0 LDI R17,\$02 CP R16,R17 BRGE FIN INC R0 FIN: BREAK</pre> <table border="1" style="width:100%; border-collapse: collapse; margin-top: 20px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="text-align: center;">R0</td><td style="text-align: center;">01</td></tr> </table> | Valores finales en hexadecimal | | R0 | 01 | <p align="center"><i>(0.5 puntos)</i></p> <pre>LDS R0,\$0150 MOV R1,R0 NEG R1 ADD R0,R1 BREAK</pre> <table border="1" style="width:100%; border-collapse: collapse; margin-top: 20px;"> <tr><th colspan="2" style="text-align: center;">Contenido de la MEMORIA (HEX)</th></tr> <tr><td style="text-align: center;">\$0150</td><td style="text-align: center;">05</td></tr> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="text-align: center;">R0</td><td style="text-align: center;">00</td></tr> <tr><td style="text-align: center;">R1</td><td style="text-align: center;">FB</td></tr> </table> | Contenido de la MEMORIA (HEX) | | \$0150 | 05 | Valores finales en hexadecimal | | R0 | 00 | R1 | FB |
|--|--------------------------------|--|-----|-------------|-----|-------------|---|--------------------------------|--|-----|-----------|-----|-----------|--|--------------------------------|--|----|-----------|---|-------------------------------|--|--------|----|--------------------------------|--|----|-----------|----|-----------|
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R16 | \$31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R17 | \$10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R16 | 00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R17 | FF | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R0 | 01 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Contenido de la MEMORIA (HEX) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0150 | 05 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R0 | 00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R1 | FB | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Problema 2. El programa mostrado en el siguiente cuadro está escrito en lenguaje ensamblador para el microcontrolador ATmega328P. Responda las siguientes cuestiones:

```
.EQU N=10
.EQU Q=20

LDI XH,$01
LDI XL,$00
LDI YH,$01
LDI YL,$50
LDI R16,N
BUCLE:
LD R17,X+
CPI R17,Q
BRSH SIGUE
ST Y+,R17
SIGUE:
DEC R16
BRNE BUCLE
BREAK
```

- (a)** Describa verbalmente cual es la función del programa en su conjunto y como se ve afectado el resultado del programa si se cambian los valores de las definiciones de N y Q. *(1 punto)*

- (b)** Realice un programa equivalente para el procesador CS2010, considere que las direcciones \$0100 y \$0150 deben ser sustituidas por las direcciones \$00 y \$50 respectivamente. *(1 punto)*

Problema 3. Usando el lenguaje ensamblador del microcontrolador ATMEGA328P realice lo siguiente:

- (a) Escriba una **subrutina** que devuelva en R0 el número de elementos negativos de un vector. Los parámetros de entrada son: R1: el número de elementos y X dirección de inicio del vector . (2 puntos)
- (b) Escriba un programa que use la subrutina anterior para calcular en R2 el número de elementos negativos de los dos siguientes vectores: (2 puntos)
- Vector 1: vector de 15 elementos que comienza en \$0120
 - Vector 2: vector de 30 elementos que comienza en \$0150

Problema 4. Escriba un programa en lenguaje ensamblador para el microcontrolador ATMEGA328P que concatene dos vectores V1 y V2 de N elementos en un tercer vector V3. El programa debe comenzar el código ensamblador mostrado a continuación. (2 puntos)

```
.include "m328def.inc"
.EQU N=15

.DSEG
.ORG 0X0100
    V1: .BYTE N
    V2: .BYTE N
    V3: .BYTE N+N

.CSEG

; Escriba su programa a partir de aquí
```

Soluciones propuestas

PROBLEMA 3

```
.include "m328def.inc"  
.EQU N=10  
.EQU Q=20
```

```
.EQU V1=$0120  
.EQU V2=$0150  
.EQU N1=15  
.EQU N2=30
```

```
CLR R2 // RESULTADO
```

```
LDI R16,N1  
MOV R1,R16  
LDI XH,HIGH(V1)  
LDI XH,LOW(V1)  
CALL NEGATIVOS  
MOV R2,R0
```

```
LDI R16,N2  
MOV R1,R16  
LDI XH,HIGH(V2)  
LDI XH,LOW(V2)  
CALL NEGATIVOS  
ADD R2,R0  
BREAK
```

```
NEGATIVOS:  
PUSH R16  
CLR R0
```

```
NB:  
LD R16,X+  
SBRC R16,7  
INC R0  
DEC R1  
BRNE NB  
POP R16  
RET
```

PROBLEMA 4

```
.include "m328def.inc"  
.EQU N=15
```

```
.DSEG  
.ORG 0X0100  
V1: .BYTE N  
V2: .BYTE N  
V3: .BYTE N+N
```

```
.CSEG
```

```
LDI R16,N  
LDI XH,HIGH(V1)  
LDI XL,LOW(V1)  
LDI YH,HIGH(V2)  
LDI YL,LOW(V2)  
LDI ZH,HIGH(V3)  
LDI ZL,LOW(V3)  
LDI R17,N
```

```
BUCLE1:  
LD R0,X+  
ST Z+,R0  
DEC R17  
BRNE BUCLE1  
LDI R17,N
```

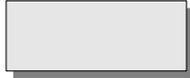
```
BUCLE2:  
LD R0,Y+  
ST Z+,R0  
DEC R17  
BRNE BUCLE2  
RET
```



Grado en Ingeniería Informática - Ingeniería del Software

Estructura de Computadores - Prueba 2 - Junio 2016

Apellidos, Nombre: _____



Problema 1. Indique el resultado final de los registros ejecución de cada uno de los siguientes programas:

| <p>(1 punto)</p> <pre>LDI R16,\$00 LDI R17,\$01 LDI R18,\$20 LDI R19,\$00 SUB R17,R19 SBC R16,R18 BREAK</pre> <table border="1" style="width: 100%; margin-top: 10px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="width: 50%; text-align: center;">R16</td><td style="width: 50%;"></td></tr> <tr><td style="text-align: center;">R17</td><td></td></tr> </table> | Valores finales en hexadecimal | | R16 | | R17 | | <p>(1 punto)</p> <pre>CLR R17 LDI R16,4 B:ROR R16 CPSE R17,R16 JMP B BREAK</pre> <table border="1" style="width: 100%; margin-top: 10px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="width: 50%; text-align: center;">R16</td><td style="width: 50%;"></td></tr> <tr><td style="text-align: center;">R17</td><td></td></tr> </table> | Valores finales en hexadecimal | | R16 | | R17 | | <p>(1 punto)</p> <pre>LDI R16,\$80 LDI R17,\$02 LDI R18,\$04 ADD R16,R16 ADC R17,R17 ADC R18,R18 BREAK</pre> <table border="1" style="width: 100%; margin-top: 10px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="width: 50%; text-align: center;">R16</td><td style="width: 50%;"></td></tr> <tr><td style="text-align: center;">R17</td><td></td></tr> <tr><td style="text-align: center;">R18</td><td></td></tr> </table> | Valores finales en hexadecimal | | R16 | | R17 | | R18 | | <p>(1 punto)</p> <pre>LDS R0,\$0150 MOV R1,R0 NEG R1 ADD R0,R1 BREAK</pre> <table border="1" style="width: 100%; margin-top: 10px;"> <tr><th colspan="2" style="text-align: center;">Contenido de la MEMORIA (HEX)</th></tr> <tr><td style="width: 50%; text-align: center;">\$0150</td><td style="width: 50%; text-align: center;">05</td></tr> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="width: 50%; text-align: center;">R0</td><td style="width: 50%;"></td></tr> <tr><td style="text-align: center;">R1</td><td></td></tr> </table> | Contenido de la MEMORIA (HEX) | | \$0150 | 05 | Valores finales en hexadecimal | | R0 | | R1 | |
|---|--------------------------------|--|-----|--|-----|--|--|--------------------------------|--|-----|--|-----|--|---|--------------------------------|--|-----|--|-----|--|-----|--|--|-------------------------------|--|--------|----|--------------------------------|--|----|--|----|--|
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R18 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Contenido de la MEMORIA (HEX) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0150 | 05 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Problema 2. El programa mostrado en el siguiente cuadro está escrito en lenguaje ensamblador para el microcontrolador ATmega328P. Responda las siguientes cuestiones:

```
.EQU N=10

LDI XH,$01
LDI XL,$00
LDI R16,N
BUCLE:
LD R17,X
NEG R17
ST X+,R17
DEC R16
BRNE BUCLE
BREAK
```

- (a) Describa verbalmente cual es la función del programa en su conjunto y como se ve afectado el resultado del programa si se cambia el valor de la definición de N. (1 punto)
- (b) Realice un programa equivalente para el procesador CS2010, considere sustituir la dirección \$0100 por la dirección \$20. (1 punto)

Problema 3. Usando el lenguaje ensamblador del microcontrolador ATMEGA328P realice lo siguiente:

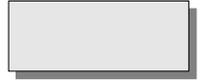
- (a)** Escriba una **subrutina** que devuelva en R0 el número de elementos mayores que el número contenido en el registro R2, siguiendo las siguientes especificaciones: *(2 puntos)*
- Los números son de 8 bits con notación en complemento a 2.
 - Los parámetros de entrada son: R1 el tamaño del vector, R2 el valor a comparar y X dirección de inicio del vector.
 - Debe usar adecuadamente las instrucciones PUSH y POP para que la subrutina no tenga efectos laterales sobre el programa que lo llama.
- (b)** Escriba un programa que use la subrutina anterior dos veces para calcular en R16 el número de elementos mayores de 0x20 del primer vector mas el número de elementos mayores de 0x20 del segundo vector: *(2 puntos)*
- Vector 1: vector de 15 elementos que comienza en \$0120
 - Vector 2: vector de 30 elementos que comienza en \$0150



Grado en Ingeniería Informática - Ingeniería del Software

Estructura de Computadores - Prueba 2 - Junio 2017

Apellidos, Nombre: _____



Problema 1. Dado el programa escrito en lenguaje ensamblador del procesador CS2010, realice lo siguiente:

```
.EQU N=10

LDI R0,N
LDI R1,$10
BUCLE:
LD R2,(R1)
ROL R2
ST (R1),R2
ADDI R1,1
SUBI R0,1
BRZS FIN
JMP BUCLE
FIN:
STOP
```

- (a) Describa verbalmente la función del programa en su conjunto y como se ve afectada la memoria tras la ejecución del mismo. (1 punto)
- (b) Realice una programa equivalente escrito en lenguaje ensamblador de microcontroladores AVR, pero comenzando en la dirección de memoria \$0100 en vez de la \$10. (1 punto)

Problema 2. Indique el resultado final de los registros ejecución de cada uno de los siguientes programas:

| <p>(0.75 puntos)</p> <pre>LDI R16,\$00 LDI R17,\$01 SUB R16,R17 BRMI FIN INC R16 FIN: BREAK</pre> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="text-align: center;">R16</td><td style="text-align: center;">0xFF</td></tr> <tr><td style="text-align: center;">R17</td><td style="text-align: center;">0x01</td></tr> </table> | Valores finales en hexadecimal | | R16 | 0xFF | R17 | 0x01 | <p>(0.75 puntos)</p> <pre>LDI R16,\$F8 PUSH R16 ORI R16,\$0F POP R17 BREAK</pre> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="text-align: center;">R16</td><td style="text-align: center;">0xFF</td></tr> <tr><td style="text-align: center;">R17</td><td style="text-align: center;">0xF8</td></tr> </table> | Valores finales en hexadecimal | | R16 | 0xFF | R17 | 0xF8 | <p>(0.75 puntos)</p> <pre>LDI R16,\$08 B:SBR R17,1 LSL R17 DEC R16 BRNE B BREAK</pre> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="text-align: center;">R16</td><td style="text-align: center;">0xFF</td></tr> <tr><td style="text-align: center;">R17</td><td style="text-align: center;">0xFE</td></tr> </table> | Valores finales en hexadecimal | | R16 | 0xFF | R17 | 0xFE | <p>(0.75 puntos)</p> <pre>LDS R16,\$0150 SER R17 MOVW R18,R16 BREAK</pre> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr><th colspan="2" style="text-align: center;">Contenido de la MEMORIA (HEX)</th></tr> <tr><td style="text-align: center;">\$0150</td><td style="text-align: center;">05</td></tr> <tr><th colspan="2" style="text-align: center;">Valores finales en hexadecimal</th></tr> <tr><td style="text-align: center;">R18</td><td style="text-align: center;">0x05</td></tr> <tr><td style="text-align: center;">R19</td><td style="text-align: center;">0xFF</td></tr> </table> | Contenido de la MEMORIA (HEX) | | \$0150 | 05 | Valores finales en hexadecimal | | R18 | 0x05 | R19 | 0xFF |
|--|--------------------------------|--|------------|------|------------|------|---|--------------------------------|--|------------|------|------------|------|--|--------------------------------|--|------------|------|------------|------|---|-------------------------------|--|---------------|----|--------------------------------|--|------------|------|------------|------|
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R16 | 0xFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R17 | 0x01 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R16 | 0xFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R17 | 0xF8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R16 | 0xFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R17 | 0xFE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Contenido de la MEMORIA (HEX) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| \$0150 | 05 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Valores finales en hexadecimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R18 | 0x05 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R19 | 0xFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Problema 3. Usando el lenguaje ensamblador de microcontroladores AVR realice lo siguiente:

- (a)** Un programa que calcule la siguiente resta: $0x1212 - 0x2424$. Los números están escritos en hexadecimal, son de 16bits y el resultado debe almacenarse a partir de la dirección $0x0100$ de la memoria. *(1 punto)*
- (b)** Realice un programa que incremente en uno cada elemento de un array de 50 bytes almacenado en memoria a partir de la dirección $0x0150$. *(1.5 puntos)*
- (c)** Escriba una subrutina cuyo parámetro de entrada es el registro R0 que contiene un número con signo. La subrutina debe calcular el valor absoluto de R0 y devolverlo usando el registro R1. *(1 punto)*
- (d)** Realice un programa que utilice la subrutina anterior para calcular la suma de todos los valores absolutos de un array de 20 elementos situado en la dirección de memoria $0x0150$. *(1.5 puntos)*

Solución

Problema 1

a) El programa recorre un vector de 10 elementos que comienza en la dirección 0x10. Cada elemento del vector es rotado a la izquierda.

b)

```
ldi r16,N
ldi xh,0x01
ldi xl,0x00
bucle:
ld r2,x
rol r2
st x+,r2
dec r16
brne bucle
break
```

Problema 3

a)

Esquema de la resta a:

$$\begin{array}{r} 1212 \\ -2424 \\ \hline \end{array} \quad \begin{array}{r} r17:r16 \\ - r19:r18 \\ \hline \\ r19:r18 \end{array}$$

```
ldi r17,0x12
ldi r16,0x12
ldi r19,0x24
ldi r18,0x24
sub r18,r16
sbc r19,r17
sts 0x0100,r19
sts 0x0101,r18
```

b)

```
ldi r16,50
ldi xh,0x01
ldi xl,0x50
bucle:
ld r17,x
inc r17
st x+,r17
dec r16
brne bucle
break
```

c)

```
subrutina:
mov r1,r0
sbrc r1,7
neg r1
ret
```

d)

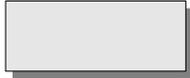
```
ldi r16,20
ldi xh,0x01
ldi xl,0x50
bucle:
ld r0,x
call subrutina
st x+,r1
dec r16
brne bucle
break;
```



Grado en Ingeniería Informática - Ingeniería del Software

Estructura de Computadores - Prueba 2 - Junio 2018

Apellidos, Nombre: _____



Problema 1. Dado el programa escrito en lenguaje ensamblador del procesador CS2010, realice lo siguiente:

| <pre>.EQU N=7 LDI R1,0 LDI R2,1 BUCLE: ST (R1),R2 ADDI R2,2 ADDI R1,1 CPI R1,N BRLO BUCLE STOP</pre> | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Dir</th> <th style="width: 90%;">Contenido Memoria de Programa</th> </tr> </thead> <tbody> <tr><td>00</td><td></td></tr> <tr><td>01</td><td></td></tr> <tr><td>02</td><td></td></tr> <tr><td>03</td><td></td></tr> <tr><td>04</td><td></td></tr> <tr><td>05</td><td></td></tr> <tr><td>06</td><td></td></tr> <tr><td>07</td><td></td></tr> <tr><td>08</td><td></td></tr> </tbody> </table> | Dir | Contenido Memoria de Programa | 00 | | 01 | | 02 | | 03 | | 04 | | 05 | | 06 | | 07 | | 08 | | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Dir</th> <th style="width: 90%;">Contenido Memoria de Datos</th> </tr> </thead> <tbody> <tr><td>00</td><td></td></tr> <tr><td>01</td><td></td></tr> <tr><td>02</td><td></td></tr> <tr><td>03</td><td></td></tr> <tr><td>04</td><td></td></tr> <tr><td>05</td><td></td></tr> <tr><td>06</td><td></td></tr> <tr><td>07</td><td></td></tr> <tr><td>08</td><td></td></tr> </tbody> </table> | Dir | Contenido Memoria de Datos | 00 | | 01 | | 02 | | 03 | | 04 | | 05 | | 06 | | 07 | | 08 | |
|---|--|-----|-------------------------------|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|---|-----|----------------------------|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|
| Dir | Contenido Memoria de Programa | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 02 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 03 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 04 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 05 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 06 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 07 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 08 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dir | Contenido Memoria de Datos | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 02 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 03 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 04 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 05 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 06 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 07 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 08 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

- (a) Rellene el código máquina correspondiente al programa en la tabla *Contenido Memoria de Programa*. Rellene la tabla en **binario**. (1 punto)
- (b) Rellene la tabla *Contenido Memoria de Datos* con los datos correspondientes tras la ejecución del programa. Rellene la tabla en **decimal**. (1 punto)
- (c) Realice un programa equivalente escrito en lenguaje ensamblador de microcontroladores AVR, pero comenzando en la dirección de memoria 0x0100 en vez de la 0x00. (1 punto)

Problema 2. Indique el resultado final de los registros tras la ejecución de cada uno de los siguientes programas, escritos en lenguaje ensamblador para microcontroladores AVR.

(0.25 puntos)

| | |
|--|--|
| <pre>CLR R16 LDI R17,0x44 ORI R16,0x71 AND R17,R16 BREAK</pre> | |
| Valores finales en hexadecimal | |
| R16 | |
| R17 | |

(0.25 puntos)

| | |
|--|--|
| <pre>LDI R16,0x01 LDI R17,0x80 ADD R17,R17 ADC R16,R16 BREAK</pre> | |
| Valores finales en hexadecimal | |
| R16 | |
| R17 | |

(0.25 puntos)

| | |
|--|--|
| <pre>LDI R16,0x08 SBRS R16,3 SUBI R16,2 SUBI R16,4 BREAK</pre> | |
| Valor final en hexadecimal | |
| R16 | |

(0.25 puntos)

| | |
|--------------------------------------|--|
| <pre>EOR R16,R16 COM R16 BREAK</pre> | |
| Valor final en hexadecimal | |
| R16 | |

Problema 3. El programa mostrado pretende realizar el siguiente cálculo: Escribir en la dirección 0x0150 de la memoria RAM el resultado de sumar todos los bytes de un vector de tamaño N.

```
.include <m328def.inc>
```

```
.EQU N=8
```

```
LDI X,0x0150  
CLR R1  
CLR R16
```

```
bucle:
```

```
LD R0,X  
ADD R1,R0  
ADDI X,1  
INC R16  
BRNE bucle  
STS 0x100,R1
```

```
break
```

(a) Indique las 2 instrucciones que no son correctas en lenguaje ensamblador de AVR. (0.5 puntos)

(b) Indique cual es la forma correcta de realizar estas instrucciones. (0.5 puntos)

(c) Además de las 2 instrucciones erróneas el bucle no está realizado correctamente. Proponga una solución correcta para el programa para que realice la operación solicitada. (1 punto)

(d) Otro posible problema es el desbordamiento en el resultado. Proponga una nueva solución donde el resultado sea de 16bits y no se produzca el desbordamiento. (1 punto)

Problema 4. Usando el lenguaje ensamblador de los microcontroladores AVR realice lo siguiente:

(a) Una subrutina que calcule una suma de 16 bits cumpliendo las siguientes especificaciones: (1 punto)

1. El primer número está almacenado en la dirección de memoria 0x0102 y 0x0103, estando la parte más significativa del número en la dirección de memoria 0x0102
2. El segundo número está almacenado en la dirección de memoria 0x0104 y 0x0105, estando la parte más significativa del número en la dirección de memoria 0x0104
3. El resultado se devolverá en los registros R1:R0, estando la parte más significativa en el registro R1
4. La subrutina no debe tener efectos laterales sobre el programa principal, es decir, todos los registros que se utilicen debe salvarse y recuperarse usando la pila del sistema.

(b) Usando la subrutina anterior realice un programa capaz de sumar los números: 0x1234 + 0x3210. Los números están escritos en hexadecimal y el resultado debe almacenarse a partir de la dirección 0x0100 de la memoria. (1 punto)

Problema 5. Realice un programa que configure los dos primeros pines del puerto C como salidas. Tras ello, el programa debe entrar en un bucle infinito generando una señal cuadrada en PINC0 y la inversa en PINC1, siguiendo las siguientes especificaciones: (0.75 puntos)

1. En el primer paso establecerá PORTC1=0V y PORTC0=5V **simultáneamente**.
2. En el segundo paso establecerá PORTC1=5V y PORTC0=0V **simultáneamente**.
3. Tras el paso anterior saltará incondicionalmente al paso 1.

¿Cree que la señal medida en el osciloscopio será exactamente cuadrada?. Justifique su respuesta.

(0.25 puntos)

Nota: El registro DDRC configura como entrada/salida el puerto C y el registro PORTC se utiliza para escribir en el puerto.