

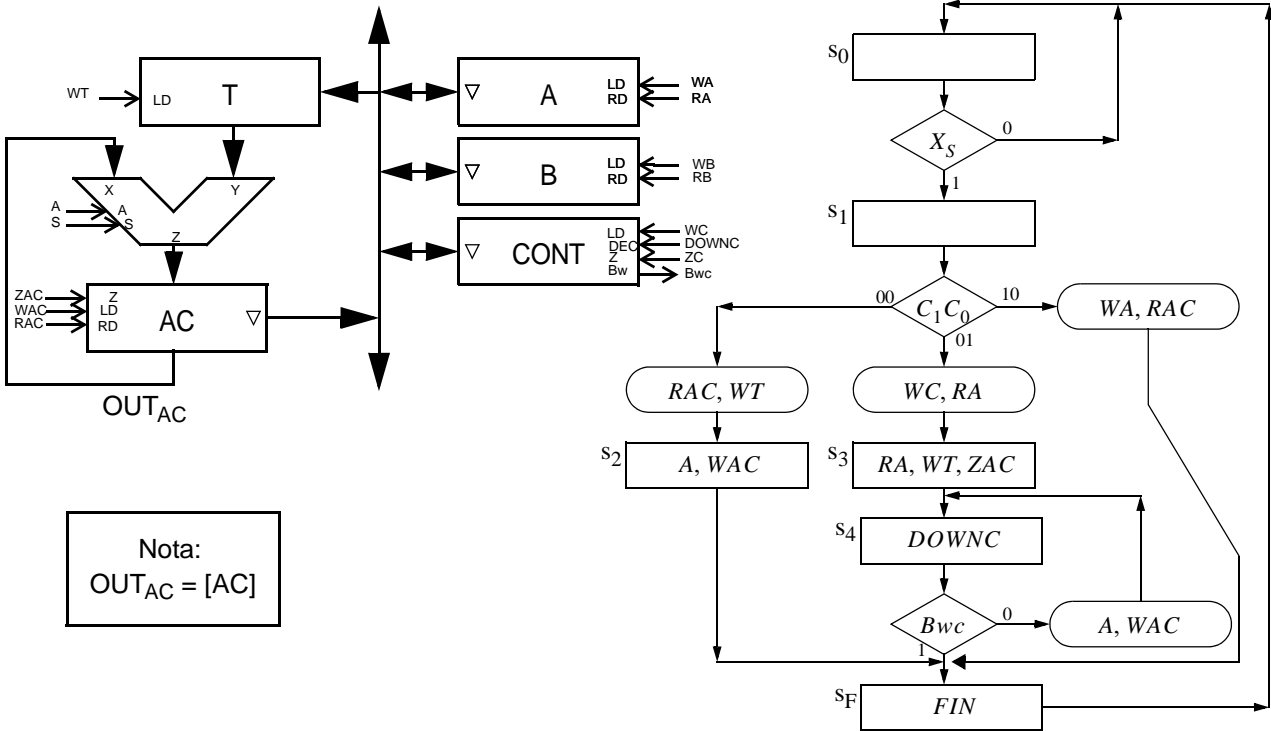
Apellidos:.....

1	2	3

Nombre:.....

**Duración 3 h. Todos los problemas valen igual**

1.- Las siguientes figuras representan una unidad de datos y la carta ASM de control de un sistema digital.



Nota:  
OUT<sub>AC</sub> = [AC]

- a) Diseñe la unidad de control para la carta ASM de la figura, utilizando la técnica de diseño basada en un biestable por estado.
  - b) Describa las distintas macrooperaciones (en función de C<sub>1</sub> y C<sub>0</sub>) que se realizan.
  - c) Describa formalmente (tabla a nivel RT) los siguientes componentes: CONT, AC, T y A. Interprete las señales según el nombre utilizado.
  - d) Realice la carta ASM de datos equivalente a la carta ASM de control de la figura.
- 2.- Se desea añadir la instrucción `IJMP Rn` al conjunto de instrucciones del CS2010. Esta instrucción realiza un salto indirecto a la dirección de memoria contenida en el registro Rn. La instrucción sigue el formato A y no usa el campo registro destino (almacena Rn en registro fuente).
- a) Indique qué cambios habría que realizar a la unidad de datos del sistema para que pudiera implementarse dicho modo.
  - b) Descomponga la nueva instrucción en microoperaciones.
  - c) Repita los apartados a) y b) para la instrucción `PUSH PC`, que guarda el contenido del PC en la pila. ¿Qué formato de instrucción usaría en este caso? Elija entre los actualmente disponibles.
- 3.- La subrutina `ContBit0Byte` cuenta el número de bits con valor 0 del registro R2. Por su parte, la subrutina `ContBit0` usa la subrutina anterior para contar el número de bits a 0 de un vector de 16 bytes almacenados a partir de la dirección \$100 y lo almacena en el registro R7.
- a) Explique ambos algoritmos con pseudocódigo o diagramas de flujo.
  - b) Escriba ambas subrutinas en ensamblador del AVR.

***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Para un  $\mu P$  de AB[16] y DB[8], construya una unidad de memoria principal con memoria no-volátil de 8KBytes en las direcciones mayores y 40KBytes de memoria de lectura y escritura en las direcciones menores. Se dispone de memorias RAMs de 16Kx8 y 8Kx8 y de ROMs de 8Kx8, todas con un único bus de datos de 8 bits y selección de chip activa en L; las RAMs poseen una señal de R/W' (escribe si R/W' = L).

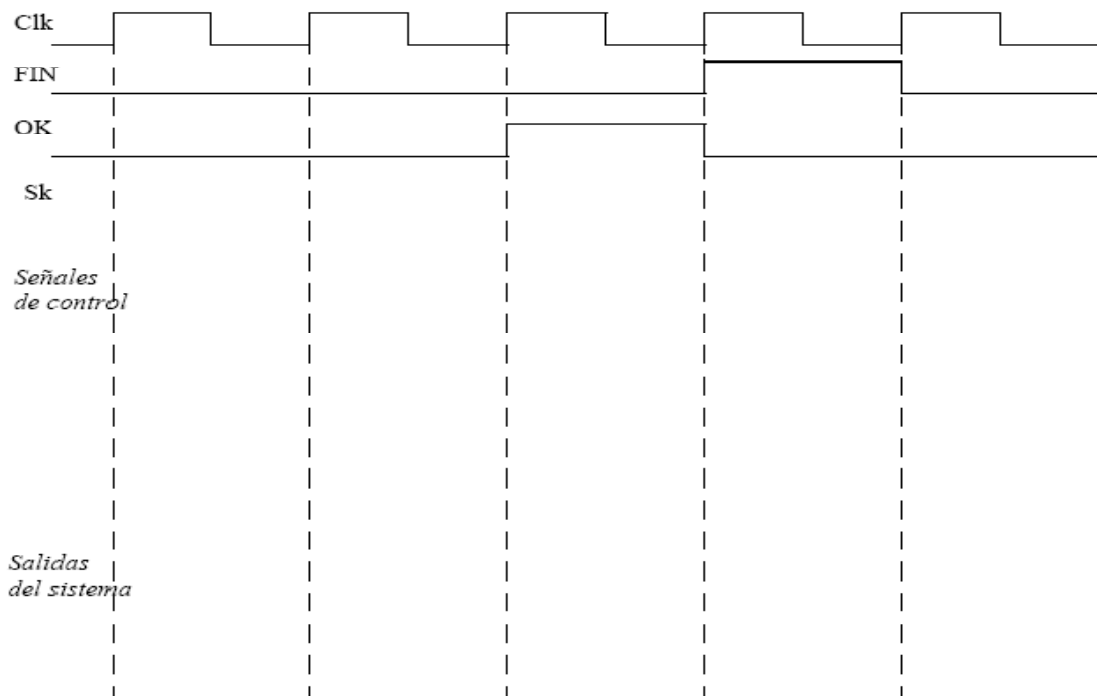
- a. Diseñe el circuito de memoria. Debe dar el mapa, el circuito de decodificación y, para cada chip de memoria, especificar todas sus conexiones.
- b. Indique, en hexadecimal, todas las direcciones que se pueden escribir en el bus AB del  $\mu P$  para acceder a la palabra de dirección física "1 1 1 1 1 0 ... 0" en cada uno de los dispositivos de memoria utilizados.
- c. Indique, en hexadecimal, las palabras físicas a las que se accede con las direcciones lógicas AB = \$5050, \$8765 y \$E044.

Apellidos, Nombre: \_\_\_\_\_

Problema 2. Un sistema digital recibe datos de 8 bits por su bus de entradas en paralelo ( $X_{in}$ ) de la siguiente forma. Cuando se activa la señal  $X_s$  (durante 1 ciclo de reloj), comienza la operación. Desde fuera se pone un dato de referencia ( $X_{in}=REF$ ) en el bus de entrada. El sistema debe capturar ese dato cuando se active la señal de entrada de validación  $VREF$ . Después se va a enviar una secuencia de 1024 datos por sus entradas  $X_{in}$ . Cada vez que uno de estos datos esté estable en  $X_{in}$ , se enviará un 1 por una señal de entrada ( $OK$ ) al sistema digital.

El sistema digital deberá contar cuántas de las palabras recibidas son mayores que el dato  $REF$ , y sacar ese número a través de las salidas adecuadas. (Puede facilitarle la solución que suponga que al menos una de las palabras recibidas será menor que  $REF$ .) Cuando ese número esté disponible, el sistema activará una señal de  $FIN$ .

- Dibuje el sistema digital como un todo, identificando y dimensionando adecuadamente todas sus entradas y salidas. (1 punto)
- Diseñe una Unidad de Datos específica para resolver este sistema, evitando registros con alta impedancia. (No debe usar la Unidad de Datos de la calculadora.) (3 puntos)
- Especifique la carta ASM de datos y la de control, y diseñe la unidad de control mediante la estrategia de 1 biestable por estado. (4 puntos)
- En el siguiente diagrama de tiempo, se han dibujado 5 ciclos de reloj que corresponden a la parte final de la operación del sistema digital. Para situarse, observe que  $FIN$  está activa en el cuarto de los ciclos dibujados. Se ha supuesto que el dato en  $X_{in}$  es mayor que  $REF$  cuando  $OK=1$ . Suponga que antes de ese dato se llevaban contados 35 valores mayores que  $REF$ . Complete ese diagrama dando los valores adecuados a  $S_k$  (que son los estados del controlador que haya usado en su solución), a Señales de control (indicando las que se activen en su caso) y a Salidas del sistema (las que usted haya incluido en su diseño). (2 puntos)



Apellidos, Nombre: \_\_\_\_\_

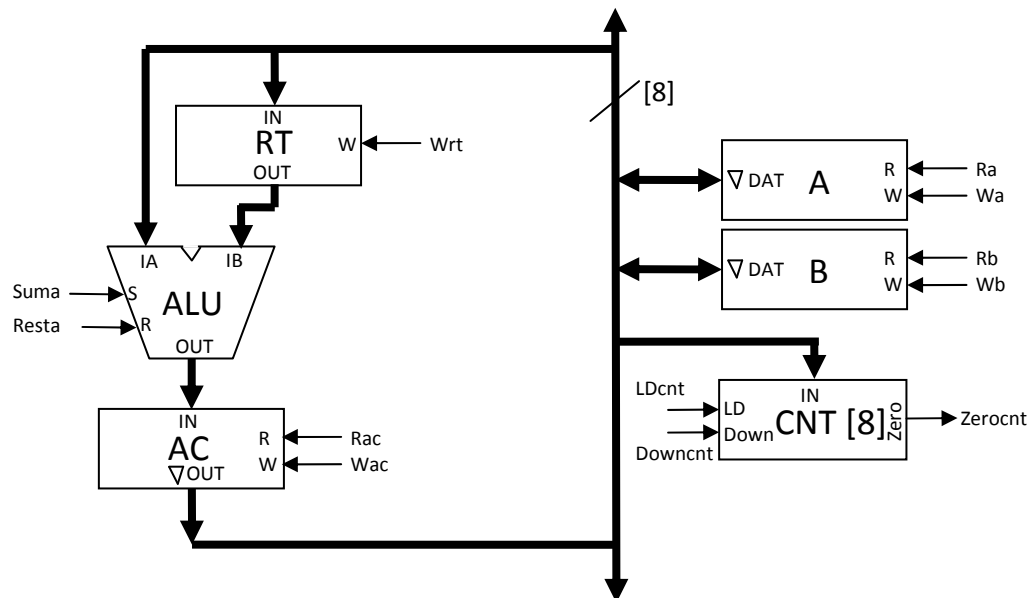
Problema 3. Diseñe un programa en lenguaje ensamblador del AVR que realice lo siguiente: dado un vector de tamaño N cuyos elementos van desde 1 hasta n [ $i_1, i_2, i_3 \dots i_n$ ], debe devolver el mismo vector con los elementos en orden inverso, es decir, desde n hasta 1 [ $i_n, \dots i_3, i_2, i_1$ ].

Datos a tener en cuenta:  $0 < N < 200$ , el vector se encuentra almacenado en posiciones consecutivas a partir de la \$100 en el segmento de datos, cada elemento es de tamaño byte, el vector es único (el programa debe escribir sobre el vector inicial).

- a. Explique mediante un diagrama de flujo o un pseudocódigo las principales decisiones de diseño de su programa.
- b. Escriba el código ensamblador AVR del programa principal. (Suponga que el vector ya está inicializado en memoria).
- c. Añada el código complementario necesario para hacer una prueba de su programa con un simulador, inicializando un vector de 10 elementos con valores iniciales [1, 2, 3 ... 10] y el resto de valores necesarios.
- d. Describa brevemente los cambios necesarios del programa si los elementos del vector fueran tamaño *word*.

Apellidos, Nombre: \_\_\_\_\_

Problema 1. Considere la Unidad de Datos de la figura. Podrá observar que RT tiene salidas incondicionales, que los registros A y B son iguales y tienen I/O bidireccional, que AC es similar a ellos pero con líneas unidireccionales de I y de O, que hay un contador descendente con carga en paralelo y señal de fin de ciclo de cuenta (paso por cero), y una ALU que solamente suma ( $IA+IB$ ) o resta ( $IA-IB$ ).



Las descripciones Verilog de los módulos usados tienen como cabeceras las siguientes:

```
Module regtipo1 (input clk, W, input [7:0] IN, output [7:0] OUT);
Module regtipo2 (input clk, W, R, input [7:0] IN, output [7:0] OUT);
Module regtipo3 (input clk, W, R, inout [7:0] DAT);
Module contador (input clk, LD, DOWN, input [7:0] IN, output ZERO);
Module uniaritlog (input S,R, input [7:0] IA, IB, output [7:0] OUT);
```

- Describa en Verilog (estructural) la Unidad de Datos de la figura.
- Describa en Verilog el contador descendente mód. 256, CNT.
- Se desea realizar como macro-operación el producto de números sin signo A y B,  $A \leftarrow A \times B$ , mediante sumas sucesivas. Indique la carta ASM de datos para ello (con Xs y FIN).
- Indique la carta ASM de control y diseñe el circuito de control con la técnica de un biestable por estado.

Apellidos, Nombre: \_\_\_\_\_

Problema 2. Se desea añadir al CS2010 una nueva instrucción, SWAP Ra,Rb. Esta instrucción intercambiará los contenidos de ambos registros:  $Ra \leftrightarrow Rb$ .

- a. Indique:
  - i. Un código de instrucción válido para esta instrucción, sin eliminar ninguna de las existentes.
  - ii. Si es posible realizar esta instrucción sin modificar la Unidad de Datos o si hay que modificarla, en cuyo caso, debe mostrar cómo se modifica.
- b. Para la solución aportada en el apartado anterior, desarrolle la secuencia de micro-operaciones en la ejecución de SWAP, indicando, además, las señales de control implicadas.

Apellidos, Nombre: \_\_\_\_\_

Problema 3. Se tienen dos vectores,  $P(i)$  y  $Q(i)$  cada uno con  $N$  componentes consistentes en números con signo de 8 bits. Los primeros elementos de cada vector están en las posiciones  $\text{primdirP}$  y  $\text{primdirQ}$ .

- a. Escriba una rutina para inicializar  $N$ ,  $\text{primdirP}$  y  $\text{primdirQ}$ . Asígneles los valores 10, \$100 y \$120.
- b. Se pretende que  $P(i)$  se quede con la componente mayor y  $Q(i)$  con la menor. Esto es, si inicialmente  $P(1)=+3$  y  $Q(1) =+7$ , y  $P(2)= -3$  y  $Q(2)=-7$ , los valores finales serán:  $P(1)=+7$ ,  $Q(1) =+3$ , y  $P(2)= -3$  y  $Q(2)= -7$ .
  - i. Indique un organigrama que resuelva este problema.
  - ii. Escriba el programa en ensamblador.

***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Un sistema digital tiene como entradas la señal  $X_s$ , una señal  $T$  y un bus de 16 líneas,  $BUS$ , por el que entrarán números en complemento a 2. El sistema comenzará tras la activación de  $X_s$ . Primero, cuando se active la señal  $T$ , el dato  $A$  se almacenará en un registro interno  $RA$ . Después, y tras activarse nuevamente  $T$ , el dato  $B$  se almacenará en otro registro  $RB$ . Se desea almacenar el mayor de los dos datos en  $RA$  y el menor de ellos en  $RB$ .

- a. Diseñe una **unidad de datos específica** para este problema.
- b. Escriba la carta ASM de la unidad de control.

Problema 2. Tomando como punto de partida la arquitectura del CS2010, determine a qué afecta o qué cambios debe hacer para los siguientes casos (tomados por separado):

- a. Se ampliará la memoria de programa.
- b. Se ampliará la longitud de las instrucciones.
- c. Se ampliará la memoria de datos.
- d. Se ampliará la longitud de los datos.
- e. Se permitirá que el registro  $AC$  sea un operando de  $ALU$ .
- f. Se permitirán instrucciones con más de una palabra de código máquina.

Problema 3. La subrutina *cpex* compara dos números de una longitud  $L$ , que es variable. El primer número se encuentra apuntado por  $X$  y el segundo por  $Y$ . En  $R16$  se almacena la longitud de los números en bytes. *cpex* afecta a los bits del  $SR$  para que posteriormente se pueda saltar en función del resultado.

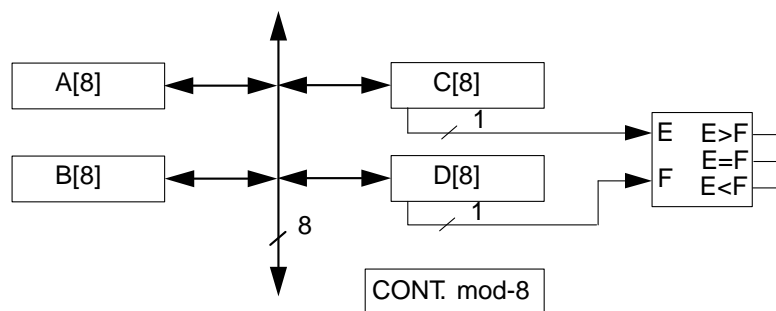
- a. Describa el algoritmo mediante pseudocódigo o diagrama de flujo y escriba *cpex* en ensamblador del AVR.
- b. Suponga que tiene almacenados en  $\$100$  y  $\$200$  sendos números con signo de 16 bits. Escriba un trozo de código que use *cpex* (llamándola con  $CALL\ cpex$ ) para comparar ambos números y dejar el mayor en  $\$300$ .
- c. Repita el ejercicio anterior suponiendo que los números son de 32 bits y sin signo.



**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 3** **13-diciembre-2012**

**Apellidos, Nombre:** \_\_\_\_\_

- Problema 1. Para la unidad de datos que se presenta, se quiere realizar un sistema digital capaz de comparar dos números de 8 bits (A y B), y almacenar en A el mayor de ellos y en B el menor. Tras finalizar la operación, el sistema generará una señal de FIN.
- Defina correctamente las operaciones de los registros.
  - Obtenga la carta ASM.
  - Sin añadir elementos nuevos, ¿se puede simplificar la arquitectura de esta unidad de proceso? Razone la respuesta.



- Problema 2. Se desea cambiar la semántica de las instrucciones ST y LD del CS2010 para que usen el modo de direccionamiento indirecto con postincremento. Su semántica es idéntica a la del modo indirecto normal salvo que, tras realizarse el acceso a memoria, el registro base se incrementa en una unidad.

- Indique qué cambios habría que realizar a la unidad de datos del sistema para que pudiera implementarse dicho modo (véase figura del CS2010).
- Descomponga las nuevas instrucciones en microoperaciones.

- Problema 3. La subrutina strwc recibe en X la dirección del primer carácter de una cadena de caracteres (el final de la cadena viene marcado por un byte a 0) y devuelve en el par de registros R16:R17 el número de letras de la cadena y en el par R18:R19 el número de palabras de la misma. Las cadenas sólo contienen letras y espacios. No se tendrá en cuenta el caso de que haya caracteres de control (Escape, Intro, etc.), dígitos y otros caracteres imprimibles. *Ejemplo: " El perro de San Roque ya no tiene rabo " devuelve 9 palabras y 30 letras.*

- Explique con palabras el algoritmo que va a emplear para la subrutina strwc.
- Escriba la subrutina en ensamblador del AVR.

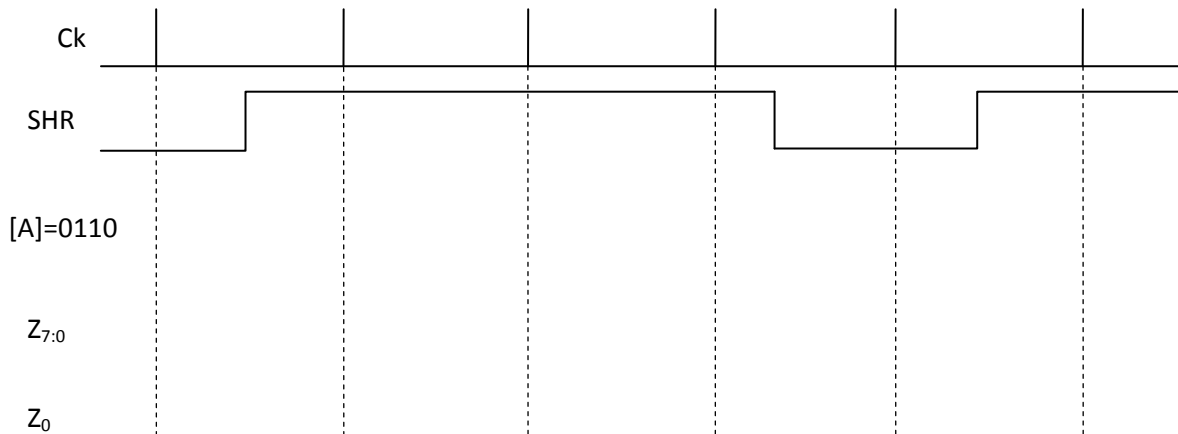
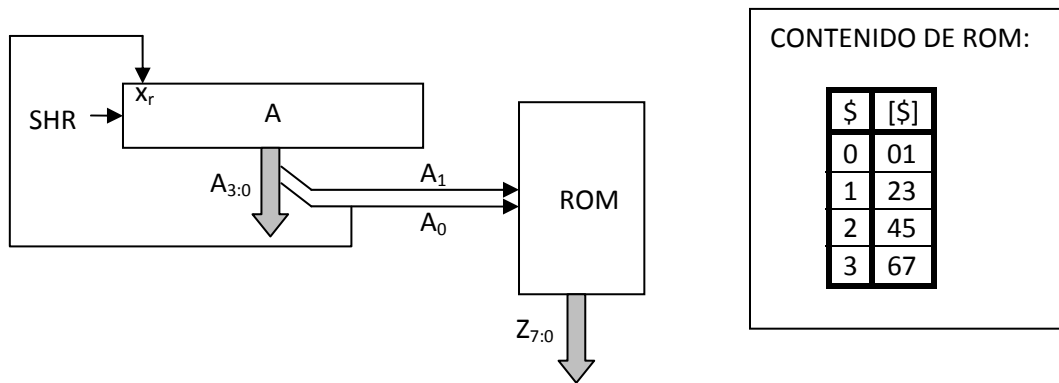
# ESTRUCTURA DE COMPUTADORES - GII - IC

## Examen Final Convocatoria 1

13-junio-2013

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Inicialmente el registro A del circuito de la figura contiene el dato "0110". Complete el cronograma dando A, Z<sub>7:0</sub> en hexadecimal y dibujando la forma de onda de la salida Z<sub>0</sub>.



Problema 2. Se desea añadir al CS2010 una nueva instrucción, que permita sumar un registro Rd con el dato almacenado en una posición de memoria dada por su dirección, dir. Rd es el destino del resultado.

- a. Indique:
  - i. Proponga una sintaxis (mnemónico y argumentos) para la instrucción que siga las formas habituales del conjunto de instrucciones del CS2010.
  - ii. Un código de instrucción válido para esta instrucción, sin eliminar ninguna de las existentes.
  - iii. Si es posible realizar esta instrucción sin modificar la Unidad de Datos o si hay que modificarla, en cuyo caso, debe mostrar cómo se modifica.
- b. Para la solución aportada en el apartado anterior, desarrolle la secuencia de micro-operaciones en la ejecución de la nueva instrucción, indicando, además, las señales de control implicadas.

Problema 3. Escriba en ensamblador del AVR la subrutina *CalcParVec* que calcula la paridad par a cada uno de los elementos de un vector de bytes. *CalcParVec* recibe en X la dirección del primer componente del vector y en R16 el número de elementos del mismo. Esta subrutina debe procesar los 7 bits menos significativos de cada elemento del vector y escribir en el bit más significativo el bit de paridad calculado (se debe ignorar el valor previo de dicho bit, que puede venir a 0 o a 1).

Debe explicar y documentar adecuadamente su solución.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 2** **10-septiembre-2013**

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. La hoja adjunta contiene una descripción verilog de la unidad de control de un sistema digital.

- a. Escriba la carta ASM.
- b. Dibuje el circuito con la estrategia de un biestable por estado.

Problema 2. El siguiente programa en el ámbito del computador CS2010 viene parte escrito en código máquina y parte en lenguaje ensamblador:

	\$F D 0 A	
	\$F C 0 0	
	\$1 8 A 0	
	\$F 9 B 0	
siguiente:	\$0 A 0 1	
	SUB R2,R0	;A
	BRCS etiq	
	ADDI R4,1	
atrás:	SUBI R5,1	
	CPI R5,0	;B
	BREQ etiq2	
	ADDI R1,1	
	JMP siguiente	
etiq:	JMP atrás	
etiq2:	STS \$C0, R4	
	STOP	

- a) Indique cuál es la instrucción (mnemónico y argumentos) en ensamblador referenciada en cada una de las líneas escritas en código máquina.
- b) Traduzca a código máquina las instrucciones contenidas entre la marca A y B (ambas incluidas).
- c) Indique la evolución de los valores de registros y/o direcciones de memorias involucrados en cada instrucción.
- d) Indique verbalmente qué tarea GLOBAL desarrolla el programa completo.

Problema 3. Se tiene un sistema basado en un procesador que maneja datos de tamaño  $W$  y  $B$  (par e impar) en un espacio de direccionamiento lógico de 16MB. Los buses son  $DB[16]$ ,  $AB[23]$  y señales  $BE_1BE_0$ . Implementado con chips de memoria RAM de  $256K \times 8$  con señal de selección activa a nivel bajo, el sistema tiene construido la siguiente memoria principal:

- 256kW a partir de la dirección 0.
  - 256kW a partir de la dirección  $\$1\ 0\ 0\ 0\ 0\ 0$ .
  - 256kW en las últimas direcciones ( $\$F\ F\ F\ F\ F$  y anteriores)
- a) Obtenga la tabla mapa de memoria y el sistema de decodificación correspondiente, e indique las conexiones de los chips de memoria con los buses  $AB$  y  $DB$  del procesador.
- b) Responda a las siguientes preguntas sobre accesos a datos en memoria
- 1) Para la dirección lógica  $AB_{lógica}[24] = \$FC0123$ , indique la palabra física (debe identificar el chip y dar en hexadecimal la dirección física) e identificar el tamaño del dato y las líneas del  $DB$  que lo transportan.
  - 2) Repita si  $AB$  lógica[24] =  $\$FC0122$  y el tamaño del dato es  $W$ .
  - 3) Indique qué dirección lógica ( $AB_{lógica}[24]$ ) hay que poner para acceder a la palabra  $\$123$  de cada una de las memorias.

## Hoja adjunta al Problema 1

//Descripción verilog del problema de la prueba

```
module u_control(  
    input xs, clk, busc0, cycont,  
    output reg clsincr, fin, wa, wd, wsuml, wsumh, shrsum, upcont);
```

```
parameter S0=0,S1=1,S2=2,S3=3,SF=4;
```

```
reg [2:0] estado_actual,siguiente_estado;
```

```
always @(posedge clk)  
estado_actual <= siguiente_estado;
```

```
always @( xs, busc0, cycont, estado_actual)  
begin
```

```
    clsincr=0;  
    wa=0;  
    wd=0;  
    wsuml=0;  
    wsumh=0;  
    upcont=0;  
    shrsum=0;  
    fin=0;
```

```
case(estado_actual)
```

```
    S0:  
        if(xs)  
            begin  
                clsincr=1;  
                siguiente_estado = S1;  
            end  
        else  
            siguiente_estado = S0;
```

```
    S1:  
        begin  
            wa=1;  
            wsuml=1;  
            siguiente_estado = S2;  
        end
```

```
    S2:  
        begin  
            wd=1;  
            if(busc0)  
                wsumh=1;  
                siguiente_estado = S3;  
        end
```

```
//Continuación del código
```

```
    S3:  
        begin  
            upcont=1;  
            shrsum=1;  
            if(cycont)  
                siguiente_estado=SF;  
            else  
                siguiente_estado=S2;  
        end  
    SF:  
        begin  
            fin=1;  
            siguiente_estado=S0;  
        end  
    default:  
        siguiente_estado=S0;
```

```
endcase  
end  
endmodule
```

# ESTRUCTURA DE COMPUTADORES – GII – IC

## Examen Final Convocatoria 1

26-junio-2014

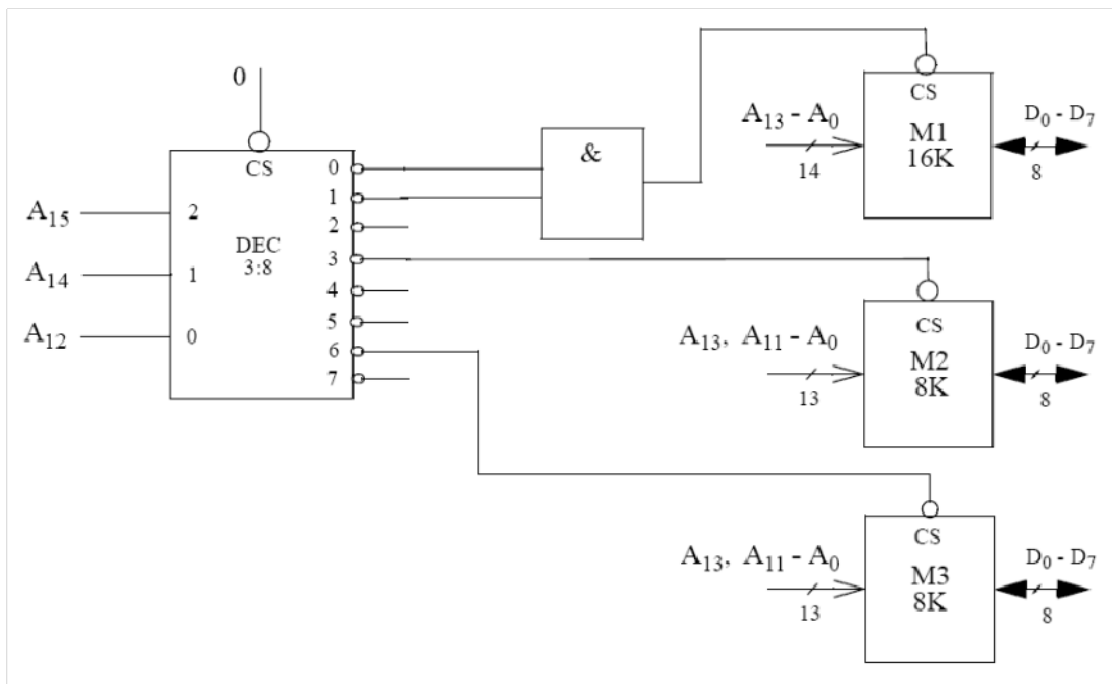
Apellidos, Nombre: \_\_\_\_\_

Problema 1. Se desea transferir ordenadamente el contenido de las memorias M2 y M3 a la memoria M1 (ver figura). Se dispone de la instrucción

**TRANSFIERE (\$ Fuente, \$ Destino)**

la cual transfiere un bloque de 2k palabras consecutivas desde la dirección fuente hacia la destino.

Justifique y escriba el programa necesario. (El sistema interpreta y ejecuta esta instrucción).



Nombre \_\_\_\_\_

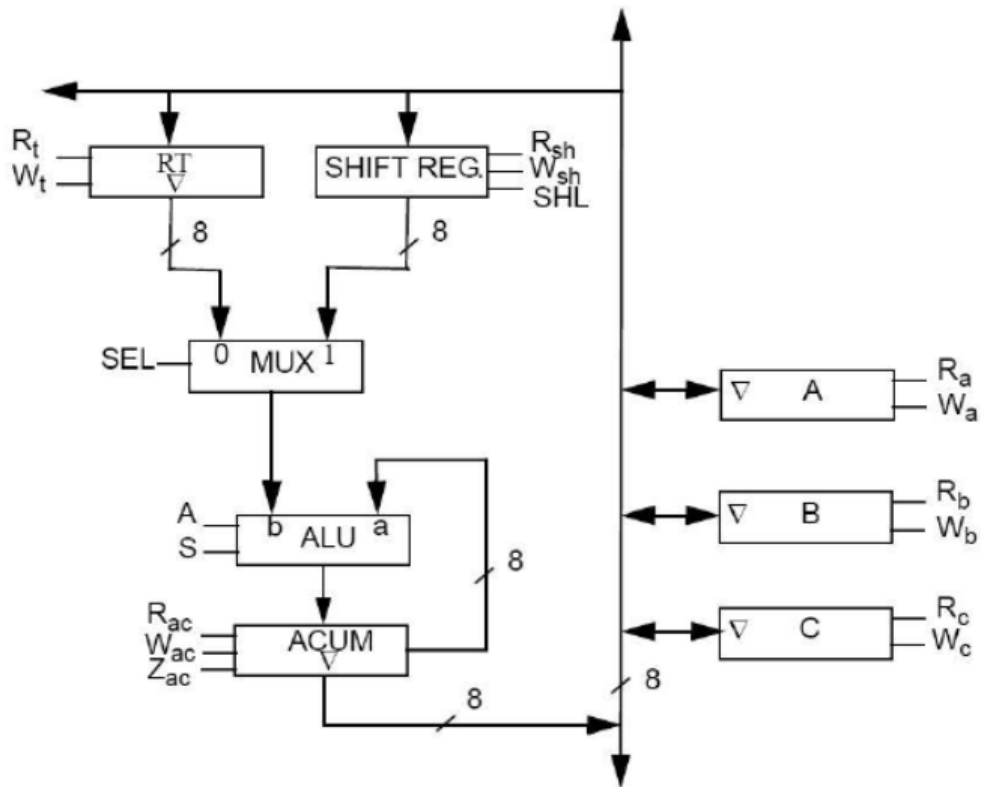
Problema 2. Considere la unidad de datos de la figura.

- Describa en lenguaje RT y en Verilog el registro ACUM.
- Obtenga las cartas ASM de datos y de control de forma que, en función de un bit de entrada de selección de operación ( $I_0$ ), pueda elegirse entre una de las dos macrooperaciones siguientes:

1)  $A \leftarrow A + 2B$

2)  $C \leftarrow A - 2B$

- Diseñe la Unidad de control con la técnica de un biestable por estado).



Nota. ALU suma cuando  $A=1$ :  $a + b$  y resta cuando  $S=1$ :  $a - b$



Nombre \_\_\_\_\_

Problema 3. Considere en el CS2010 la subrutina “subrut” que se muestra. Esta subrutina hace una operación unaria sobre el dato inicial (número sin signo N, no nulo) que hay almacenado en R0 y **devuelve dos valores** como resultado de la operación, uno en R1 y otro en R2. (Durante la subrutina también se usa el registro R3).

subrut:	MOV	R2, R0
	LDI	R1, 0
	LDI	R3, 1
otro:	CP	R2, R3
	BRLO	retorno
	SUB	R2, R3
	ADDI	R1, 1
	ADDI	R3, 2
	JMP	otro
retorno:	RET	

- 3.1. Haga el flujo de datos suponiendo que N=7.
- 3.2. Repita para N=16.
- 3.3. Identifique qué operación unaria es la que hace esta subrutina, eligiendo entre las siguientes:  
a/Inverso (1/N).            b/Opuesto de la magnitud (-|N|).            c/Cuadrado (N<sup>2</sup>)  
d/Raíz cuadrada (N<sup>1/2</sup>). e/Módulo sobre 10 (mod(N)|<sub>10</sub>)  
(Si lo necesitara para determinar la operación, puede desarrollar otros flujos de datos, aconsejándole que elija números pequeños del tipo 3, 4, 5, etc.).
- 3.4. Considere ahora el siguiente programa cuyas instrucciones 3, 4 y 5 se dan en código máquina y que, como pueden ver, llama a la subrutina anterior.
  - 3.4.1. Escriba las correspondientes instrucciones en ensamblador.
  - 3.4.2. Indique qué tarea hace el programa.

; El código máquina está en HEXADECIMAL

```
LDI R5, $50
LDI R6, $40
FF30
lazo: 0807
C701
CALL subrut
ST (R6), R1
ST (R5), R2
ADDI R6, 1
ADDI R5, 1
CPI R7, $3A
BRLO lazo
STOP
```

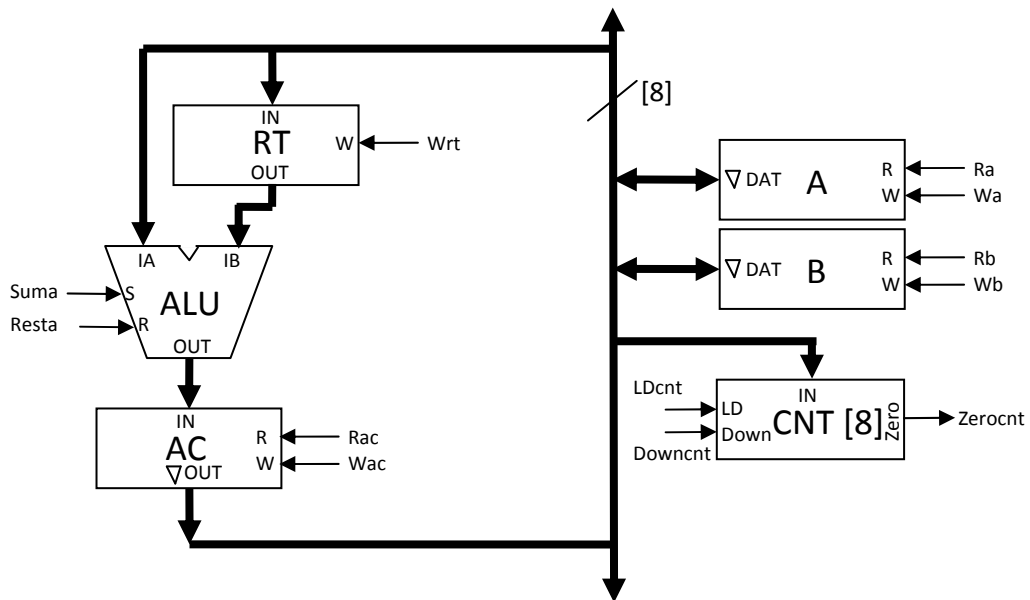
; Aquí estaría la subrutina del enunciado

# ESTRUCTURA DE COMPUTADORES – GII – IC

## Examen Final Convocatoria 2 8-septiembre-2014

Apellidos, Nombre: \_\_\_\_\_

Problema 1. Considere la Unidad de Datos de la figura. Podrá observar que RT tiene salidas incondicionales, que los registros A y B son iguales y tienen I/O bidireccional, que AC es similar a ellos pero con líneas unidireccionales de I y de O, que hay un contador descendente con carga en paralelo y señal de paso por cero, y una ALU que solamente suma (IA+IB) o resta (IA-IB).



Las descripciones Verilog de los módulos tienen las siguientes cabeceras:

- Module regtipo1 (input clk, W, input [7:0] IN, output [7:0] OUT);
- Module regtipo2 (input clk, W, R, input [7:0] IN, output [7:0] OUT);
- Module regtipo3 (input clk, W, R, inout [7:0] DAT);
- Module contador (input clk, LD, DOWN, input [7:0] IN, output ZERO);
- Module uniaritlog (input S, R, input [7:0] IA, IB, output [7:0] OUT);

- a. Describa en Verilog estructural la Unidad de Datos de la figura.
- b. Describa en Verilog el contador descendente mód. 256, CNT.
- c. Se desea realizar la macro-operación  $A \leftarrow A \cdot B$  (producto A·B de números sin signo) mediante sumas sucesivas. Indique la carta ASM de datos para ello (incluyendo Xs y FIN).
- d. Indique la carta ASM de control y diseñe el circuito de control con la técnica de un biestable por estado.

Nombre: \_\_\_\_\_

Problema 2. Se desea introducir en el CS2010 la instrucción

**CPSE Rd, N**

Esa instrucción compara el registro Rd con la constante N y salta (esquiva) la siguiente instrucción si son iguales.

- 2.1. Proponga un código de operación e indique qué formato de instrucción usará.
- 2.2. Indique el código máquina de CPSE R3, 8.
- 2.3. Indique la secuencia de microoperaciones de esta instrucción (especifique tanto las transferencias a nivel RT como las señales que activaría la Unidad de Control). Si necesitara modificar la unidad de datos, explique cómo lo haría, describiéndolo adecuadamente.
- 2.4. Indique qué registros internos se ven afectados al ejecutar CPSE Rd,N. ¿Y los bits de estado?
- 2.5. Encuentre alguna solución que no modifique ningún bit de estado, indicando si ha cambiado la unidad de datos y cómo quedaría la nueva secuencia de microoperaciones.
- 2.6. Si se quisiera introducir también la instrucción equivalente para comparar registros: **CPSE Rd, Rf**, indique el nuevo formato de instrucción y la secuencia de microoperaciones a nivel RT, así como si debe modificar la Unidad de Datos y, en su caso, en qué consistiría.

Nombre: \_\_\_\_\_

3. En un programa en ensamblador de AVR se tiene el siguiente código:

<pre>.include "m328Pdef.inc" .dseg .org \$100 Dato1: .byte 0x100 Dato2: .byte 0x100 Result: .byte 0x100 .cseg .org 0         jmp programa .org \$40 programa: ldi xh, high(Dato1)           ldi xl, low(Dato1)           ldi yh, high(Dato2)           ldi yl, low(Dato2)           ldi zh, high(Result)           ldi zl, low(Result) // Sigue en (*)</pre>	<pre>//      (*)           call exam ;           brge G1b           st z+, r2           st z+, r3 fin:      rjmp fin G1b:     st z+, r0           st z+, r1           rjmp fin // exam:    ld r0, x+           ld r1, x+           ld r2, y+           ld r3, y+           cp r1 ,r3           brne Volver           cp r0, r2 Volver:  ret</pre>
--	---

3.1. Indique los registros internos (R0:R31) y valores de memoria que se modifican al ejecutar el programa completo considerando los datos iniciales siguientes:

- 3.1.1.  $M(\$100)=\$23$ ,  $M(\$101)=\$45$ ,  $M(\$102)=\$67$ ,  $M(\$103)=\$89$ ,...  
 $M(\$200)=\$F0$ ,  $M(\$201)=\$12$ ,  $M(\$202)=\$34$ ,  $M(\$203)=\$56$ ,...  
A partir de  $M(\$2B0)$  todos son 0.

3.1.2. Repita lo anterior si cambia un solo valor a:  $M(\$101)=\$12$ .

3.2. Indique qué operación hace el programa anterior y sobre qué tipo de datos especificando, al menos, su tamaño y si es o no un número (sin/con signo).

3.3. Si hubieran dos vectores de 20 componentes cada uno (cada componente del mismo tipo que el programa original), ubicados a partir de las posiciones Dato1 y Dato2, modifique el código para que se escriba el vector resultado a partir de Result.

# ESTRUCTURA DE COMPUTADORES – GII – IC

## Examen Final Convocatoria 1

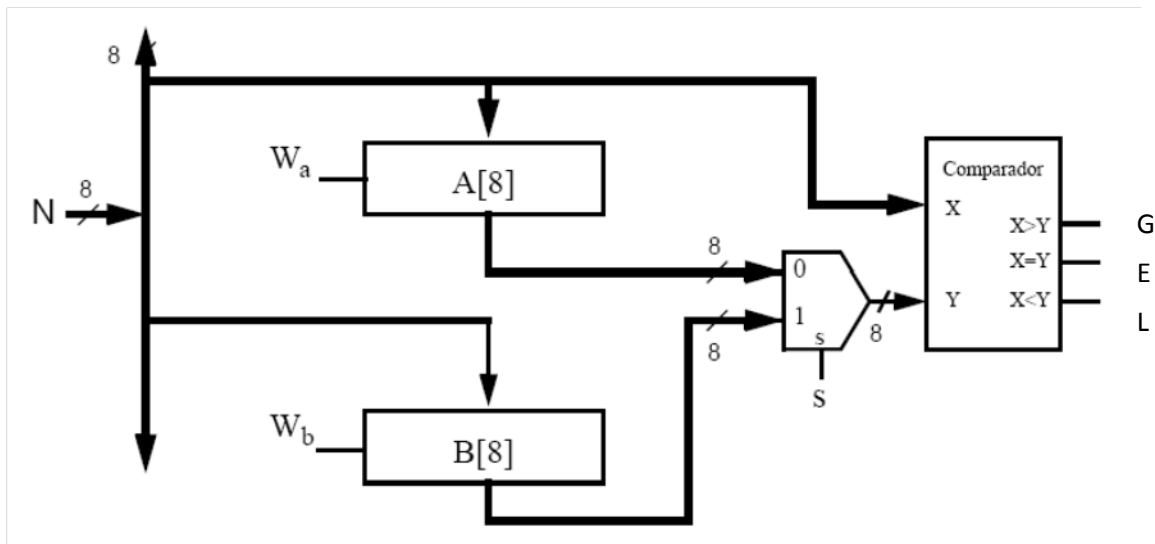
19-junio-2015

Apellidos, Nombre: \_\_\_\_\_

Problema 1. Se desea usar la Unidad de Datos de la figura para almacenar en A el mayor de una serie de números de entrada y en B el menor (números sin signo).

El sistema deberá operar como sigue. Una vez que se active **Xs** y tras un tiempo indefinido, una señal **SerieN** se activará (a 1) cuando se vaya a comenzar el envío de la serie de números. Después irán llegando uno a uno los datos por la entrada **N** transcurriendo un número indeterminado pero suficiente de ciclos de reloj entre dos datos consecutivos. Cada vez que aparece un nuevo dato habrá una señal **NewN** que se pondrá a 1 varios ciclos de reloj. Entonces se procesará el número de la entrada **N** y se actualizarán los registros A y B. La señal **NewN** pasará a 0 y, de nuevo tras un tiempo indefinido, se volverá a poner a 1 cuando haya un nuevo dato que procesar en **N**. El proceso se repetirá mientras la señal **SerieN** sea 1. Cuando **SerieN** sea 0, se activará la señal **FIN** y se volverá al estado inicial a la espera de un nuevo pulso **Xs**.

- Diseñe las cartas ASM de las unidades de datos (UD) y control (UC).
- Diseñe el circuito que implementa la UC utilizando la técnica de un biestable por estado.



Nombre \_\_\_\_\_

Problema 2. Diseñe la subrutina SUB16 para el computador CS2010 que realiza la resta de dos números de 16 bits:  $R1:R0 = R1:R0 - R3:R2$ . **La subrutina debe funcionar para números con y sin signo.**

(NOTA: En el par Rb:Ra se almacena un número de 16 bits, estando el MSB en Rb y el LSB en Ra.)

- a. Comente cómo hacer la operación, indique un organigrama (o similar) que explique su solución y escriba el código ensamblador que la realiza.
- b. Suponga que dos números sin signo se encuentran almacenados a partir de la direcciones \$00 y \$02 respectivamente de la memoria de datos (**byte menos significativo en direcciones pares**). Escriba un trozo de código que reste dichos números llamando a SUB16 y guarde el resultado a partir de la dirección \$04. Asimismo, guarde en la dirección \$06 la constante 1 si hubo error en la resta y 0 en caso contrario.

Nombre \_\_\_\_\_

Problema 3. Realice el sistema de memoria de un computador con un espacio de direcciones lógicas de 64kB ( $2^{16}$  bytes; bits de dirección A15:0). El procesador posee bus de direcciones A15:1 y bus de datos segmentado de 16 bits (D15:0, BE1:0#, donde x# denota "x activa en baja"). Como es habitual, BE1# controla el acceso al segmento del byte alto (D15:8), al cual le corresponde una dirección lógica par, mientras que BE0# controla el acceso al segmento del byte bajo (D7:0), al cual le corresponde una dirección lógica impar.

Se desean instalar los siguientes bloques de memoria física **sin dejar huecos entre los bloques**: 8KB de ROM en la zona más baja de memoria seguidos de 16KB de RAM y a continuación otros 8KB de ROM.

Sabiendo que dispone de chips de 8KB, tanto de ROM como de RAM, con señales CS# y, en su caso, R/W#:

- a. Diseñe el sistema de memoria completo. Debe incluir el mapa, el circuito de selección y todas las interconexiones de los chips de memoria.
- b. Indique (dando las direcciones en hexadecimal):
  - i. qué palabras físicas (chip y dirección) son accedidos cuando el procesador hace un acceso de lectura en los siguientes casos:
    - \* Tamaño B en la dirección \$3456
    - \* Tamaño B en la dirección \$4567
    - \* Tamaño W en la dirección \$789A
    - \* Tamaño B en la dirección \$A987
  - ii. qué dirección lógica y con qué tamaño de acceso (B o W) hay que utilizar para acceder a la palabra física \$21 de **cada uno de los cuatro chips** de memoria utilizados.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 2** **7-septiembre-2015**

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Se quiere diseñar un sistema digital que controle la temperatura de un cultivo celular en un laboratorio, con el objetivo de mantener una temperatura óptima. La temperatura será óptima si está entre dos valores, una máximo (TMax) y otro mínimo (Tmin).

El sistema digital debe operar teniendo en cuenta que:

- Tras activarse la señal de comienzo Xs, el sistema esperará a que una **señal Ctrl** esté activa (en 1) para comenzar a controlar la temperatura. Mientras esa señal Ctrl esté activa efectuará los ciclos de control.
- Las temperaturas máximas y mínimas de referencia se configuran desde el exterior mediante un sistema de 16 *switches* (conmutadores), 8 para TMax y 8 para Tmin. Los valores deberán ser guardados en sendos registros de 8 bits (A para TMax y B para Tmin).
- Un sensor de temperatura codifica en cada instante la temperatura como un dato de 8 bits. Este valor deberá ser almacenado en otro registro T (también de 8 bits).
- La temperatura se evaluará cada 5 minutos y se actuará sobre el sistema bien calentándolo (activando una **señal CAL**), bien enfriándolo (activando una **señal ENF**) o bien no haciendo nada.
- Cuando **Ctrl** se desactive el sistema volverá al estado de no operación.

El sistema dispone de 3 registros de 8 bits (A, B y T), 1 contador (CN) de módulo 8, un temporizador (T5) que activa una **señal Five** cada 5 minutos. Además, para realizar la comparación, dispone de 2 comparadores de 1 bit.

- 1.1. Dibuje como un bloque la Unidad de datos. Describa en Verilog **únicamente** la declaración del módulo correspondiente y de los puertos de entrada y salida.
- 1.2. Describa a nivel RT el registro T.
- 1.3. Dibuje la Unidad de datos, incluyendo todos los componentes y conexiones internas y con el exterior.
- 1.4. Presenta la carta ASM de datos y de control.
- 1.5. Realice la Unidad de control (1 biestable por estado).



Nombre: \_\_\_\_\_

Problema 2. Se desea obtener el punto medio  $(X_m, Y_m)$  de otros dos puntos del plano cartesiano,  $(X_0, Y_0)$  y  $(X_1, Y_1)$ . Todos los valores son **números positivos en complemento a 2 (Ca2)**. Suponga que los valores de partida se encuentran almacenados en  $\$80 (X_0)$ ,  $\$81 (X_1)$ ,  $\$90 (Y_0)$  y  $\$91 (Y_1)$ . Los destinos se darán en  $\$82 (X_m)$  y  $\$92 (Y_m)$ .

- a. Desarrolle un programa basado en el CS2010 que realice esa tarea. El programa deberá poner a 0 los 8 bits menos significativos de la palabra de memoria  $\$A0$  si el resultado es exacto y a 1 si no lo es.
  - i. En primer lugar describa la solución verbalmente o mediante ordinograma.
  - ii. Escriba el programa en lenguaje ensamblador del CS2010.
- b. Considere problemas de desbordamiento (No hay que modificar el programa):
  - i. Justifique si hay problemas de desbordamiento y, en su caso, indique qué habría que hacer para avisar al usuario.
  - ii. Repita "i" si los valores **pueden ser números positivos o negativos**.

Nombre: \_\_\_\_\_

Problema 3. Para el AVR. Se desea obtener la suma de las componentes de un vector que son números sin signo. El vector es de 10 componentes tamaño Byte y se encuentra almacenado a partir de la dirección \$100 de la memoria de datos. Haga un programa que escriba el resultado en la dirección \$200. Considere el posible desbordamiento: en la dirección \$201 debe escribir un 0 si no hay desbordamiento y \$FF cuando lo haya.

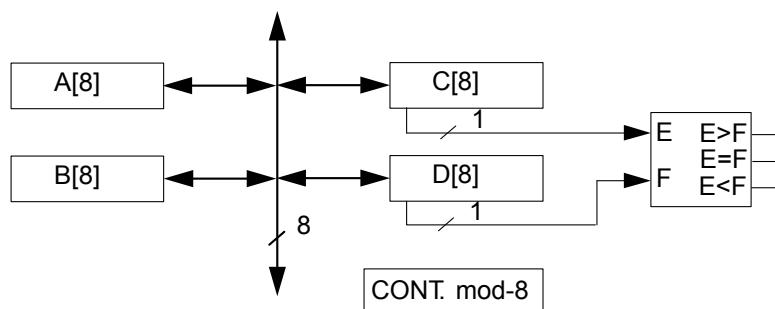
# ESTRUCTURA DE COMPUTADORES – GII – IC

## Examen Final Convocatoria 3

04-diciembre-2015

Apellidos, Nombre: \_\_\_\_\_

- Problema 1. Para la unidad de datos que se presenta, se quiere realizar un sistema digital capaz de comparar dos números de 8 bits (A y B), y almacenar en A el mayor de ellos y en B el menor. Tras finalizar la operación, el sistema generará una señal de FIN.
- Defina correctamente las operaciones de los registros.
  - Obtenga la carta ASM.
  - Sin añadir elementos nuevos, ¿se puede simplificar la arquitectura de esta unidad de proceso? Razone la respuesta.



- Problema 2. Se desea cambiar la semántica de las instrucciones ST y LD del CS2010 para que usen el modo de direccionamiento indirecto con postincremento. Su semántica es idéntica a la del modo indirecto normal salvo que, tras realizarse el acceso a memoria, el registro base se incrementa en una unidad.
- Indique qué cambios habría que realizar a la unidad de datos del sistema para que pudiera implementarse dicho modo (véase figura del CS2010).
  - Descomponga las nuevas instrucciones en microoperaciones.

- Problema 3. La subrutina strwc recibe en X la dirección del primer carácter de una cadena de caracteres (el final de la cadena viene marcado por un byte a 0) y devuelve en el par de registros R16:R17 el número de letras de la cadena y en el par R18:R19 el número de palabras de la misma. Las cadenas sólo contienen letras y espacios. No se tendrá en cuenta el caso de que haya caracteres de control (Escape, Intro, etc.), dígitos y otros caracteres imprimibles. *Ejemplo: "El perro de San Roque ya no tiene rabo" devuelve 9 palabras y 30 letras.*
- Explique con palabras el algoritmo que va a emplear para la subrutina strwc.
  - Escriba la subrutina en ensamblador del AVR.

## ESTRUCTURA DE COMPUTADORES – GII – IC

### Examen Final Convocatoria 1

13-junio-2016

***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Las siguientes cuestiones son independientes. Las respuestas deben ser breves pero razonadas.

- a. Memorias de acceso aleatorio. (Indique cuáles son, para qué se usan y qué tecnologías hay).
- b. Dentro del amplio tema de realización de un sistema electrónico, que abarca desde la concepción hasta el disfrute por el usuario final, ¿para qué sirve Verilog?
- c. Compare los términos: transferencia entre registros, estado, micro-operación, macro-operación e instrucción.
- d. Suponga que en el CS2010 se van a introducir dos nuevas instrucciones, una de 2 palabras de código máquina (con el código de operación \$0E) y otra de 3 palabras (con el código de operación \$1E). Explique qué modificaciones hay que realizar.
- e. Si en la posición \$89AB de la memoria de programa del AVR hay una instrucción “CALL etiqueta” y la “etiqueta” está en la posición \$88AB”, indique qué registros y palabras de memoria cambian al ejecutarla y qué valores finales quedan en los elementos afectados. Repita si la instrucción que hay es “RCALL etiqueta”.

Nombre \_\_\_\_\_

Problema 2. Se dispone de dos registros de **12 bits**, A y B, ya cargados con sendos datos, que son números con signo en notación Ca2. Se desea realizar un sistema digital que permita realizar una de dos posibles operaciones:

1/**Ordenar**: con ella se escribirá el mayor de estos números en un registro MAY y el menor en otro registro MEN;

2/**Sumar**: la suma se guardará en un tercer registro SUM (resultado también en notación Ca2). En caso de desbordamiento deberá activar una salida DESBORDA.

- a. Especifique adecuadamente una Unidad de Datos **específica** para este sistema digital. (“**Específica**” quiere decir que tendrá los componentes y conexiones necesarias para resolver estas tareas –Ordenar y Sumar- y no otras genéricas).
- b. Escriba la carta ASM de datos y de control.
- c. Diseñe la Unidad de Control mediante 1 biestable por estado.

Nombre \_\_\_\_\_

Problema 3. La siguiente subrutina está escrita en **ensamblador del CS2010**.

Analícela y determine su función. En particular:

- 1) Analice cómo van cambiando los datos si inicialmente R0=1010 0000.
- 2) Determine la operación global que hace este programa al ejecutarse. (Indíquela con palabras, ya que hace una función reconocible).

```
-----  
; QueSera v1.0  
;  
; Entradas: R0 dato  
; Salidas: R0 dato procesado  
; No se conserva el valor inicial de R1  
-----
```

```
QueSera: CLC  
          LDI R1, 1  
  
Buc:     ROR R0  
          ROL R1  
          BRCS QueFin  
          JMP Buc  
  
QueFin:  MOV R0, R1  
          RET
```

## ESTRUCTURA DE COMPUTADORES – GII – IC

### Examen Final Convocatoria 2

6-septiembre-2016

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Considere un registro (RIC) de “n” bits con entradas (Xin) y salidas (Zout) separadas que debe poseer puesta a 0 (asíncrona, CLA), carga en paralelo (síncrona, W), cuenta creciente (síncrona, Up) y lectura condicional (R). Su operación debe ser lo menos restrictiva posible.

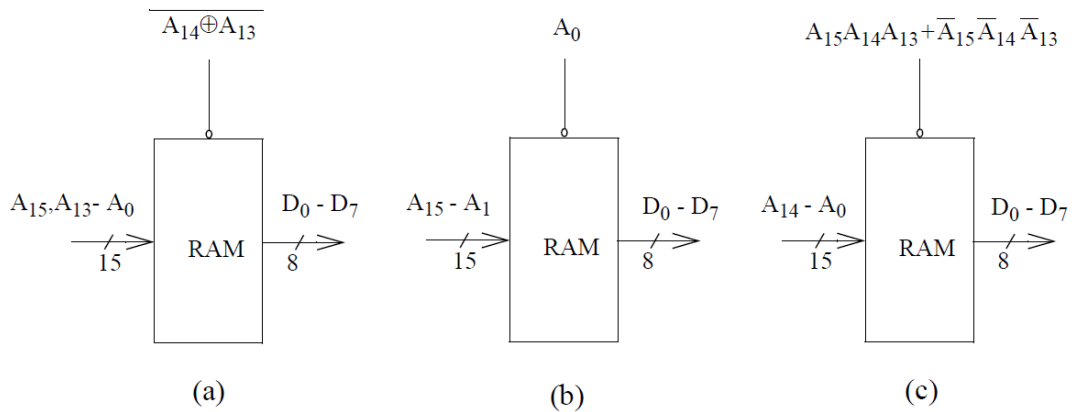
- Haga la descripción completa de RIC a nivel RT.
- Describa RIC en Verilog.
- Discuta si es posible realizar una memoria LIFO de fondo 8 usando registros RIC (y componentes combinacionales).
- Se propone usar RIC en la implementación de CS2010. Justifique brevemente si puede usarse o no como:
  - IR
  - AC
  - SP
  - PC
- Complete la siguiente secuencia en el tiempo (| indica el flanco activo).

Ciclo de reloj:	1		2		3		4		5	
CLA:	1		0							
Xin:	5		12				7		2	
W:	1		1			0		1		0
Up:	0		0			1		0		
R:	1						0			1

Nombre \_\_\_\_\_

Problema 2. En el mapa de memoria de un microcomputador de 16 líneas de dirección (A15:A0) se han ubicado dos memorias, una RAM de 8K en las primeras 8K posiciones de memoria y una ROM de 8K en las últimas 8K posiciones de memoria.

Se desea incluir otra memoria, una RAM de 32K, para lo que se han propuesto los 3 diseños de la figura. Indique en qué medida es posible cada uno de los diseños y determine qué palabra de la RAM se direcciona cuando A15:A0= \$ABCD (hexadecimal). ¿Qué dirección hay que poner en el BUS de direcciones para leer la posición \$4680 de la RAM (en cada caso posible)?





Nombre \_\_\_\_\_

Problema 3. Realice en ensamblador del AVR la subrutina **strcomp** que compara dos cadenas de caracteres tipo C (ver Nota 1), la primera comienza donde apunta X y la segunda donde apunta Y, y devuelve un número en R31 que será:

- 0 si ambas cadenas son iguales.
- Un número positivo si la primera cadena es mayor que la segunda (ver Nota 2).
- Un número negativo en el resto de los casos.

Parámetros de entrada: X apunta a la primera cadena e Y a la segunda.

Valor de retorno (salida): en R31 se devuelve el número que indica la relación entre ambas cadenas.

- a. Haga un diagrama de flujo o explique con palabras el algoritmo que usará para resolver el problema.
- b. Escriba la subrutina en ensamblador del AVR.

**NOTAS:**

1. Una cadena de caracteres tipo C es un vector en el que cada carácter ocupa un byte y se codifica en ASCII. El final de la cadena viene marcado por un carácter NULL cuyo código es el 0. Ejemplo: "Hola" se almacena como 'H', 'o', 'l', 'a', 0, donde, p.ej., 'a' significa código ASCII de la a. En este caso concreto, 'a'=\$61=97.

2. Una cadena es mayor que otra cuando alfabéticamente va después. Ejemplo: La cadena "Estructura de computadores" es mayor que la cadena "Circuitos electrónicos digitales".

Casos especiales a tener en cuenta:

- Se considerará que alfabéticamente las mayúsculas van antes que las minúsculas.
- Cuando una de las cadenas coincide con el principio de la otra, pero es más corta, la cadena corta es la menor.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 1** **05-diciembre-2016**

***Apellidos, Nombre:*** \_\_\_\_\_

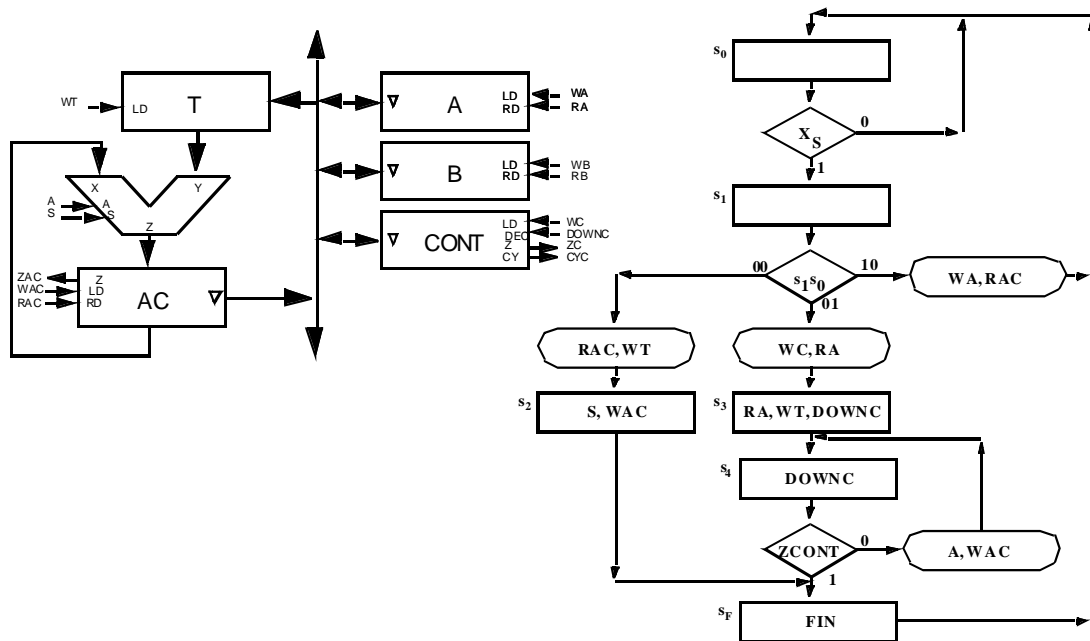
Problema 1. Las siguientes cuestiones son independientes. Las respuestas deben ser breves pero razonadas.

- a. Explique el concepto de “jerarquía de memoria” y exponga las razones que conducen a jerarquizar la unidad de memoria en computadores.
- b. Explique qué es la lista de sensibilidad en Verilog, cuándo se usa y qué puede contener.
- c. En el CS2010, escriba las micro-operaciones de búsqueda y de ejecución de las dos instrucciones siguientes. Obtenga el código máquina (resultado en hexadecimal) de ambas instrucciones si  $Rd=R2$  y  $dato=\$CA$ :
  - i. LDI  $Rd,dato$
  - ii. CPI  $Rd,dato$
- d. Un procesador tiene un espacio de 16MB,  $DB[16]$  e instrucciones de tamaño B (par e impar) y de tamaño W. Suponga que posee una memoria M3 de 2 MB situada en los Bytes impares y que se selecciona cuando  $A23=1$  y  $A22=0$ . Si se quiere borrar la palabra  $M3(\$124)$  mediante la instrucción **CLR.B dir**, obtenga el valor hexadecimal de “dir”.

Nombre \_\_\_\_\_

Problema 2. Las siguientes figuras representan una unidad de datos y la carta ASM de control de un sistema digital.

- Diseñe la unidad de control para la carta ASM de la figura, utilizando la técnica de diseño basada en un biestable por estado.
- Describa las distintas macrooperaciones que realiza la carta ASM.
- Describa formalmente (tabla a nivel RT) los siguientes componentes: CONT, AC, T y A. Interprete las señales según el nombre utilizado.
- Realice la carta ASM de datos equivalente a la carta ASM de control de la figura.



Problema 3. Se desea incorporar al CS2010 una nueva instrucción de movimientos de datos cuya sintaxis es **MPI (Rd)+,(Rf)+** donde MPI es el acrónimo de Move Post-Increment y se ha seguido la sintaxis habitualmente utilizada en mnemónico.

- a. Especifique formalmente (use el lenguaje RT) **todas las modificaciones de datos** que causa la ejecución de esta instrucción. (Añada una breve descripción en lenguaje natural para dejar claro cuál es la operación de esta instrucción).
- b. Explique si para ejecutar esta instrucción hay que modificar o no la Unidad de Datos del CS2010; en su caso, explique con rigor qué cambios hace.
- c. Indique las micro-operaciones necesarias para implementar **MPI (Rd)+,(Rf)+**. Debe dar tanto las transferencias de datos como las señales de control.
- d. Suponga, por último, que existe un lenguaje de alto nivel con la instrucción

**MVector DirDest, DirFue, NumCom**

donde **MVector** alude a mover un vector, **DirDest** es la dirección de destino de la primera componente del vector, **DirFue** es dirección de origen (fuente) de esa primera componente del vector y **NumCom** es el número de elementos (o componentes) del vector. Las componentes del vector ocupan posiciones crecientes consecutivas. Haga una subrutina en el CS2010 que interprete esta instrucción de alto nivel. (considere el conjunto de instrucciones del CS2010 ampliado con **MPI (Rd)+,(Rf)+**).

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 1** **08-junio-2017**

***Apellidos, Nombre:*** \_\_\_\_\_

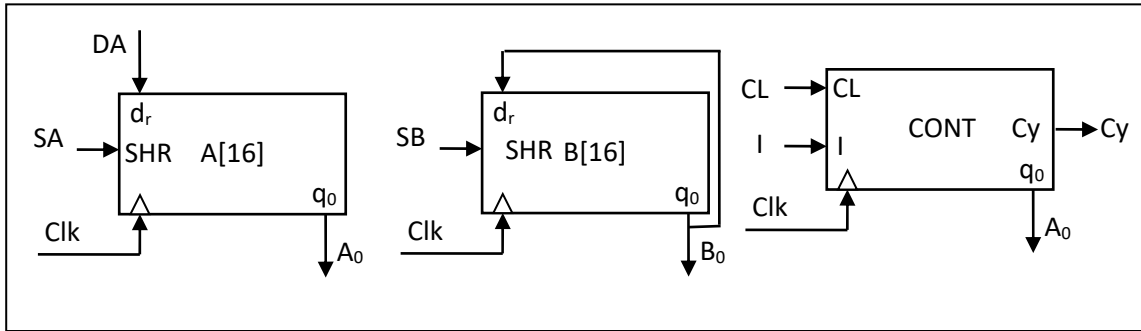
Problema 1. Considere el CS2010 (tanto la Unidad de Datos como la de Control). Debe discutir los cambios que implicarían las siguientes modificaciones sobre la arquitectura inicial. (Cada modificación es independiente de los demás).

- a. La anchura de datos pasa a ser de 16 bits (esto es, tamaño W).
- b. La anchura de las instrucciones (código máquina) pasa a ser de 20 bits.
- c. La memoria de datos pasa a tener el cuádruple de datos.
- d. La memoria de código pasa a tener un bit de dirección más.

Nombre \_\_\_\_\_

Problema 2. La Unidad de Datos de un sistema digital es la mostrada en la figura (está completa, no falta ninguna componente). Supuesto que A y B ya contienen sendos números (sin signo), el sistema debe ejecutar la operación  $A \leftarrow A + B$  (suma algebraica de A y B). Ignore el desbordamiento final en la suma.

- Dimensione adecuadamente el contador CONT y descríballo a nivel RT.
- Diseñe la carta ASM de datos y de control.
- Realice una implementación de la Unidad de Control usando la técnica de un biestable por estado.



Nombre \_\_\_\_\_

Problema 3. Se desea realizar en ensamblador del ATmega328P el siguiente programa. Existen ya escritos en memoria un número determinado de 8 bits, que se usará como referencia, en la posición "refer" y un vector con un número de componentes "ncomp" (con  $ncomp < 500$ ) situado en posiciones crecientes de memoria a partir de la dirección "dirv".

El programa deberá obtener el número de componentes del vector que son mayores o iguales que el número de referencia, escribiendo dicho valor a partir de la dirección \$100 de la memoria.

- a. Defina mediante las directivas adecuadas el valor del número de componentes  $ncomp=432$  ( $432 = 0x1B0$ ) y sitúe los datos en memoria de la siguiente forma: a partir de la dirección \$200 está escrito primero el número de referencia e inmediatamente después todas las componentes del vector.
- b. Escriba el código ensamblador para realizar la tarea perseguida si todos los números son sin signo.
- c. Modifique su programa si los números son con signo (en Ca2). (Indique claramente qué instrucciones debe cambiar).

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final                                      Convocatoria 2            05-septiembre-2017**

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1.    Responda a cada una de las preguntas siguientes:

- a. En el contexto de Verilog:
  - i. Explique qué significa que las sentencias se ejecutan concurrentemente y qué es lo opuesto.
  - ii. Indique qué es, para qué sirve y en qué lenguaje se escribe un fichero de testbench.
  - iii. Escriba la palabra binaria y el valor decimal de los siguientes números verilog:
    1. 9’h15
    2. 9’h815
    3. 9’15
    4. ‘h15
- b. Describa las micro-operaciones de búsqueda y de ejecución de las siguientes instrucciones del CS2010:
  - i. BRLT dir
  - ii. ROL Rd
- c. *Lenguaje ensamblador:* ¿Qué es? Compare el del CS2010 y el del ATmega328P.
- d. En el contexto del AVR considere que R27 = R29 = R31 = \$4 y que R26=\$12, R28=\$34, R30=\$56 y que la memoria de datos contiene  $M(\$...H_1H_0) = \$H_0H_1$  siendo  $H_1$  y  $H_0$  dígitos hexadecimales.  
Indique todas las instrucciones de carga al registro R19 que posean direccionamiento indirecto con registro base Y, presentando en todos los casos el valor final de R19 y del registro base Y.



Nombre \_\_\_\_\_

Problema 2. Dispone de chips de ROM y de RAM de diverso tamaño, decodificadores y puertas. Se desea hacer un sistema de memoria de un procesador con bus de direcciones de 16 bits y bus de datos de 16 bits segmentado en sus dos Bytes (A15:1, D15:0, BE1:0#, donde # significa activa en baja) que dispone de 24KB de ROM en las direcciones más bajas de memoria y, justo a continuación, 16 KB de RAM (no hay separación entre los dos bloques).

- a. Dibuje el mapa de memoria, especificando claramente las direcciones de cada bloque. Obtenga las ecuaciones de selección de los chips involucrados en el diseño para realizar una decodificación completa y suponiendo que minimiza el número de chips de memoria.
- b. Considere ahora que no se esperan futuras expansiones en el sistema de memoria. Diseñe el nuevo sistema buscando una lógica de selección optimizada (la más simple).
- c. Dibuje el circuito del sistema de memoria de este último caso.

Nombre \_\_\_\_\_

Problema 3. Se desea construir un sistema digital que añada un bit de paridad par a un dato de "N" bits que **se recibe en modo serie**. El dato con paridad par **se mostrará en paralelo** en la salida.

En particular, el sistema tiene dos entradas de datos, "Env" y "Din" y dos salidas de datos, "ParEnv" y "Dout". Además, tiene las habituales "Xs" y "Fin". Tras comenzar, el sistema esperará a que "Env" sea 1. En el mismo ciclo en que Env=1 el sistema empieza a recibir por "Din" un dato en serie, un bit cada ciclo de reloj, comenzando por el LSB. El envío del dato terminará cuando "Env" sea 0. El sistema deberá enviar **en paralelo** por "Dout" el mismo dato añadiendo un bit de paridad par como MSB. Para avisar de que el dato "Dout" ya es válido, el sistema pondrá "ParEnv" a 1 durante un ciclo de reloj y terminará su operación.

- a. Para N=16, dé una Unidad de Datos específica así como la carta ASM de datos y control para resolver este problema. No olvide dimensionar adecuadamente todas las componentes y señales.
- b. Realice la Unidad de control mediante la técnica de un biestable por estado.
- c. Diga qué cambios hay que realizar en el sistema si N=128.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 1** **30-noviembre-2017**

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1.

1) Se ha organizado una pila basada en puntero (SP, Stack Pointer) en las últimas **direcciones de una RAM**. La pila va desde  $1...111_2$  hacia direcciones decrecientes:  $1...110_2$ , etc. En estado ocioso, SP apunta a la primera palabra vacía. Suponga que en este momento ya se han hecho 7 operaciones de escritura en pila. En los próximos ciclos se hace: 1 PUSH (dato: \$BC), 1 NOP, 2PUSH (datos: \$35 y \$21) y 1 PULL. Escriba la secuencia de contenidos de SP[16 bits] y de la memoria afectada.

2) En el contexto de Verilog, escriba el valor binario y decimal de la variable VAR, de 8 bits, para cada caso que sigue:

VAR = 8'o36 = \_\_\_\_\_ = \_\_\_\_\_  
VAR = 8'h36 = \_\_\_\_\_ = \_\_\_\_\_  
VAR = 12'hcb0 = \_\_\_\_\_ = \_\_\_\_\_  
VAR = 8'hc = \_\_\_\_\_ = \_\_\_\_\_  
VAR = 8'shc = \_\_\_\_\_ = \_\_\_\_\_

3) Interprete el siguiente código Verilog y describa el módulo a nivel RT:

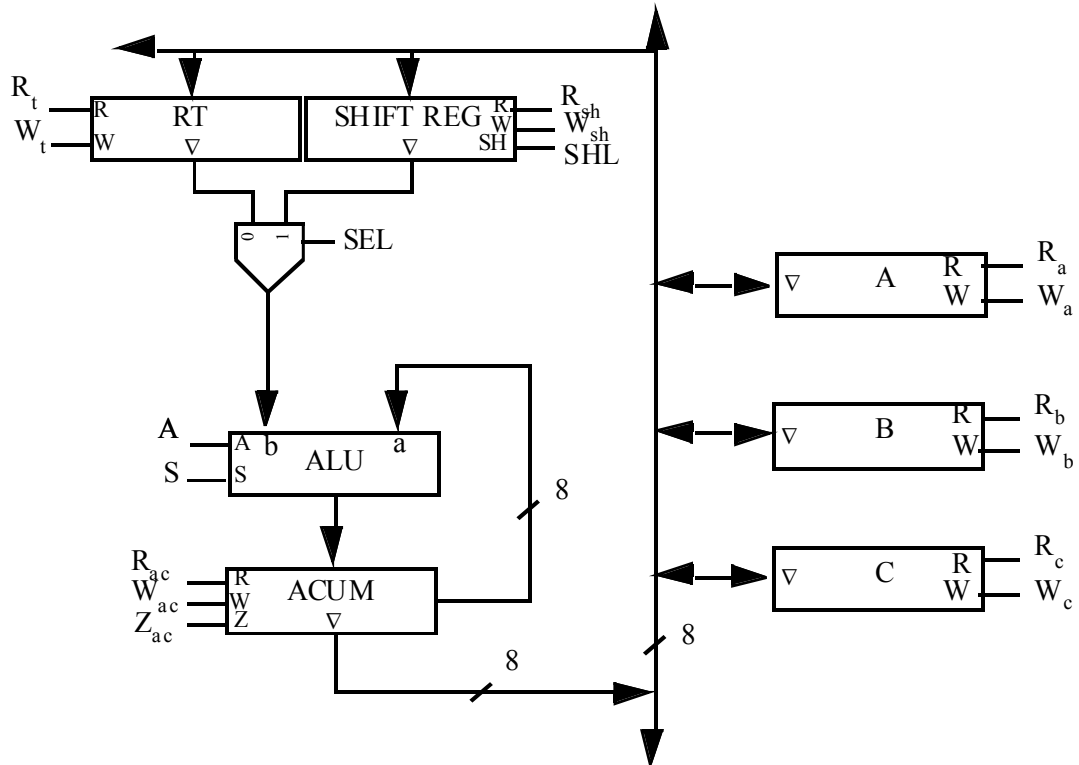
```
module caso2c #(parameter NN=32)
  (input ck, cl, ld, rd, input [NN-1 : 0] xin, output [NN-1 : 0] zout);
  reg [NN-1 : 0] est;
  always @ (posedge ck, negedge cl)
    if (cl==0) est <= 0;
    else if (ld) est <= xin;
  assign zout = rd? est : 'hZ;
endmodule
```

4) En el contexto del AVR: Qué diferencias encuentra entre las distintas opciones de instrucción de salto a subrutina: CALL, RCALL e ICALL. Indique, en cada caso, cómo se modifican los registros y datos de memorias que intervienen en cada instrucción.

5) En el contexto del AVR: Qué diferencias hay entre instrucción de salto incondicional y de salto “tipo Skip”. Escriba un ejemplo de instrucción para cada uno de ellos y explique sus argumentos y efectos sobre una sección genérica de programa.

Nombre \_\_\_\_\_

Problema 2. Considere la unidad de datos de la figura:



- Describa en lenguaje RT el registro ACUM.
- Obtenga las cartas ASM de datos y de control de forma que, en función de un bit de entrada de selección ( $l_0$ ), pueda elegirse entre una de las dos macrooperaciones siguientes:
  - $A \leftarrow B + 2 C$
  - $C \leftarrow A - 2 B$
- Diseñe la unidad de control con la técnica de un biestable por estado.

**NOTA:** ALU suma cuando  $A=1$  y resta cuando  $S=1$ .

Nombre \_\_\_\_\_

Problema 3. Para el siguiente programa en el ámbito del computador CS2010:

- Escriba el código máquina (en hexadecimal) del segmento de código que va desde la etiqueta "Atras" hasta la etiqueta "Etiq", ambas incluidas.
- Explique justificadamente qué hace el programa. (Como solución no se le pide que indique lo que hace cada instrucción individualmente, sino el programa completo).

	LDI R5, 10
	LDI R4,0
	LDS R0, \$A0
	LDI R1, \$B0
Siguiente:	LD R2, (R1)
	SUB R2,R0
	BRCS Etiq
	ADDI R4,1
Atras:	SUBI R5,1
	CPI R5,0
	BREQ Etiq2
	ADDI R1,1
	JMP Siguiente
Etiq:	JMP Atras
Etiq2:	STS \$C0, R4
	STOP

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 1**

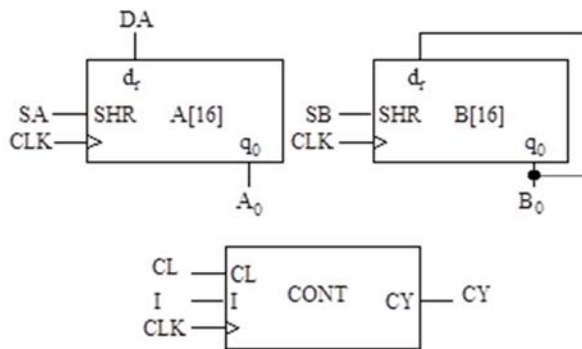
**26-junio-2018**

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- a. Un procesador (AB[16], DB[8], R/W') tiene un mapa de memoria simple. Se desea ubicar una **RAM 16kB** a partir de la dirección **\$6000**. Dibuje todas las conexiones entre el procesador y las líneas de la RAM. Indique en hexadecimal las direcciones lógicas correspondientes a RAM(0111...11) y a RAM(100...00).
- b. Explique cómo es la sintaxis de la sentencia "case" sobre los valores de dos **variables escalares, x e y**. Ponga como ejemplo el decodificador 2:4 (**decodificando "x y"**).
- c. Considere la calculadora de 8 registros que tiene en la hoja de material para exámenes. Realice las operaciones siguientes bien en una o bien en varias micro-operaciones. (Siempre tiene que cumplirse el principio de afectar solamente al contenido de los registros visibles implicados en la operación/instrucción).
  1.  $R1 \leftarrow 0$
  2.  $R1 \leftarrow R2 + R3$
- d. Resuma el modo de operación de la pila en el CS2010 mencionando en qué instrucciones se hace uso de ella.
- e. Explique qué significan los bits de estado S y N en el ATmega328P. Ponga ejemplos de datos A y B que clarifiquen el significado de ambos bits (S y N) en la suma  $A + B$ ; lo mismo para la resta  $A - B$ .

Problema 2. Para la unidad de datos de la figura:



Restador completo

Ci	A	B	Co	R
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

- Diseñe las cartas ASM de las Unidades de Datos y de Control de un sistema que permita realizar la operación  $A \leftarrow A - B$ , donde '-' es la operación resta algebraica (vea la tabla de verdad del restador completo arriba). Indique el módulo de CONT. El valor de B no debe resultar alterado al final del proceso. Ignore el posible bit de acarreo final.
- Realice una implementación de la Unidad de Control usando la técnica de un biestable por estado.

Problema 3. Considere el CS2010. Se pretende realizar la rotación a la izquierda de un registro interno Rd un número N veces, donde el número N se encuentra almacenado en el registro Rf. (La rotación en CS2010 incluye al bit de estado C). Se debe realizar de dos formas diferentes:

- a. Por software, usando el juego de instrucciones del CS2010.  
Escriba la subrutina correspondiente.
- b. Por hardware, incorporando al CS2010 una nueva instrucción “**NROL Rd,Rf**” que efectúe la tarea requerida.
  - b1. Indique si modifica o no la Unidad de Datos y, en caso afirmativo, especifique claramente cómo es la modificación incluyendo en todo caso el dibujo de la parte nueva de la Unidad de Datos.
  - b2. Proporcione un formato de instrucción adecuado.
  - b3. Muestre la carta ASM **de datos y de control** correspondiente a la ejecución de esa instrucción. (Alternativamente a la carta ASM puede mostrar la secuencia de microoperaciones).
- c. Razone si se puede sustituir alguna instrucción del CS2010 por esta nueva sin que pierda funcionalidad.



**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 2** **04-septiembre-2018**

***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Responda a las siguientes cuestiones:

- a. En una organización de memoria existe un chip de 4kB conectado a un procesador de AB[16], y DB[8]. Se ha comprobado que la dirección lógica  $A_{15:0} = \$0123$  accede a la dirección interna  $\$123$ , que la dirección lógica  $A_{15:0} = \$FCBA$  accede a la dirección interna  $\$CBA$  y que la dirección lógica  $A_{15:0} = \$1321$  accede a la dirección interna  $\$321$ . Indique todas las conexiones de ese chip de memoria y su situación completa en el mapa de memoria correspondiente.

Para cada una de las siguientes **instrucciones del AVR**, muestre los contenidos de todos los registros implicados tras su ejecución. Para ello, suponga que la instrucción se encuentra almacenada a partir de la dirección  $\$90$  de la memoria, que inicialmente los registros contienen  $R25=\$43$  y  $R24=\$C0$ , que “etiqueta” apunta a la dirección  $\$120$  y que, en la memoria de datos, la zona de pila está ocupada hasta la dirección  $\$2E0$  inclusive.

(Cada instrucción se responde por separado).

- b. CALL etiqueta
- c. SBIW R24, 0b011010

Problema 2. Un sistema digital tiene un único bit **Xin** como dato de entrada (además, posee la señal de comienzo **Xstart**). Por él llegan datos con el siguiente protocolo: durante varios ciclos de reloj  $X_{in}=0$ ; una vez que  $X_{in}=1$ , por esa misma línea se transmiten bit a bit los 15 bits de un cierto dato comenzando por su LSB. Tras acabar el último de los bits del dato,  $X_{in}$  vuelve a hacerse 0, valor donde permanecerá varios ciclos. En concreto, sería como:

$X_{in}:\dots 0, 0, 0, 1, D_0, D_1, D_2, \dots, D_{14}, 0, 0, 0, \dots$

El sistema deberá añadir un bit de paridad par al dato y situarlo en su posición MSB. Cuando el dato con paridad esté completo avisará al exterior activando durante un ciclo de reloj una señal **Valid** que indicará que el dato con paridad es válido y, así, podrá ser leído en paralelo mediante un bus de salida **XOUT**. Después esperará a recibir un nuevo dato. Si durante 1024 ciclos de reloj no se inicia la llegada de un nuevo dato serie, el sistema digital acabará su operación (activando **FIN**).

Diseñe el sistema digital.

(Deberá dar una Unidad de Datos específica, la carta ASM de datos y de control y el diseño de la Unidad de Control, todo ello adecuadamente descrito).

Problema 3. Considere el CS2010.:

- a. Se desea añadir la instrucción **CALL (Rf)**. Dicha instrucción realiza un salto a la subrutina cuya dirección de memoria de programa es la contenida en el registro Rf. Indique:
  - i. Formato de instrucción usará.
  - ii. Conjunto de microoperaciones.
- b. Escriba **la subrutina NEG** que calcula el opuesto (en complemento a 2) de R0 y lo deja en R0.
- c. Llame a la subrutina anterior para escribir **una nueva subrutina llamada ABS** que calcula el valor absoluto de R0 y lo deja en R0.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 1** **04-diciembre-2018**

***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Responda a las siguientes cuestiones teóricas:

- a. Memorias: compare las memorias EEPROM y la SRAM
- b. Verilog: explique las diferencias entre asignamientos bloqueantes y no bloqueantes. Ayúdese de un ejemplo.
- c. RT: explique las similitudes y las diferencias entre la calculadora de 8 registros estudiada y el computador simple 1 tanto a nivel de Unidad de Datos como de Control.
- d. CS: describa 6 modos de direccionamiento que están disponibles en alguno de los computadores estudiados (CS1, CS2, CS2010, AVR).

Problema 2. Un sistema digital recibe 1024 datos por un bus de entrada de 16 bits, XIN, y se quiere averiguar cuántos de ellos son menores que un cierto valor umbral, UMB. Todos los datos son números sin signo. El sistema mostrará el resultado de la cuenta por sus salidas OUT.

El sistema operará de la siguiente forma: Tras recibir la señal de comienzo Xs, el sistema capturará cada dato cuando se active una señal externa DVAL (Dato Válido). El primer dato que recibe es el valor umbral UMB. A continuación irá recibiendo los 1024 datos, cada uno tras su correspondiente validación con DVAL.

- a. Dibuje un bloque que muestre todos los terminales del sistema. Tiene que dar el tamaño adecuado de cada terminal.
- b. Proponga una **Unidad de Datos específica** para este sistema. Describa a nivel RT los componentes que utilice y dimensione los buses de conexión.
- c. Dé la carta ASM de Datos y de Control.
- d. Diseñe la Unidad de Control basada en un biestable por estado.

Problema 3. Sean  $N = N_7...N_1N_0$  y  $R = R_7...R_1R_0$  un número y un registro de 8 bits, respectivamente. El MSB de  $N$  es  $N_7$  y el LSB es  $N_0$ . Por su parte,  $R_7$  es la posición más a la izquierda de  $R$  y  $R_0$  es la que está más a la derecha. El almacenamiento por defecto de  $N$  en  $R$  se hace de la forma directa ( $R = N_7...N_1N_0$ ), mientras que la forma *bit reverse* se hace al revés,  $R = N_0...N_6N_7$ .

Para el conjunto de instrucciones del ATmega328:

- a. Desarrolle la subrutina "revertirB" que ejecute  $R0 \leftarrow \text{reverse}(R0)$  donde la operación reverse cambia la representación bit directa a bit reverse o viceversa
  - i. Asuma que puede modificar el contenido de otros registros internos, indicando cuáles.
  - ii. Proponga otra solución para conservar los valores de todos los registros internos (obviamente, salvo R0).
- b. Suponga que hay un vector V1 de dimensión DIM ( $<255$ ) almacenado en la memoria de datos desde la dirección DIR1 y siguientes. Debe obtener un programa que almacene en otro vector, V2, el "reverse" de cada Byte de V1. El vector V2 debe almacenarse a partir de la dirección DIR2.
  - i. Escriba **mediante directivas** la parte correspondiente en ensamblador para dar los valores DIM=75=\$4B, Vdir=\$300 y Wdir=\$400.
  - ii. Escriba el programa para obtener  $V2(k) \leftarrow \text{reverse}(V1(k))$  (para todo k).

## ESTRUCTURA DE COMPUTADORES – GII – IC

### Examen Final Convocatoria 1

24-junio-2019

***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- a. Explique la arquitectura interna de un dispositivo de memoria tipo RAM de 4kB.
- b. Indique qué es una lista de sensibilidad en Verilog y cómo se define para un subsistema con señal de reloj CLK activa en baja y señal de CLEAR asíncrona activa en alta.
- c. Añada un “Status Register (SR)” a la calculadora de la Hoja de Examen con los bits C (carry/borrow), V (overflow) y Z (Zero). Redefina la ALU e indique los cambios que necesite en la calculadora.
- d. En el CS2010, escriba las micro-operaciones para ejecutar las instrucciones **LD Rd, (Rb)** y **LDS Rd, dir**. Debe dar tanto los *datos* como las *señales de control* que se activan.
- e. Explique qué significan en AVR los bits de estado S y N. Ponga dos ejemplos de datos A y B que clarifiquen el significado de ambos bits.

Problema 2. En el contexto del CS2010:

- a. Considere la siguiente secuencia de micro-operaciones de control para una instrucción con formato de tipo A/B (la 1 es la primera micro-operación y n es la última):

1	$W_{MAR}, R_{SP}, W_{AC}, OP_2, OP_1$
2	$W_{MDR}, R_{AC}, D_{SP}$
3	$W_{MEM}$
<i>Sigue una secuencia de micro-operaciones indeterminadas</i>	
n-3	$I_{SP}$
n-2	$W_{MAR}, R_{SP}$
n-1	$W_{MDR}, I/O^*_{MDR}, R_{MEM}$
n	$W_{REG}, I/O^*_{MDR}$

- a1. Obtenga las micro-operaciones a nivel RT de datos.  
a2. Interprete para qué sirven las micro-operaciones especificadas anteriormente.
- b. Incluya una nueva micro-operación, **MOVM (Rd), (Rf)**, la cual realiza la operación
- $$M(Rd) \leftarrow M(Rf)$$
- donde Rd y Rf pueden ser cualesquiera de los registros del CS2010.
- b1. Indique el formato usado y el código máquina para el caso concreto **MOVM (R3), (R2)**  
b2. Desarrolle las micro-operaciones de ejecución de la nueva instrucción, explicando si necesita cambiar la Unidad de datos y, en su caso, detallando ese cambio. Debe dar tanto el flujo de datos como las señales de control a activar.  
b3. Muestre el valor final de todos los registros del CS2010 y de la memoria de datos afectados tras ejecutar **MOVM (R3), (R2)**.



Problema 3. Realice la subrutina *strchr* que busca un carácter en una cadena de caracteres. Cada carácter es ASCII y ocupa 1Byte. En X se encuentra la dirección del comienzo de la cadena y en R28 el carácter a buscar.

La subrutina devuelve en Z la dirección de la primera ocurrencia y en R29 el número de ocurrencias. Si no se encontró, devolverá un 0 en Z.

NOTA: Una cadena de caracteres es un vector de caracteres codificados en ASCII que **termina en el código NUL (0)**.

- a. Explique el algoritmo que usará para resolver el problema (organigrama, con palabras,...).
- b. Escriba la subrutina en ensamblador del AVR.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 2** **02-septiembre-2019**

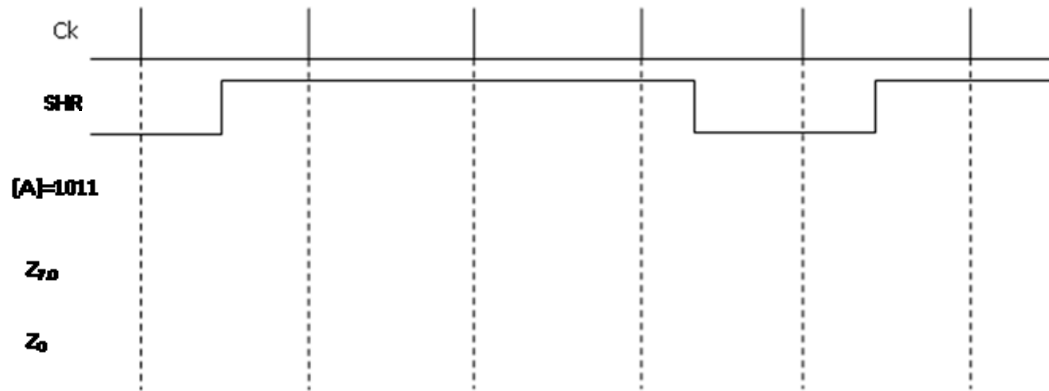
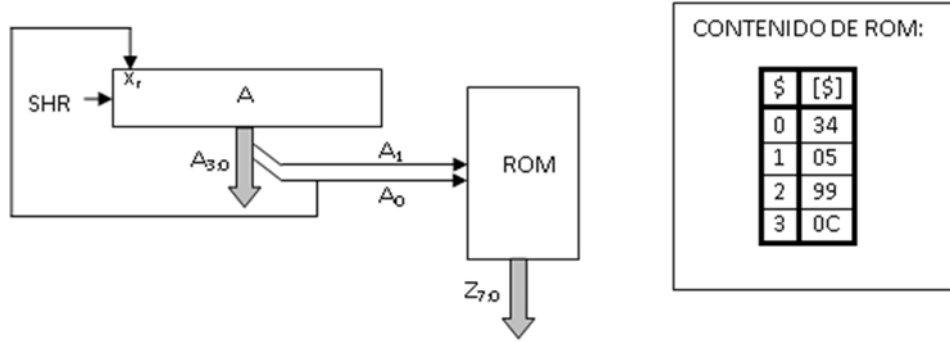
***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- a. Ampliación de memorias en anchura.
- b. Describa en Verilog un contador ascendente módulo 16 disparado por flanco negativo con entrada de inhibición y salida de *carry*.
- c. Arquitecturas de referencia para computadores: descripción y comparación.
- d. Modos de direccionamiento del CS2010.

Problema 2. Considere el circuito de la figura:

- a. Si el registro A inicialmente contiene el dato "1011", complete el cronograma dando A, Z<sub>7:0</sub> en hexadecimal y dibujando la forma de onda de la salida Z<sub>0</sub>. (Responda en la propia figura).
- b. Describa en *verilog* el circuito de la figura:
  - b1. Describa el registro A.
  - b2. Describa la memoria ROM.
  - b3. Describa la Unidad de Datos completa.



Problema 3. Realice en el ensamblador de AVR:

- a. Una subrutina que permita obtener en R0 el cociente y en R1 el resto de la división de R16 (dividendo) entre R17 (divisor). Esto es,  $R0 \leftarrow R16/R17$  y  $R1 \leftarrow R16 \% R17$
- b. Un programa que divida cada componente de un vector V entre la de un vector W, dando los vectores cociente COC y resto REST. El número de componentes será el indicado por la variable NCOMP, que será siempre no nula y menor o igual que 256. El vector V está almacenado en la SRAM a partir de la posición Vdir, el vector W a partir de la posición Wdir, el vector COC a partir de COCdir y el vector REST a partir de RESTdir.
  - b1. Concrete mediante directivas los valores siguientes:  
NCOMP=43, Vdir=\$100 Wdir=\$200 COCdir=\$300 RESTdir=\$400
  - b2. Desarrolle el programa.

NOTA: Debe incluir un organigrama o similar además de las instrucciones.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 1** **02-diciembre-2019**

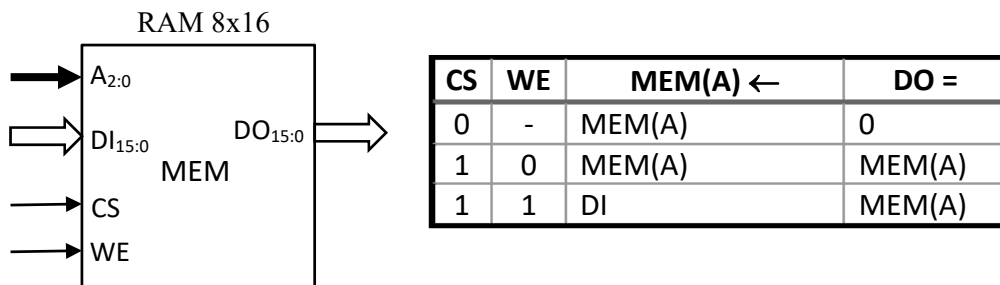
***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- a. Represente estructuralmente y funcionalmente una memoria LIFO de fondo 10 y datos bidireccionales de entrada/salida.
- b. En un sistema digital síncrono cuyo reloj es activo en la subida se desea hacer la operación  $A \leftarrow B \leftarrow C$ . ¿Es posible realizarla en una sola micro-operación? ¿Y en dos micro-operaciones? Dibuje una Unidad de Datos específica para realizarla en el menor número posible de micro-operaciones y liste dichas micro-operaciones.
- c. Explique en qué consiste la arquitectura de Von Neumann y en qué la de Harvard.
- d. ¿Qué es un ensamblador? Indique qué directivas pueden usarse para definir valores constantes en variables y qué instrucciones permiten asignar valores constantes en registros.

Problema 2. Se desea diseñar un circuito que controle la escritura de una memoria RAM de 8 palabras de 16 bits. El circuito tiene una entrada de 8 bits por el que va recibiendo los bytes a escribir en la memoria. Una señal Xs iniciará el proceso de escritura. Tras la activación, en el siguiente ciclo de reloj se recibe el Byte a escribir en la parte menos significativa de la fila 0 de la memoria. El segundo Byte recibido hay que escribirlo en la parte más significativa de la fila 0. Después se reciben los Bytes de la fila 1 (de la misma forma), después los de la fila 2, y así sucesivamente hasta completar la memoria. Todos los Bytes se reciben de forma consecutiva (en 16 ciclos de reloj).

- Proponga una unidad de datos lo más sencilla posible. Describa a nivel RT los componentes que necesite.
- Diseñe las cartas ASM de las Unidades de Datos y de Control de un controlador que permita realizar esta operación.
- Realice una implementación de la UC usando la técnica de un biestable por estado.

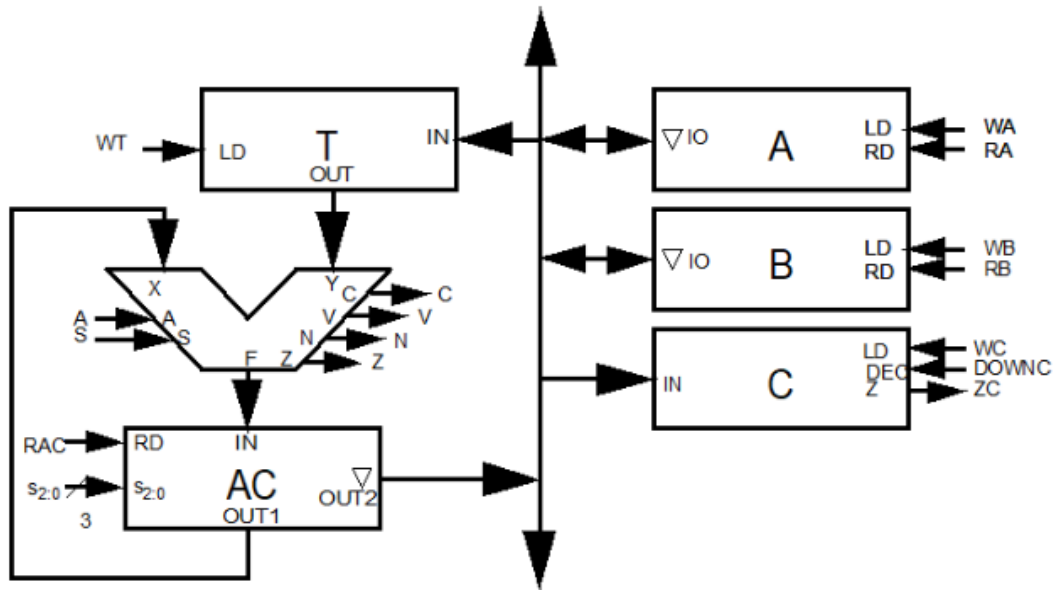


Problema 3. Se desea obtener el vector  $W(i)$  que multiplica por una constante (CTE) el vector  $V(i)$ ,  $W(i) = V(i) \times CTE$ . Tanto CTE como las componentes de  $V(i)$  son de 1Byte.

El número de componentes del vector  $V$  será el indicado por la variable  $NV$ , que será siempre no nula y menor o igual que 256. El vector  $V$  está almacenado en la SRAM a partir de la posición  $Vdir$  y el vector  $W$  se guardará en la SRAM a partir de la posición  $Wdir$ .

- a. Desarrolle un programa en ensamblador de AVR si tanto  $V(i)$  como CTE son números sin signo. En las directivas, debe particularizar para los siguientes valores:  
CTE=43, NV=128, Vdir=\$100 y Wdir=\$300.
- b. Utilizando para  $V(i)$  los valores de datos de la Hoja para Examen, muestre el valor de todos los registros del ATmega y palabras de SRAM afectados tras guardar  $W(12)$ .
- c. Indique los cambios que hay que hacer el programa si:
  - i. CTE es sin signo pero  $V(i)$  es un número con signo.
  - ii. CTE es un número con signo pero  $V(i)$  es sin signo.

La UD de la figura corresponde a un sistema digital basado en la arquitectura genérica con ALU con un bus:



- Describa a nivel RT los registros AC y C. (NOTA: el registro AC controla la lectura de su bus OUT2 con RD y soporta las operaciones siguientes en función de la palabra  $s_{2:0}$ , 0: NOP, 1: puesta a cero, 2: desplazamiento a la derecha, 3: desplazamiento a la izquierda, 4: carga en paralelo, 5: incremento, 6: decremento, 7: carga 11...11.
- Descomponga en microoperaciones cada una de las siguientes macrooperaciones.  
**Son 4 de las ocho y se han formado 16 exámenes con los números:**  
 1234, 1235, 1264, 1265, 1734, 1735, 1764, 1765,  
 8234, 8235, 8264, 8265, 8734, 8735, 8764, 8765,
- Suponiendo que sólo va a implementar la última, proponga una carta ASM de Datos y de Control usando la estructura típica con señal de comienzo Xs y señal de FIN.

- $C \leftarrow Ca1(A)$
- $C \leftarrow Abs(A)$  (Valor absoluto de número en complemento a 2)
- $C \leftarrow 20 * A$  (Operación producto algebraico)
- $C \leftarrow Mayor(A,B)$  (Para A y B sin signo)
- $C \leftarrow Menor(A,B)$  (Para A y B sin signo)
- $C \leftarrow 24 * A$  (Operación producto algebraico)
- $C \leftarrow (A+B)/2$
- $C \leftarrow Ca2(A)$



Considere la subrutina siguiente escrita en ensamblador del CS2010:

```

;-----
; MulU                                v1.0
;
; Multiplicación algebraica de 2 números de 8 bit sin signo.
; Entradas:
; R0 y R1 dos números de 8 bits sin signo, A y B.
; Salidas:
; R1:R0 = AxB resultado de 16 bits sin signo, donde R1 es el MSB y R0 el LSB.
;-----
MulU:    mov    r4, r1                ; 1:
         ldi    r2, 0                ; 2:
         ldi    r3, 8                ; 3:
MuluBuc: ror    r4                    ; 4:
         brcs  MuluSum              ; 5:
         jmp   MuluNoSum            ; 6:
MuluSum: add    r2, r0              ; 7:
MuluNoSum: ror   r2                  ; 8:
         ror   r1                    ; 9:
         subi  r3, 1                 ;10:
         brzs  MuluFin              ;11:
         jmp   MuluBuc              ;12:
MuluFin: mov    r0, r1              ;13:
         mov   r1, r2                ;14:
         ret                               ;15:

```

- Analice e indique qué hace cada instrucción. Use comentarios significativos (no es necesario que copie el programa; use el número de línea que se pone como comentario).
- Obtenga el código máquina en hexadecimal de las dos primeras instrucciones.
- Sea A el número formado por los 2 dígitos más significativos de su DNI y B el formado por los 2 menos significativos. Si cualquiera de ellos es igual a 00 o 01, deseche el dígito más externo y coja el siguiente más interno (Ejemplo 1, si DNI=12,345.678-K, A=12, B=78; Ejemplo 2, si DNI=01,234.500-X, A=12, B=50). Suponga que R0=A y que R1=B. Rellene el siguiente formulario (en binario) en el proceso de análisis (no olvide considerar el *carry* en las instrucciones de desplazamiento). Indique el valor de cada registro justo antes de la instrucción 4 (iteración 0) y justo después de la instrucción 10 (iteración i). Añada las líneas que necesite y compruebe que el algoritmo funciona:

Iteración	R4	R3	R2 (antes de 8)	R2	R1
0					
1					
...					

- Describa qué papel juega cada registro en la subrutina.
- Escriba un trozo de código en ensamblador que cargue sus A y B personales en R0 y R1 respectivamente y llame a MulU.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 2** **15-septiembre-2020**

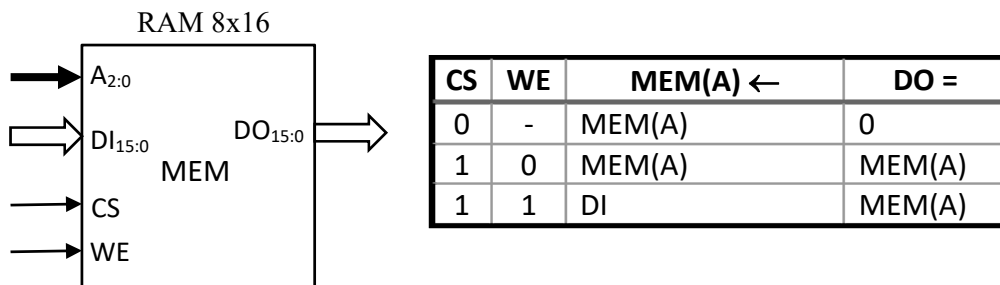
***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- a. Represente estructuralmente y funcionalmente una memoria LIFO de fondo 10 y datos bidireccionales de entrada/salida.
- b. En un sistema digital síncrono cuyo reloj es activo en la subida se desea hacer la operación  $A \leftarrow B \leftarrow C$ . ¿Es posible realizarla en una sola micro-operación? ¿Y en dos micro-operaciones? Dibuje una Unidad de Datos específica para realizarla en el menor número posible de micro-operaciones y liste dichas micro-operaciones.
- c. Explique en qué consiste la arquitectura de Von Neumann y en qué la de Harvard.
- d. ¿Qué es un ensamblador? Indique qué directivas pueden usarse para definir valores constantes en variables y qué instrucciones permiten asignar valores constantes en registros.

Problema 2. Se desea diseñar un circuito que controle la escritura de una memoria RAM de 8 palabras de 16 bits. El circuito tiene una entrada de 8 bits por el que va recibiendo los bytes a escribir en la memoria. Una señal Xs iniciará el proceso de escritura. Tras la activación, en el siguiente ciclo de reloj se recibe el Byte a escribir en la parte menos significativa de la fila 0 de la memoria. El segundo Byte recibido hay que escribirlo en la parte más significativa de la fila 0. Después se reciben los Bytes de la fila 1 (de la misma forma), después los de la fila 2, y así sucesivamente hasta completar la memoria. Todos los Bytes se reciben de forma consecutiva (en 16 ciclos de reloj).

- Proponga una unidad de datos lo más sencilla posible. Describa a nivel RT los componentes que necesite.
- Diseñe las cartas ASM de las Unidades de Datos y de Control de un controlador que permita realizar esta operación.
- Realice una implementación de la UC usando la técnica de un biestable por estado.



Problema 3. Se desea obtener el vector  $W(i)$  que multiplica por una constante (CTE) el vector  $V(i)$ ,  $W(i) = V(i) \times CTE$ . Tanto CTE como las componentes de  $V(i)$  son de 1Byte.

El número de componentes del vector  $V$  será el indicado por la variable  $NV$ , que será siempre no nula y menor o igual que 256. El vector  $V$  está almacenado en la SRAM a partir de la posición  $Vdir$  y el vector  $W$  se guardará en la SRAM a partir de la posición  $Wdir$ .

- a. Desarrolle un programa en ensamblador de AVR si tanto  $V(i)$  como CTE son números sin signo. En las directivas, debe particularizar para los siguientes valores:  
CTE=43, NV=128, Vdir=\$100 y Wdir=\$300.
- b. Utilizando para  $V(i)$  los valores de datos de la Hoja para Examen, muestre el valor de todos los registros del ATmega y palabras de SRAM afectados tras guardar  $W(12)$ .
- c. Indique los cambios que hay que hacer el programa si:
  - i. CTE es sin signo pero  $V(i)$  es un número con signo.
  - ii. CTE es un número con signo pero  $V(i)$  es sin signo.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 1** **17-diciembre-2020**

***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Este problema contiene dos partes al 50% de nota: una primera parte de 10 preguntas tipo test y una segunda con dos preguntas de respuestas cortas:

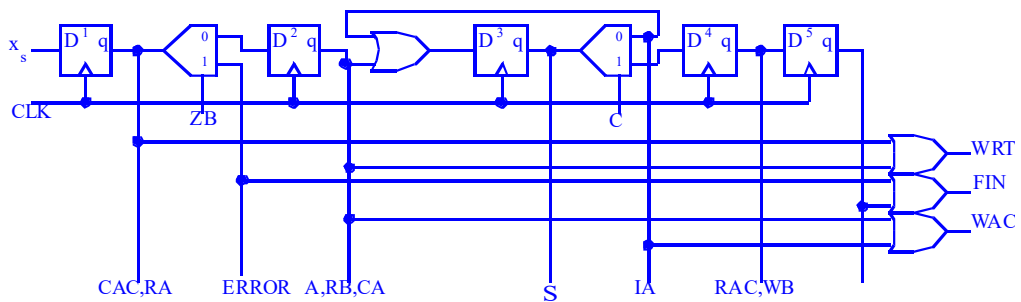
a. Test

1. La rapidez de una memoria se mide a través de:
  - a) Si es de acceso secuencial o aleatorio
  - b) De los kB, MB, GB,..., que pueda transferir
  - c) De la velocidad y del tiempo de acceso
  - d) El coste: mientras más caras, más rápidas
  
2. Una memoria de acceso aleatorio tiene, al menos, direcciones físicas entre \$654 y \$ABCD. ¿Cuántas líneas de dirección tiene como mínimo?
  - a) 14
  - b) 16
  - c) 18
  - d) 20
  
3. En una descripción verilog de un circuito con entrada SET (puesta a 1) se ha escrito un `always@(posedge clk, negedge SET)`:
  - a) Se trata de una puesta a 1 asíncrona y SET es activo en alta
  - b) Se trata de una puesta a 1 síncrona y SET es activo en alta
  - c) Se trata de una puesta a 1 asíncrona y SET es activo en baja
  - d) Se trata de una puesta a 1 síncrona y SET es activo en baja
  
4. La sentencia `case ()`:
  - a) Se puede usar en descripción procedimental exclusivamente
  - b) Se puede usar en descripción funcional exclusivamente
  - c) Se puede usar en descripción estructural exclusivamente
  - d) Se puede usar en cualquier tipo de descripción.
  
5. Un sistema digital puede tener:
  - a) Varias micro-operaciones que duren diferentes ciclos
  - b) Tiene que tener solamente una macro-operación
  - c) Varias macro-operaciones que deben durar los mismos ciclos
  - d) Varias macro-operaciones que duren diferentes ciclos
  
6. Si desea diseñar la calculadora de 8 registros sin multiplexores:
  - a) Es imposible porque los multiplexores son insustituibles
  - b) Basta cambiar los multiplexores por decodificadores
  - c) Basta unir las salidas de los registros de lectura incondicional a dos buses compartidos.
  - d) Utilizo buses compartidos con registros con salida de alta impedancia
  
7. La arquitectura de Harvard en los computadores se refiere a:
  - a) La jerarquía de memoria de los programas desde la masiva hasta IR

- b) La disponibilidad de doble memoria, de código y de datos
  - c) La jerarquía de memoria de los datos desde la masiva hasta Rd (R0...R7)
  - d) La disponibilidad de una memoria con código y datos compartidos
8. El SP de un computador es el registro...:
- a) Que apunta a la memoria de código (SP: Señala Programa)
  - b) Donde se guardan los datos de interfaz a la memoria (push/pop)
  - c) Donde se guardan los flags de estado
  - d) Que apunta a la pila (posición ociosa: primera palabra vacía)
9. ¿En qué memoria se guardan datos en un AtmegaX8P?
- a) En EEPROM
  - b) En SRAM
  - c) En Flash
  - d) Son falsas las otras respuestas
10. Siendo X=\$31A, indique el operando de memoria accedido por LD R0,-X para los valores iniciales dados en la **Hoja para Examen**:
- a) \$A1
  - b) \$91
  - c) \$1B
  - d) \$1A

b. Preguntas de respuestas cortas:

1. Describa en verilog un contador mód. 8 con puesta a 0, cuenta ascendente y salidas de *carry* (acarreo) y del estado de cuenta. Especifique lo restante que necesite para lograr una descripción válida.
2. Dada la siguiente implementación de una carta ASM usando la técnica de un biestable por estado, obtenga la carta ASM de la UC correspondiente



Problema 2. Un sistema digital incorpora una memoria Mem de 256 B. Se desea que pueda escribir lotes de 16 datos que se escribirán en posiciones consecutivas. Tanto la dirección de comienzo del lote como los 16 datos los recibirá por un bus de entrada Din tal como se explica a continuación.

El sistema comenzará tras la activación de Xs. Primero, en Din aparecerá la primera dirección cuando se active una entrada de validación de direcciones, Vadr = 1. Después, cuando se active una entrada de validación de datos, Vdat = 1, aparecerá en Din el valor de cada uno de los 16 datos de forma consecutiva, uno por cada ciclo de reloj, para que se escriban en la Mem. Tras terminarse de escribir los 16 datos en Mem, el sistema generará FIN y parará.

- a. Diseñe una unidad de datos específica para este problema.
- b. Escriba la carta ASM de datos y de control.

Problema 3. Considere la subrutina siguiente escrita en ensamblador del CS2010:

```

;-----
; MulU                                v1.0
;
; Multiplicación algebraica de 2 números de 8 bit sin signo.
; Entradas:
; R0 y R1 dos números de 8 bits sin signo, A y B.
; Salidas:
; R1:R0 = AxB resultado de 16 bits sin signo, donde R1 es el MSB y R0 el LSB.
;-----
MulU:   mov    r4, r1        ; 1:
        ldi    r2, 0        ; 2:
        ldi    r3, 8        ; 3:
MulBuc: ror    r4            ; 4:
        brcs  MuluSum      ; 5:
        jmp   MuluNoSum    ; 6:
MulSum: add    r2, r0       ; 7:
MulNoSum: ror   r2          ; 8:
        ror   r1           ; 9:
        subi  r3, 1        ;10:
        brzs  MuluFin      ;11:
        jmp   MuluBuc      ;12:
MulFin: mov    r0, r1       ;13:
        mov   r1, r2       ;14:
        ret                               ;15:

```

- Analice e indique qué hace cada instrucción. Use comentarios significativos.
- Obtenga el código máquina en hexadecimal de las dos primeras instrucciones.
- Suponga que  $R0=12$  y  $R1=78$ . Rellene en binario ( $R3$  puede dejarlo en decimal) la tabla de la siguiente página durante el proceso de análisis (**no olvide considerar el carry en las instrucciones de desplazamiento**). Para ello, en la fila "iteración 0" indique el valor de los registros justo antes de la instrucción 4. En las siguientes iteraciones (1, ...), indique los valores justo después de la instrucción 10. Compruebe que el algoritmo funciona.
- Describa qué papel juega cada registro en la subrutina.
- Escriba un trozo de código en ensamblador que cargue sus A y B personales en R0 y R1 respectivamente y llame a MulU.





## ESTRUCTURA DE COMPUTADORES – GII – IC

### Examen Final Convocatoria 1

5-julio-2021

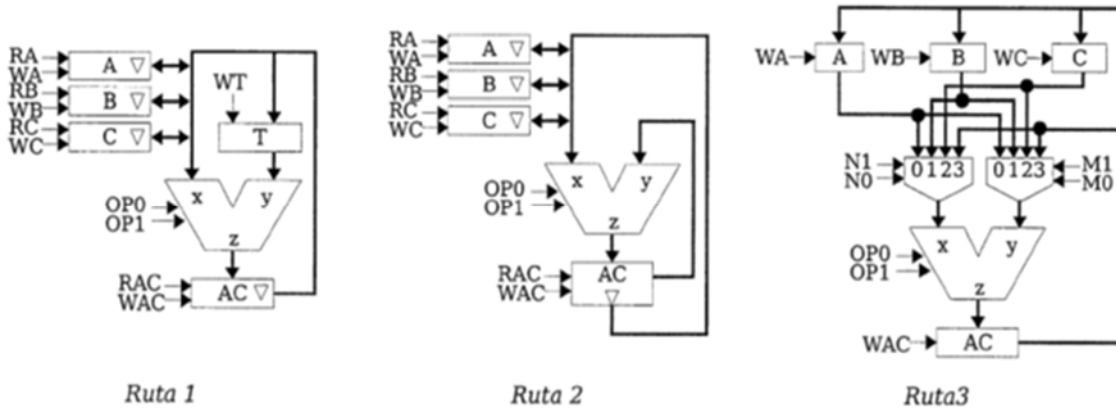
**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- a. Un procesador con AB[18] y DB[8] tiene acceso a datos de tamaño B únicamente (o sea, es no segmentado). En su mapa de memoria se desea ubicar un chip de memoria **M** de **64kB** con selección de chip activa en H a partir de la dirección **\$18000** sin ocupar más espacio lógico que el que ocupa M físicamente. Indique con rigor:
  - a/**Todas las conexiones** entre el procesador y M.
  - b/**Las direcciones lógicas** correspondientes a M(011...11) y a M(100...00).
- b. Un circuito posee **tres variables binarias escalares** de un bit (**a, b y c**). Para su descripción verilog se va a usar la sentencia "case":
  - a/Explique brevemente la **sintaxis** de esa sentencia "case".
  - b/ Utilícela para **describir un decodificador 3:8** cuyas entradas son esas tres variables (**a, b y c**, de MSB a LSB).
- c. Considere la calculadora de 8 registros que tiene en la hoja de material para exámenes. Se desean hacer las operaciones siguientes **sin que resulten afectados otros registros** que los que implicados en ellas. (Las dos operaciones son independientes entre sí). Para cada operación especifique la secuencia de micro-operaciones necesarias (sean una o varias)
  1.  $R4 \leftarrow 0$
  2.  $R1 \leftarrow R7 + R3$
- d. Considere en el ATmega328P.
  - a/Explique qué significan los bits de estado S y N
  - b/Aplique la instrucción ADD R20, R3 al caso en que os valores iniciales sean  $R20 = \$77$  y  $R3 = \$38$ , obteniendo S y N. Compruebe que estos resultados son coherentes con su respuesta al epígrafe anterior.

Problema 2. En la figura se muestran tres unidades de datos diferentes (denominadas Ruta 1, 2 y 3) que interconectan los registros A, B, C y AC con una ALU.

- a. Describa formalmente los siguientes componentes a nivel RT. Para ello debe dar un nombre interno apropiado a cada señal de cada módulo; p.ej. los del registro AC de Ruta 3 podrían ser "in" para sus entradas de dato, "out" para sus salidas y "w" para la señal de control que está conectada a WAC.
  - i. La ALU, asumiendo que debe hacer las cuatro operaciones siguientes: suma, resta ("x" como minuendo), incrementa "x" y transfiere "x".
  - ii. El registro AC de cada una de las tres Rutas. (Por tanto, hay que dar tres descripciones. Observe que AC tiene dos buses de salida diferentes en la Ruta 2).
- b. Presente la carta ASM de **Datos** y de **Control** para realizar la macro-operación  $A \leftarrow A + B + C$  usando:
  - i. La Ruta 1.
  - ii. La Ruta 2.
  - iii. La Ruta 3.
- c. Compare las tres soluciones discutiendo sus ventajas e inconvenientes.



Problema 3. En el **CS2010**, suponga que dispone de una subrutina **MulU** que calcula el producto de dos números de 8 bits sin signo A y B y obtiene un resultado de 16 bits. Recibe en R0 y en R1 los parámetros A y B, deja el resultado en el par R1:R0 (R1 es el MSB) y en su ejecución altera los registros R2, R3 y R4.

- a. Explique el algoritmo (con pseudocódigo o diagrama de flujo) y escriba en el ensamblador del CS2010 una subrutina, llamada **Abs**, que realice  $R2 \leftarrow |R2|$ , siendo el contenido inicial de R2 un número de 8 bits con signo (Ca2). Además, escribirá en R3 un 0 si el número era positivo y 1 si era negativo.
- b. Explique el algoritmo (con pseudocódigo o diagrama de flujo) y escriba un programa denominado MulS que llama a MulU y a Abs para multiplicar los 2 números de 8 bits con signo en Ca2. Los argumentos de entrada y de salida de MulS son los mismos que MulU.

**Nota:** Recuerde que, para cualquier número de bits de A, el Ca2(A) lo puede obtener bien como  $0 - A$ , bien como  $Ca1(A)+1$ , bien copiar los bits comenzando por el lsb y, una vez que se copie el primer 1, los restantes se copian invertidos.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 2** **14-septiembre-2021**

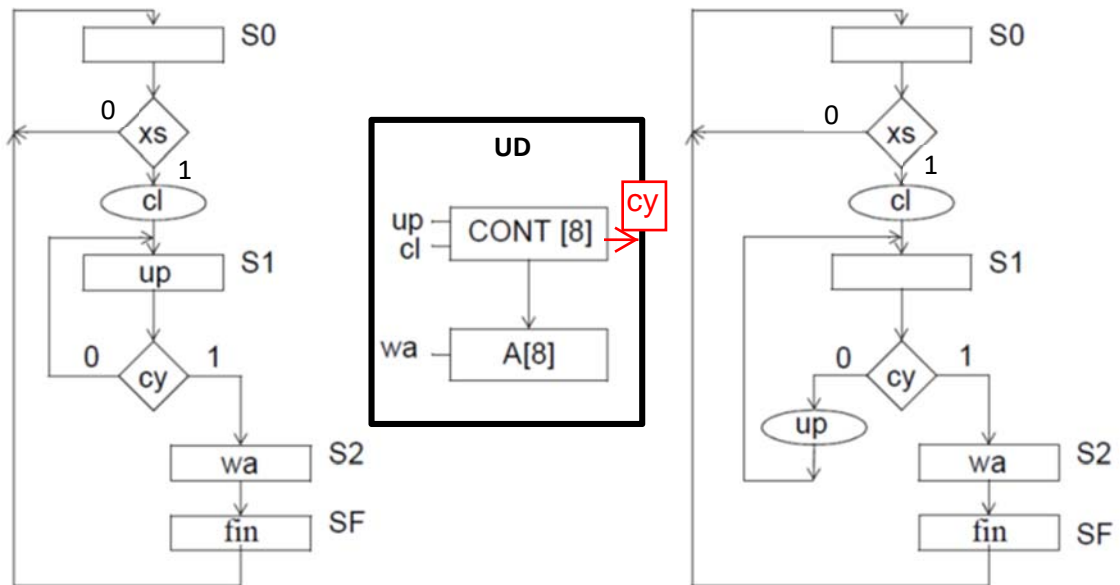
***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Responda breve y claramente a las siguientes cuestiones:

- a. Describa y compare las distintas formas que conozca de implementar una pila.
- b. Describa en Verilog un contador reversible módulo 32 disparado por flanco positivo con entradas de inhibición, carga en paralelo asíncrona y salida de *carry/borrow*.
- c. Arquitecturas de referencia básicas para almacenar datos/programas en computadores: descripción y comparación.
- d. Indique las zonas de memoria que es posible encontrar en la memoria de datos del AVR, así como las instrucciones de movimiento de datos y modos de direccionamiento para acceder a las mismas (no es necesario que indique las direcciones de los bloques).

Problema 2. Considere la Unidad de Datos (UD) de la figura, donde A y CONT son un registro y un contador de 8 bits respectivamente, y las dos cartas ASM asociadas.

- Describa en Verilog la UD. (No olvide incluir una señal de reloj, clk, activa en el flanco positivo, no dibujada en la figura).
- Explique la diferencia entre ambas cartas ASM e indique qué valor tomará el registro A al finalizar la ejecución en cada caso.
- ¿Cuántos ciclos de reloj transcurren en cada caso desde que se captura XS=1 hasta que se activa FIN?
- Explique si esa UD permite sustituir las cajas de estado S2 por cajas de acción condicional y, en caso afirmativo, diga qué efecto tendría esto sobre la operación del sistema.
- Obtenga la Unidad de Control de cada carta ASM original.



Problema 3. Se tienen en la memoria de datos dos vectores P y Q cada uno con **N** componentes consistentes en números con signo (Ca2) de 8 bits, P(i) y Q(i). Se pretende que, para cada pareja, P(i) se quede con la componente mayor y Q(i) con la menor. (Por ejemplo, si inicialmente P(1)=+3 y Q(1) =+7, los valores finales serán: P(1)=+7, Q(1) =+3, mientras que si P(2)= -3 y Q(2)=-7, entonces, finalmente, P(2)= -3 y Q(2)= -7).

Los primeros elementos de cada vector están en las posiciones **primeP** y **primeQ**. Realice las siguientes tareas en el **ensamblador del AVR**:

- a. Defina mediante directivas **N**, **primeP** y **primeQ**, e inícielas con los valores 10, \$100 y \$120, respectivamente.
- b. Exprese un organigrama o pseudocódigo que resuelva la función enunciada.
- c. Escriba el programa en ensamblador correspondiente.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 1** **16-diciembre-2021**

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

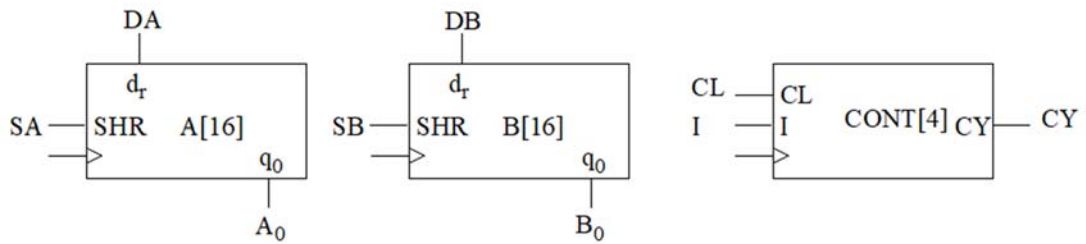
- a. Considere un chip de memoria **M** de **32kB** con selección de chip CS activa en H, cuyas líneas de dirección son  $\mathbf{a}_k:\mathbf{a}_0$  y las de datos,  $\mathbf{d}_j:\mathbf{d}_0$ . El chip M forma parte de la memoria de un procesador con AB[16] y DB[8] y se activa con  $\mathbf{CS} = \mathbf{A}_{15}\oplus\mathbf{A}_{14}$ . Indique con rigor:
- a.1. La dimensión de los buses de M y los valores de “k” y de “j”.
  - a.2. Indique las conexiones entre el procesador y M.
  - a.3. Dibuje el mapa de memoria.
  - a.4. Las direcciones lógicas correspondientes a los dos posiciones de la memoria M siguientes:  $M(0)$  y  $M(11\dots11)$
- b. El código Verilog adjunto contiene **5 errores**. Encuéntrelos. Corríjalos.

```
// Código con 5 errores para un contador
module contador_mod_16
    input clk, up, reset,
    output [3:0] q, output cy;
    assign cy = &q;
    always @(posedge clk, up)
    if (Reset==1 )
    q <= 0;
    else if (up==1)
    q <= q + 1
endmodule
```

- c. En un sistema digital describa las **funciones** de:
- a/ La Unidad de Datos
  - b/ La Unidad de Control
- d. En el instante actual y para el computador simple CS2010, PC=\$23 y acaba de terminar la ejecución de una instrucción. Se sabe que MEMCOD(\$22)=\$9000, MEMCOD(\$23)=\$1E59, MEMCOD(\$24)=\$2043,...
- a/ Indique el mnemónico de la instrucción que se ejecutará seguidamente y especifique formalmente qué hace dicha instrucción
  - b/Desarrolle las micro-operaciones de ambos ciclos, *Fetch* y *Execute*
- e. Desarrolle un programa en ensamblador del AVR que permita realizar  $33_{(10)}$  veces un bucle que contiene menos de 33 instrucciones.



Problema 2. Para la unidad de datos (UD) de la figura:



- Describa a nivel RT el registro A. SHR es una señal de desplazamiento a la derecha,  $d_r$  es la entrada de datos serie y  $q_0$  es la salida del bit menos significativo.
- Describa a nivel RT el contador CONT. CL es una señal de puesta a cero síncrona e I es una señal de incremento; CY es salida de carry.
- Se pretende hacer la instrucción  $A \leftarrow Ca1(B)$  cuando  $X = 0$  y  $A \leftarrow Ca2(B)$  cuando  $X = 1$ . Obtenga las cartas ASM de datos y de control.
- Diseñe la unidad de control (UC) usando la técnica de un biestable por estado.

**RECORDATORIO:** una forma habitual de calcular el Ca2 es empezar a copiar los bits empezando por el bit menos significativo. Una vez que se copie el primer 1, los demás se copian invertidos.

Problema 3. En el ámbito del computador CS2010, considere el siguiente programa:

```
LDI R5, 10
LDI R4,0
LDS R0, $A0
LDI R1, $B0
siguiente: $0A01
           $5200
           $3108
           $C401
salta:     SUBI R5,1
           CPI R5,0
           BREQ etiq
           ADDI R1,1
           JMP siguiente
etiq:     STS $C0, R4
           STOP
```

- a. Interprete de ensamblador a código máquina y viceversa (40%). En particular:
  - i. Obtenga el **código máquina en hexadecimal** de las cuatro últimas instrucciones.
  - ii. Obtenga en **ensamblador** las cuatro instrucciones dadas en código máquina.
- b. Indique con claridad la operación del programa (60%). En particular:
  - i. Considerando que se ejecuta este programa, desarrolle una de las dos opciones siguientes (no tiene por qué realizar las dos): 1/Presente el flujo de datos; 2/Realice un organigrama.
  - ii. Exprese con claridad la tarea global que realiza el programa e identifique los registros afectados.

## ESTRUCTURA DE COMPUTADORES – GII – IC

### Examen Final Convocatoria 1

20-junio-2022

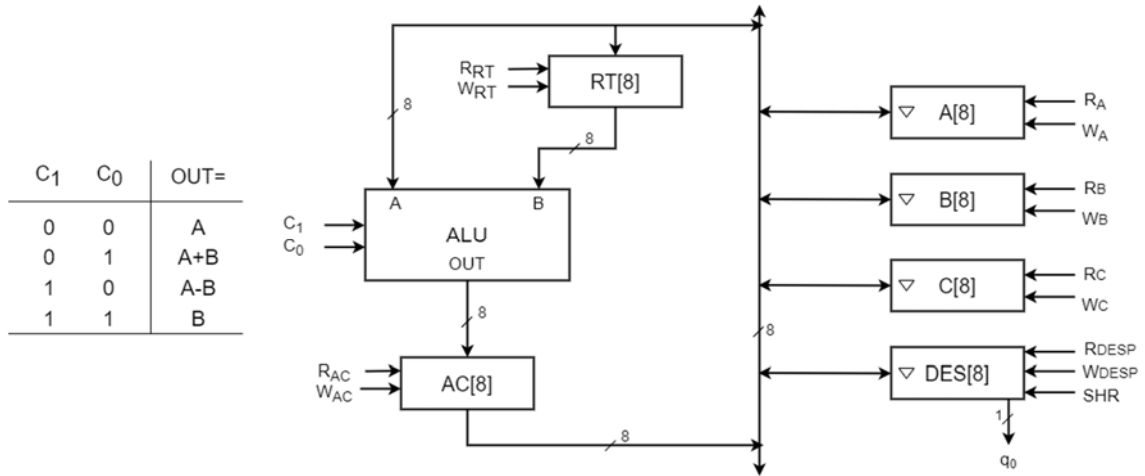
**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- 1) Se desea ubicar un chip de memoria **M** de **128 kB** con selección de chip CS activa en H sin ocupar más espacio lógico que el que ocupa M físicamente a partir de la dirección **\$10000** a un procesador con AB[19], DB[8] y no segmentado (esto es, sus instrucciones de memoria sólo acceden a 1 B). Indique con rigor:
  - a/**Todas las conexiones** de CS, direcciones y datos entre el procesador y M. (No considere R ni W).
  - b/**Las direcciones lógicas** correspondientes a M(011...11) y a M(100...00).
  - c/Dato físico al que se accede en los casos: AB = \$20000 y AB= \$DCAFE.
  
- 2) Se desea un circuito decodificador 3:8 con *enable* EN activo en alta, cuyas entradas sean X2, X1, X0 y cuyas salidas sean OUT(j) con j=0, 1,...7. Realice su descripción verilog usando la sentencia "case". En concreto:
  - a/Explique brevemente la **sintaxis** de esa sentencia "case".
  - b/Usándola, **describa en verilog el decodificador 3:8** enunciado.
  
- 3) Con la calculadora de 8 registros que tiene en la hoja de material para exámenes se desean hacer las dos operaciones siguientes. Los valores iniciales afectados solamente deben ser los de los registros implicados en ellas. Las dos operaciones son independientes entre sí. Para cada una, operación especifique la secuencia de microoperaciones necesarias (sean una o varias):
  1.  **$R1 \leftarrow 0$**
  2.  **$R1 \leftarrow R2 - R3$**
  
- 4) Considere la instrucción **LD R2, (R3)** del CS2010. Explique con rigor:
  - a/Su operación a nivel de usuario identificando todos los registros afectados
  - b/La secuencia de microoperaciones de su **ciclo de ejecución**.
  
- 5) Desarrolle un programa en ensamblador del AVR que permita realizar  $33_{(10)}$  veces un bucle. (Considere que contiene menos de  $30_{(10)}$  instrucciones).

Problema 2. Considere la unidad de datos de la figura. Se busca dejar el resultado de la suma entre A y B cuando A es impar. En otro caso, intercambiar el contenido de A y B, dejando en A el contenido de B y en B el contenido de A.

- a) Obtenga la carta ASM de datos y control de este sistema digital.
- b) Implemente la unidad de control utilizando la técnica de un biestable por estado.



Problema 3. Se desea obtener el punto medio ( $X_m, Y_m$ ) de otros dos puntos ( $X_0, Y_0$ ) y ( $X_1, Y_1$ ) del plano cartesiano. Todos los valores son números positivos en complemento a 2 (Ca2) de tamaño Byte. Los valores de partida se encuentran almacenados en las siguientes direcciones de memoria:  $X_0$  en la dirección \$180,  $X_1$  en la \$181,  $Y_0$  en la \$190 e  $Y_1$  en la \$191. Las coordenadas del punto medio se guardarán en \$182 ( $X_m$ ) y \$192 ( $Y_m$ ).

Desarrolle un programa que realice esa tarea en ensamblador del **AVR**. El programa deberá poner 0 en la palabra de memoria \$1A0 si  $X_m$  e  $Y_m$  han sido números enteros, un 1 si solo hay una coordenada fraccionaria y un 2 si lo son las dos. En caso de resultado fraccionario debe aplicar redondeo para obtener las coordenadas del punto medio.

- a) En primer lugar, describa su solución mediante un organigrama.
- b) Escriba el programa en lenguaje ensamblador, comentándolo adecuadamente.
- c) Justifique si hay problemas de desbordamiento. (No hay que modificar el programa).

## ESTRUCTURA DE COMPUTADORES – GII – IC

### Examen Final Convocatoria 2

13-septiembre-2022

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- 1) Considere un chip de memoria **M** de **32kB** con selección de chip **CS** activa en H y cuyas líneas de dirección físicas se llamarán **a<sub>k</sub>:a<sub>0</sub>** y las líneas de datos, **d<sub>j</sub>:d<sub>0</sub>**. El chip M forma parte de la memoria de un procesador con AB[16] y DB[8] no segmentado y se activa con  $CS = A_{15} \oplus A_{14}$ . Indique con rigor:
  - a/ Los valores de “k”, de “j”, dibuje el mapa de memoria e **indique las conexiones** entre las líneas de dirección y de datos del procesador y las de M.
  - b/ **Las direcciones lógicas** correspondientes a los 4 casos siguientes:  
M(0), M(011...1), M(100...0) y M(11...11)
  - c/ **Las direcciones físicas** a las que se accede con los 4 casos siguientes:  
AB=\$333, AB=\$789A, AB=\$89AB y AB=\$CAFE
- 2) El código Verilog adjunto contiene 5 errores. Encuéntrelos. Corrija los.

```
// Código con 5 errores
module contador_mod_16(
input clk, up, reset,
output [3:0] q, output cy;
assign CY = &q;
always @(posedge clk, up)
if (reset==1 )
q <= 0;
else if (up==1);
q <= q + 1;
endmodule
```

- 3) Describa las **funciones** de UD y de UC en un sistema digital:
  - a/ UD = Unidad de Datos
  - b/ UC = Unidad de Control
- 4) Para el computador simple CS2010, en el instante actual PC=\$23 y acaba de terminar la ejecución de una instrucción. Se sabe que MEMCOD(\$22)=\$9000, MEMCOD(\$23)=\$1E59, MEMCOD(\$24)=\$2043,...
  - a/ Indique el mnemónico de la instrucción que se ejecutará seguidamente y especifique formalmente qué hace dicha instrucción, así como el valor de PC al terminarla.
  - b/Desarrolle las micro-operaciones de ambos ciclos, *Fetch* y *Execute*.
- 5) Considere en el ATmega328P la instrucción ADD R20, R3.
  - a/Explique qué significan los bits de estado S y N
  - b/Aplíquelo al caso en que sean R20 = \$77 y R3 = \$38 los valores previos, comprobando la validez de su respuesta al epígrafe anterior.

Problema 2. Un sistema digital tiene un bus de entrada XIN de 8 bits por el que se reciben datos cuando está activa otra entrada Valin (de 1 bit). El sistema opera de la siguiente forma:

- Tras activarse Xs, por XIN se introduce un número sin signo (A) estando Valin activa.
- Después, ante una nueva activación de Valin, se introduce un dato B.
- Entonces, el sistema debe obtener el producto aritmético  $A*B$  y, al acabar, activar FIN.

Debe optar si va a hacer  $A*B$  por el algoritmo de sumas y desplazamientos a la derecha (como en la práctica) o si usará el algoritmo de sumas sucesivas (como el utilizado en algunos problemas resueltos en clase).

- 1) Diseñe una unidad de datos específica para este problema. Describa los componentes a nivel RT.
- 2) Presente la carta ASM de datos y de control.
- 3) Dibuje la unidad de control mediante 1 biestable por estado.

Problema 3. Sea un vector de 10 elementos ( $N_0, N_1, \dots, N_9$ , números sin signo de 1 Byte) situados a partir de la posición de memoria  $M(\text{DirInic})$ , donde  $\text{DirInic}$  es una dirección de la memoria de datos. Se desea un programa en ensamblador AVR que almacene a partir de la posición de memoria  $\text{DirDoble}$  el vector multiplicado por 2:  $S_0=N_0*2, S_1=N_1*2 \dots S_9=N_9*2$ . No considere desbordamiento. Así mismo, el programa debe usar el registro  $R0$  como bandera que informe con un valor de 0 cuando el programa esté ejecutándose y con un valor  $\$FF$  para indicar que ya ha terminado la tarea.

- 1) Desarrolle un organigrama que ilustre cómo realiza su programa.
- 2) Utilizando directivas, defina la variable  $\text{DirInic}$  con el valor  $\$120$  y  $\text{DirDoble}$  con el valor  $\$200$ .
- 3) Indique la solución para  $R0$ .
- 4) Escriba una subrutina denominada  $\text{Vdoble}$  que resuelva el problema planteado (vector por 2).
- 5) Presente el programa global.
- 6) Haga las modificaciones necesarias en la subrutina anterior, para crear una nueva subrutina (denominada  $\text{Vmitad}$ ) en la que, en lugar de la multiplicación por 2, se realice la división entre 2 cuando el número es par. Cuando el número es impar, el elemento debe quedar sin alterar. Indique qué y cómo cambia el programa principal completo en este caso.



**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 3** **24-noviembre-2022**

***Apellidos, Nombre:*** \_\_\_\_\_

Problema 1. Responda a las siguientes cuestiones teóricas:

- a. Memorias: compare las memorias EEPROM y la SRAM
- b. Verilog: explique las diferencias entre asignamientos bloqueantes y no bloqueantes. Ayúdese de un ejemplo.
- c. RT: explique las similitudes y las diferencias entre la calculadora de 8 registros estudiada y el computador simple 1 tanto a nivel de Unidad de Datos como de Control.
- d. CS: describa 6 modos de direccionamiento que están disponibles en alguno de los computadores estudiados (CS1, CS2, CS2010, AVR).

Problema 2. Un sistema digital recibe 1024 datos por un bus de entrada de 16 bits, XIN, y se quiere averiguar cuántos de ellos son menores que un cierto valor umbral, UMB. Todos los datos son números sin signo. El sistema mostrará el resultado de la cuenta por sus salidas OUT.

El sistema operará de la siguiente forma: Tras recibir la señal de comienzo Xs, el sistema capturará cada dato cuando se active una señal externa DVAL (Dato Válido). El primer dato que recibe es el valor umbral UMB. A continuación irá recibiendo los 1024 datos, cada uno tras su correspondiente validación con DVAL.

- a. Dibuje un bloque que muestre todos los terminales del sistema. Tiene que dar el tamaño adecuado de cada terminal.
- b. Proponga una **Unidad de Datos específica** para este sistema. Describa a nivel RT los componentes que utilice y dimensione los buses de conexión.
- c. Dé la carta ASM de Datos y de Control.
- d. Diseñe la Unidad de Control basada en un biestable por estado.

Problema 3. Sean  $N = N_7 \dots N_1 N_0$  un número de 8 bits y  $R = R_7 \dots R_1 R_0$  un registro de 8 bits. El MSB de  $N$  es  $N_7$  y el LSB es  $N_0$ . Por su parte,  $R_7$  es la posición más a la izquierda de  $R$  y  $R_0$  es la que está más a la derecha. El almacenamiento por defecto de  $N$  en  $R$  se hace de la forma directa ( $R = N_7 \dots N_1 N_0$ ), mientras que la forma *bit reverse* se hace al revés,  $R = N_0 \dots N_6 N_7$ .

Para el conjunto de instrucciones del ATmega328:

- a. Desarrolle la subrutina "revertirB" que ejecute  $R0 \leftarrow \text{reverse}(R0)$  donde la operación reverse cambia la representación bit directa a bit reverse o viceversa
  - i. Asuma que puede modificar el contenido de otros registros internos, indicando cuáles.
  - ii. Proponga otra solución para conservar los valores de todos los registros internos (obviamente, salvo  $R0$ ).
- b. Suponga que hay un vector  $V1$  de dimensión  $DIM$  ( $\leq 255$ ) almacenado en la memoria de datos desde la dirección  $DIR1$  y siguientes. Debe obtener un programa que almacene en otro vector,  $V2$ , el "reverse" de cada Byte de  $V1$ . El vector  $V2$  debe almacenarse a partir de la dirección  $DIR2$ .
  - i. Escriba **mediante directivas** la parte correspondiente en ensamblador para dar los valores  $DIM=75=9B$ ,  $Vdir=300$  y  $Wdir=400$ .
  - ii. Escriba el programa para obtener  $V2(k) \leftarrow \text{reverse}(V1(k))$  (para todo  $k$ ).

# ESTRUCTURA DE COMPUTADORES – GII – IC

## Examen Final Convocatoria 1

5-jun-2023

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- a. A un procesador simple (AB[11], DB[8] no segmentado) se le desea ubicar un chip de memoria **M** de **512 Bytes** con selección de chip activa en H a partir de la dirección **\$400** sin ocupar más espacio lógico que el que ocupa M físicamente. Indique con rigor:
  - a/**Todas las conexiones** entre el procesador y M incluyendo una expresión para la señal de selección de chip.
  - b/**Las direcciones lógicas** correspondientes a M(011...11) y a M(100...00).
  - c/**Las direcciones físicas** correspondientes a las direcciones lógicas AB1=\$567 y AB2=\$B0C.
  
- b. Un circuito posee tres variables binarias escalares (a, b y c, de MSB a LSB). Para su descripción verilog se va a usar la sentencia "if" de forma que la condición a considerar es que "abc > 5":
  - a/Explique brevemente la **sintaxis** de esa sentencia "if" sobre los valores de las tres variables **variables escalares, a, b y c**.
  - b/Aplíquela para describir un circuito cuya **función sea**  $z(abc)=\Sigma(0,1,4,5)$  si  $abc < 6$  y  $z(abc)=HI$  (High Impedance) si  $abc > 5$ .
  
- c. Considere la instrucción RET del CS2010. Explique con rigor:
  - a/Su operación a nivel de usuario identificando todos los registros afectados
  - b/La secuencia de microoperaciones de su ciclo de ejecución.

Problema 2. Se desea enviar el contenido de una memoria ROM de capacidad 256B mediante transmisión serie, es decir, usando una única línea de transmisión de tamaño 1 bit. Para ello, tras activar la señal  $X_s$ , el sistema esperará la activación de una señal "valT" para marcar el comienzo de la transmisión. Esta se realizará en cada ciclo de reloj mientras la señal "detieneT" no se active. Si "detieneT" se activa, detiene la transmisión hasta que dicha señal se desactive de nuevo. Al terminar la transmisión completa, el sistema activa la señal de Fin.

- a. Dé la descripción estructural del sistema digital completo como un bloque indicando entradas y salidas con el tamaño adecuado.
- b. Diseñe una **Unidad de Datos específica** para este problema. Debe describir a nivel RT todos los componentes que utilice para esta unidad.
- c. Determine la carta ASM de datos y de control del sistema.
- d. Realice la Unidad de Control del sistema mediante 1 biestable por estado.

Problema 3. En ATmega328P (AVR) se desea disponer de la subrutina **strup**. Esta subrutina:  
 \* recorre una cadena de caracteres que está escrita en la SRAM sustituyendo cada letra minúscula por su mayúscula correspondiente. Suponga que en la cadena no hay más de mil caracteres. Las cadenas sólo contienen letras mayúsculas, minúsculas o espacios en blanco (SP).

Ejemplo de conversión: una 'a' (ASCII \$61) se debe sustituir por 'A' (ASCII \$41).

\* la dirección del primer carácter está contenida en **X**. El final de la cadena viene marcado por el carácter NULL (ASCII \$00).

\* además, devuelve el número de letras que ha cambiado

Al final se muestra la tabla ASCII. Asuma que el código ASCII de cada carácter en la SRAM posee su MSB a 0 (B7 = 0). En la tabla ASCII puede comprobar que los códigos de las letras mayúsculas varían entre \$41 y \$5A, y el de las minúsculas, entre \$61 y \$7A. El espacio en blanco (SP) tiene el código \$20.

- a. Realice un diagrama de flujo (organigrama) que describa el algoritmo de la subrutina **strup**.
- b. Escriba el código ensamblador del AVR de la subrutina **strup**.

B3 B2 B1 B0	B6 B5 B4							
	000	001	010	011	100	101	110	111
0000	NULL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

# ESTRUCTURA DE COMPUTADORES – GII – IC

## Examen Final Convocatoria 1

5-jun-2023

### Solo EP1

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Se desea enviar el contenido de una memoria ROM de capacidad 256B mediante transmisión serie, es decir, usando una única línea de transmisión de tamaño 1 bit. Para ello, tras activar la señal  $X_s$ , el sistema esperará la activación de una señal “valT” para marcar el comienzo de la transmisión. Esta se realizará en cada ciclo de reloj mientras la señal “detieneT” no se active. Si “detieneT” se activa, detiene la transmisión hasta que dicha señal se desactive de nuevo. Al terminar la transmisión completa, el sistema activa la señal de Fin.

- Dé la descripción estructural del sistema digital completo como un bloque indicando entradas y salidas con el tamaño adecuado.
- Diseñe una **Unidad de Datos específica** para este problema. Debe describir a nivel RT todos los componentes que utilice para esta unidad.
- Determine la carta ASM de datos y de control del sistema.
- Realice la Unidad de Control del sistema mediante 1 biestable por estado.

Problema 2. Considere memorias de 32 kB, de 16 kB y de 8 kB con selección de chip activa en H. (En este problema no importa si son RAM o ROM). Con ellas se montará un sistema de memoria para un procesador cuyo espacio lógico es 1 MB y que tiene como terminales un bus de direcciones  $AB = A_k:A_0$ , uno de datos no segmentado y es de 8 bits,  $DB = D_7:D_0$  y uno de control CB.

Construya 56 kB de memoria en posiciones consecutivas a partir de  $AB=1000\dots_2$ .

- Indique el mapa de memoria y el circuito de decodificación. Debe identificar cómo están conectadas todas las líneas de las memorias con las del procesador (debe emplear minúsculas,  $a_i, d_i$  para las líneas internas de cada chip y mayúsculas,  $A_i, D_i$  para las del procesador).
- Obtenga la dirección lógica que permita acceder a la palabra  $\$4F$  de cada uno de los chips de memoria utilizados.
- Para la dirección lógica  $\$8C430$ , indique el chip seleccionado y la dirección interna a la que se accede.

## ESTRUCTURA DE COMPUTADORES – GII – IC

### Examen Final Convocatoria 1 (Parcial2)

5-jun-2023

**Apellidos, Nombre:** \_\_\_\_\_

**Problema 1.** En ATmega328P (AVR) se desea disponer de la subrutina **strup**. Esta subrutina:

- Recorre una cadena de caracteres que está escrita en la SRAM sustituyendo cada letra minúscula por su mayúscula correspondiente. Suponga que en la cadena no hay más de mil caracteres. Las cadenas sólo contienen letras mayúsculas, minúsculas o espacios en blanco (SP). Ejemplo de conversión: una 'a' (ASCII \$61) se debe sustituir por 'A' (ASCII \$41).
- La dirección del primer carácter está contenida en **X**. El final de la cadena viene marcado por el carácter NULL (ASCII \$00).
- Además, devuelve el número de letras que ha cambiado

Al final se muestra la tabla ASCII. Asuma que el código ASCII de cada carácter en la SRAM posee un MSB a 0 (B7 = 0). En la tabla ASCII puede comprobar que los códigos de las letras mayúsculas varían entre \$41 y \$5A, y el de las minúsculas, entre \$61 y \$7A. El espacio en blanco (SP) tiene el código \$20.

- a. Realice un diagrama de flujo (organigrama) que describa el algoritmo de la subrutina **strup**.
- b. Escriba el código ensamblador del AVR de la subrutina **strup**.

	B6 B5 B4							
B3 B2 B1 B0	000	001	010	011	100	101	110	111
0000	NULL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL



**Problema 2.** Se desea añadir al CS2010 una nueva instrucción, **SWAP Ra,Rb**.

Esta instrucción intercambiará los contenidos de ambos registros: **Ra ↔ Rb**.

- 2.1. Proponga un código máquina adecuado para esta instrucción. Muéstrela en binario y en hexadecimal para el caso concreto SWAP R6, R4.
- 2.2. Indique la secuencia de micro-operaciones para el ciclo de búsqueda (*Fetch*) y el de ejecución (*Execute*), indicando además todas las señales de control que se activan durante el proceso. Si necesitara modificar la unidad de datos, explique cómo lo haría, describiéndolo adecuadamente.

# ESTRUCTURA DE COMPUTADORES – GII – IC

## Examen Final Convocatoria 2 13-julio-2023

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

1.1. Para una descripción **verilog** de un **testbench** con “**timescale 1ns / 1ps**” y que ya tiene definidas las variables **tbCLK**, **tbS1** y **tbS2**, debe incorporar los fragmentos para realizar cada una de las dos siguientes tareas:

1.1.1. Señal **tbCLK** (reloj) de 1 MHz.

1.1.2. En la sección “**initial**”:

\* Inicie **tbCLK** a 1 y ponga dos señales, **tbS1** y **tbS2**, a 0.

\* Seguidamente espere 2 flancos negativos del reloj y ponga a 1 ambas señales.

\* A continuación y tras 10 flancos negativos del reloj, ponga a 0 la señal **tbS1**

\* Después de 3 flancos negativos nuevos termina el **testbench**.

1.1.3. Indique los valores finales de las señales **tbS1** y **tbS2**.

1.2. Considere las siguientes instrucciones del Atmega328P:

LD R20,Z+

LD R20,-Z

LDS R20,\$200

LDD R20,Z+1

OUT 17,R20

MOV R20,R25

IN R20,25

LDI R20,25

1.2.1. Compare las ocho instrucciones y explique sus similitudes y diferencias.

1.2.2. Supuestos los valores iniciales incluidos en la “Hoja para examen”, dé en cada una de ellas los valores finales de los registros y posiciones de memoria afectados.

1.3. Sea un sistema digital con entrada **Din** de 8 bits y registro síncrono **Reg[8]** con señales LD y SR (LD: carga en paralelo, SR: desplazamiento a derecha) cuyas salidas se muestran por el bus de salida **Dout[8]**.

1.3.1. Describa estructural y funcionalmente Reg a nivel RT.

1.3.2. Considere la siguiente información para los ciclos del reloj, **clk**:

ciclo 3)           Reg ← Din

ciclo 4)           Reg ← Reg

ciclo 5)           Reg ← SHR(Reg, Reg<sub>7</sub>)

Escriba la secuencia de datos en Reg en los ciclos representados por **clk** (donde los ciclos de reloj están separados por “|”) si:

\* en el instante inicial el dato contenido por Reg es \$B6

\* Din es Din = \$7C durante los ciclos 3 al 5

\* en los restantes ciclos Reg no opera

**clk:** ... 2       |   3       |   4       |   5       |   6 ...

Problema 2. Se desea intercambiar los datos contenidos entre dos registros internos del CS2010 Ra y Rb:  $Ra \leftrightarrow Rb$ . Debe resolver dicho intercambio tanto como programador del CS2010 original (cuestión 1) como diseñador (cuestión 2):

2.1. Como programador en ensamblador del CS2010.

2.1.1. Escriba un programa en ensamblador del CS2010 que realice el intercambio entre dos registros Ra y Rb sin cambiar el valor final de ningún otro registro R0:R7 salvo los antes mencionados Ra y Rb. Debe dar tanto un organigrama del algoritmo de su solución como el correspondiente código ensamblador.

2.1.2. Utilizando los datos de "Hoja de Examen", indique la secuencia de valores en los dispositivos afectados.

2.2. Como diseñador del computador. Se desea incorporar al CS2010 una nueva instrucción, SWAP Ra,Rb. Esta instrucción intercambiará los contenidos de ambos registros.

2.2.1. Proponga un código máquina adecuado para esta instrucción. Muéstrela en binario y en hexadecimal para el caso concreto SWAP R6, R4.

2.2.2. Indique la secuencia de micro-operaciones para el ciclo de búsqueda (*Fetch*) y el de ejecución (*Execute*), indicando además todas las señales de control que se activan durante el proceso. Si necesitara modificar la unidad de datos, explique cómo lo haría, describiéndolo adecuadamente.

Problema 3. Considere memorias de 32 kB, de 16 kB y de 8 kB con selección de chip activa en H. (En este problema no importa si son RAM o ROM). Con ellas se montará un sistema de memoria para un procesador cuyo espacio lógico es 1 MB y que tiene como terminales un bus de direcciones  $AB = A_k:A_0$ , uno de direcciones  $DB = D_j:D_0$  y uno de control CB.

- 1.1. Suponiendo que **DB no está segmentado y es de 8 bits,  $DB = D_7:D_0$** , construya **56 kB** de memoria en posiciones consecutivas a partir de  $AB=1000..._{(2)}$ .
  - 1.1.1. Indique el mapa de memoria y el circuito de decodificación. Debe identificar cómo están conectadas todas las líneas de las memorias con las del procesador (debe emplear minúsculas,  $a_i, d_i$  para las líneas internas de cada chip y mayúsculas,  $A_i, D_i$  para las del procesador).
  - 1.1.2. Obtenga la dirección lógica que permita acceder a la palabra  $\$4F$  de cada uno de los chips de memoria utilizados.
- 1.2. Suponiendo que **DB es ahora de 16 bits,  $DB = D_{15}:D_0$ , y está segmentado** en Bytes pares ( $D_{15}:D_8$ ) (asociado a la señal BE1) y Bytes impares ( $D_7:D_0$ ) (a la señal BE0) construya **56 kW** ( $1W=16$  bits) de memoria en posiciones consecutivas a partir de  $AB=1000..._{(2)}$ .
  - 1.2.1. Indique el mapa de memoria y el circuito de decodificación. Debe identificar cómo están conectadas todas las líneas de las memorias con las del procesador.
  - 1.2.2. Obtenga la dirección lógica que permita acceder a la palabra  $\$4F$  de cada uno de los chips de memoria utilizados.

**ESTRUCTURA DE COMPUTADORES – GII – IC**  
**Examen Final Convocatoria 3**

**25-octubre-2023**

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- a. Un dispositivo de memoria **M** de **64 kB** posee selección de chip CS activa en H, líneas de dirección físicas que se llamarán  $a_k:a_0$  y líneas de datos,  $d_j:d_0$ . La memoria de un procesador no segmentado con AB[20] y DB[8], activa M a partir de la dirección **\$60000** sin ocupar más espacio lógico que el que ocupa M físicamente. Indique con rigor:
- 1/ Dé los **valores** de “k” y de “j” y dibuje el **mapa de memoria**.
  - 2/ **Indique todas las conexiones** entre el procesador y M.
  - 3/ **Las direcciones lógicas** (en hexadecimal) para acceder a M(\$CA) y para acceder a M(11...11<sub>(2)</sub>). Indique también las direcciones físicas accedidas por AB=\$67 y por AB=\$6789A.

- b. El código Verilog adjunto describe un registro con carga en paralelo que posee señal de puesta a 0 asíncrona activa en alta y salida de control que detecta cero. Contiene **5 errores**, cuatro de carácter sintáctico y uno, funcional. **Encuéntrelos y corrijalos**.

```
// Código con 5 errores
module REG32
    input clk, wr, reset,
    input [31:0] din,
    output [31:0] dout, output z);
wire [31:0] q;
assign z = ~|q;
always @(posedge clk, negedge reset)
if (reset==1 )
    q <= 0
    else if (write)
        q <= din;
assign dout=q;
endmodule
```

- c. En el instante actual y para el computador simple CS2010, PC=\$32 y termina la ejecución de una instrucción. Se sabe que MEMCOD(\$31)=\$9800, MEMCOD(\$32)=\$1169, MEMCOD(\$33)=\$2043,...
- c1/ Indique en ensamblador la instrucción que se ejecutará seguidamente y especifique formalmente qué hace dicha instrucción. Indique el destino modificado dando, además, el valor final que almacena.
  - c2/Para dicha instrucción, desarrolle las micro-operaciones de ambos ciclos, *Fetch* y *Execute*

Problema 2. Un sistema digital incorpora una memoria Mem de 256 B. Se desea que el sistema escriba un lote de 16 datos tamaño Byte que recibe por un bus de entrada Din a partir de posiciones consecutivas a una dirección dada que también se recibe por Din. La operación del sistema es tal como se explica a continuación:

- \* Xs da la orden de comienzo cuando vale 1 durante un solo ciclo de reloj.
- \* La dirección del primer dato del lote aparece en Din cuando se active una señal de validación de direcciones, Vdir.
- \* Después y cuando se active una señal de validación de datos, Vdat, aparecerá en Din uno por uno cada dato del lote en cada ciclo de reloj.
- \* Esos datos hay que escribirlos en la memoria. Tras terminar, el sistema activará FIN.

- 2.1. Dé la descripción estructural del sistema digital global.
- 2.2. Diseñe una unidad de datos UD específica para esta aplicación. Será razonablemente simple y deberá describir adecuadamente los componentes que utilice.
- 2.3. Escriba la carta ASM de datos y de control.
- 2.4. Diseñe la unidad de control UC mediante la técnica de un biestable por estado.
- 2.5. Responda justificadamente los cambios que tendría que hacer si:
  - 2.5.1. Los lotes de datos fueran de 64 bits.
  - 2.5.2. La memoria fuera de 4kB

*(Asuma que la dirección recibida permitirá que se escriba todo el lote de datos en memoria).*

Problema 3. En el contexto del AVR se debe realizar la subrutina **vecSumPar**, la cual debe sumar la componente de un vector **VX** con la correspondiente de otro vector **VY** si el valor de la componente **VX** es un número par, mientras que, en el caso contrario (valor impar), debe restar ambas componentes. La subrutina recibe en el registro **X** la primera dirección del vector **VX**, en el registro **Y** la del vector **VY** y la dimensión de los vectores está dada por una constante **long**. Las componentes de los vectores son enteros sin signo de 16 bits.

El resultado de la operación se almacena en un vector resultado **VR** cuya primera dirección de memoria se indicada por la variable **dir\_res**.

Además, la subrutina deberá contemplar si ocurre un error de desbordamiento al operar entre **VX** y **VY**. Para ello destinará al registro **R0** como testigo de si ha ocurrido error de forma que valdrá 0 si no ocurre error y valdrá \$FF si ha ocurrido algún error. En todo caso la subrutina completará las operaciones para todas las componentes con los resultados obtenidos, sean erróneos o correctos.

- a. Realice un diagrama de flujo que describa el funcionamiento del algoritmo que va a emplear para la subrutina **vecSumPar**.
- b. Escriba un programa completo en ensamblador del AVR. Este programa incluirá la llamada a **vecSumPar** así como el código ensamblador de dicha subrutina.

## ESTRUCTURA DE COMPUTADORES – GII – IC

### Examen Final Convocatoria 1

03-jun-2024

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- a. Una memoria RAM es de 512 B y posee como contenido los valores indicados en las Hojas para examen. Indique:
  - a.1/ Nombre y dimensión de las líneas físicas de direcciones y de datos.
  - a.2/ Si es posible, las direcciones (en bin y en hex) que hay que poner en “a” para que  $M(a) = \$73$ .
  - a.3/ Si es posible, a qué se accede en  $M(\$ABA)$ .
- b. Describa en verilog un biestable JK disparado por el flanco negativo que posea una señal asíncrona de puesta a cero activa en alto (CL). Debe tener salidas  $q$  y  $\overline{q}$ .
- c. En el CS2010 se acaba de ejecutar la instrucción  $ADD R0, R7$ . Tras ejecutarse, los *flags* son **VN<sub>Z</sub>C = 0001**.
  - c.1/ Si los datos R0 y R7 son números sin signo, indique justificadamente si el resultado ha sido correcto o no.
  - c.2/ Si los datos R0 y R7 son números con signo, indique justificadamente si el resultado ha sido correcto o no.
  - c.3/ En caso de resultado correcto, proporcione el intervalo de valores posibles de dicho resultado.
- d. Realice un programa en Atmega328P que ejecute  $M(\$234) \leftarrow M(\$105) \times M(\$120)$  si los datos son números sin signo.



Problema 2. Un sistema digital incorpora una memoria Mem de 256 Bytes en la que se desea almacenar 16 datos tamaño Byte que se escribirán en posiciones consecutivas a partir de una dirección de comienzo. Tanto esa dirección de comienzo como los 16 datos, los recibirá por un bus de entrada Din tal como se explica a continuación.

El sistema comenzará tras la activación de Xs. Primero, en Din aparecerá la dirección de comienzo cuando se active una entrada de validación de direcciones, Vadr = 1. Después, cuando se active una entrada de validación de datos, Vdat = 1, aparecerá en Din el valor de cada uno de los 16 datos de forma consecutiva, uno por cada ciclo de reloj, para que se escriban en la Mem. Tras terminarse de escribir los 16 datos en Mem, el sistema generará FIN y parará.

- a. Dibuje como bloque el sistema digital identificando sus entradas y salidas.
- b. Diseñe una unidad de datos específica para este problema.
- c. Escriba la carta ASM de datos y de control.
- d. Diseñe la Unidad de Control.

Problema 3. Se tiene un sistema basado en un procesador que maneja datos de tamaño  $W$  y  $B$  (par e impar) en un espacio de direccionamiento lógico de 16MB. Los buses son  $DB[16]$ ,  $AB[23]$  y señales  $BE_1BE_0$ . Implementado con chips de memoria RAM de  $256K \times 8$  con señal de selección activa a nivel bajo, el sistema tiene construido la siguiente memoria principal:

- $256kW$  a partir de la dirección 0.
  - $256kW$  a partir de la dirección  $\$8\ 0\ 0\ 0\ 0\ 0$ .
  - $256kW$  en las últimas direcciones ( $\$F\ F\ F\ F\ F$  y anteriores)
- a) Obtenga el mapa de memoria y el sistema de decodificación correspondiente, e indique las conexiones de los chips de memoria con los buses  $AB$  y  $DB$  del procesador.
- b) Responda a las siguientes preguntas sobre accesos a datos en memoria
- 1) Para la dirección lógica  $AB_{lógica}[24] = \$FC0123$ , indique la palabra física (debe identificar el chip y dar en hexadecimal la dirección física) e identificar el tamaño del dato y las líneas del  $DB$  que lo transportan.
  - 2) Repita si  $AB_{lógica}[24] = \$FC0122$  y el tamaño del dato es  $W$ .
  - 3) Indique qué dirección lógica ( $AB_{lógica}[24]$ ) hay que poner para acceder a la palabra  $\$123$  de cada una de las memorias.

## SOLAMENTE EP2

Considere que opera con el ATmega328PA y dispone de una **subrutina** (*jun3*) que realiza una operación sobre R5. [Concretamente,  $R5 \leftarrow \text{jun3}(R5)$ , sin que se modifique ningún otro de los registros de propósito general].

Sea un **vector** **V** de dimensión **DIM** ( $DIM < 255$ ) cuyas componentes son de 1 Byte. El vector **V** está almacenado en la memoria de datos en posiciones consecutivas desde la dirección **Vdir**. Debe obtener un programa que genere **otro vector** (**W**) donde cada componente sea el “**jun3**” de la correspondiente componente de **V**, eso es:  $\forall k, W(k) \leftarrow \text{jun3}(V(k))$ . El vector **W** debe almacenarse a partir de la dirección **Wdir**.

- a) Escriba en ensamblador **mediante directivas** la parte correspondiente para dar los valores  $DIM = \$39$ ,  $Vdir = \$200$  y  $Wdir = \$400$ .
- b) Dé un organigrama del programa para obtener  $W(k) \leftarrow \text{jun3}(V(k))$  para todo  $k$ .
- c) Escriba en ensamblador dicho programa.

# ESTRUCTURA DE COMPUTADORES - GII - IC

## Examen Final Convocatoria 2

10-jul-2024

**Apellidos, Nombre:** \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

a. Sea la descripción verilog que se da debajo, que contiene tres módulos. Dé la descripción formal RT (tanto estructural como funcional):

a.1/ De type1.

a.2/ De type2.

a.3/ De type3.

```
//declaración del módulo type1
module type1 #(parameter width=8, initial_value=0)
  (input wire W, ck, input wire [width-1:0] IN, output reg [width-1:0] OUT=initial_value);
  always@(posedge ck)
    if(W) OUT<=IN;
endmodule

//declaración del módulo type2
module type2 #(parameter width=8, initial_value=0)
  (input wire W, R, ck, input wire [width-1:0] IN, output wire [width-1:0] OUT);
  wire [width-1:0] internal_bus;
  type1 #(width,initial_value) internal_reg(W, ck, IN, internal_bus);
  assign OUT = R ? internal_bus : 'bz;
endmodule

//declaración del módulo type3
module type3 #(parameter width=8, initial_value=0)
  (input wire W, R, ck, inout wire [width-1:0] DAT);
  type2 #(width,initial_value) internal_register(W,R,ck,DAT,DAT);
endmodule
```

b. Considere la calculadora de 8 registros diseñada en teoría, cuya UD está en la pág. 4 de las Hojas de examen). Considere que inicialmente los registros tienen los datos indicados en dichas Hojas de examen. Se ha ejecutado la siguiente secuencia de control en ciclos de reloj.

b.1/ Describa a nivel RT la **secuencia de microoperaciones** que realizan.

b.2/ Indique a nivel RT la **operación global** que se realiza.

b.3/ Pinte un **cronograma** con identificando claramente cada ciclo de reloj y los **datos** contenidos en los registros afectados (indique claramente el ciclo en el que cambie).

ciclo	W	D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	F <sub>2</sub> F <sub>1</sub> F <sub>0</sub>	P <sub>1</sub> P <sub>0</sub>
1	1	0 1 0	1 0 1	1 1
2	0	0 1 0	1 0 1	0 0
3	1	1 0 1	1 0 0	1 1
4	1	1 0 0	0 1 1	1 1
5	0	0 0 1	1 0 0	1 0
6	1	0 1 1	0 1 0	1 1

Problema 2. Se desea añadir al CS2010 una nueva instrucción, CPSE Rd,N, donde N es un dato inmediato que corresponde a un número sin signo. La instrucción (CPSE: *ComPare and Skip if Equal*) compara Rd con N y si son iguales se salta la instrucción escrita tras CPSE Rd,N. P.ej., si el programa es:

```
...  
CPSE Rd, N  
Instrucción A  
Instrucción B  
...
```

Tras la ejecución de CPSE, si son iguales, sigue por Instrucción B y si son distintos, continúa por Instrucción A.

a. Indique:

- i. Un código de instrucción válido para esta instrucción, sin eliminar ninguna de las existentes.
- ii. Si es posible realizar esta instrucción sin modificar la Unidad de Datos o si hay que modificarla, en cuyo caso, debe mostrar cómo se modifica.

b. Para la solución aportada en el apartado anterior, desarrolle la secuencia de microoperaciones en la ejecución de CPSE, indicando, además, las señales de control implicadas.

Problema 3. En un programa en ensamblador de AVR se tiene el siguiente código:

<pre> .include "m328Pdef.inc" .dseg .org \$100 Dato1: .byte 0x100 Dato2: .byte 0x100 Result: .byte 0x100  .cseg .org 0         jmp programa  .org \$40 programa: ldi xh, high(Dato1)           ldi xl, low(Dato1)           ldi yh, high(Dato2)           ldi yl, low(Dato2)           ldi zh, high(Result)           ldi zl, low(Result)  // Sigue en (*) </pre>	<pre> //      (*)           call exam ;           brge G1b           st z+, r2           st z+, r3 fin:     rjmp fin G1b:    st z+, r0           st z+, r1           rjmp fin  // exam:   ld r0, x+           ld r1, x+           ld r2, y+           ld r3, y+           cp r1 ,r3           brne Volver           cp r0, r2 Volver: ret </pre>
---	--

- a) Indique los registros internos (R0:R31) y valores de memoria que se modifican al ejecutar el programa completo considerando los datos iniciales siguientes:
- a.1.  $M(\$100)=\$23$ ,  $M(\$101)=\$45$ ,  $M(\$102)=\$67$ ,  $M(\$103)=\$89$ ,...  
 $M(\$200)=\$F0$ ,  $M(\$201)=\$12$ ,  $M(\$202)=\$34$ ,  $M(\$203)=\$56$ ,...  
A partir de  $M(\$2B0)$  todos son 0.
  - a.2. Repita lo anterior si cambia un solo valor a:  $M(\$101)=\$12$ .
- b) Indique qué operación hace el programa anterior y sobre qué tipo de datos especificando, al menos, su tamaño y si es o no un número (sin/con signo).  
Si hubieran dos vectores de 20 componentes cada uno (cada componente del mismo tipo que el programa original), ubicados a partir de las posiciones Dato1 y Dato2, modifique el código para que se escriba el vector resultado a partir de Result

# ESTRUCTURA DE COMPUTADORES – GII – IC

## Examen Final Convocatoria 3

24-octubre-2024

Apellidos, Nombre: \_\_\_\_\_

Problema 1. Responda a cada pregunta por separado.

- a. Interprete el siguiente código Verilog y describa este módulo a nivel RT (estructural y funcionalmente):

```
module caso1a #(parameter NN=32)
  (input ck, cl, ld, rd, input [NN-1 : 0] xin, output [NN-1 : 0] zout);
  reg [NN-1 : 0] est;
  always @ (posedge ck, negedge cl)
    if (cl==0) est <= 0;
    else if (ld) est <= xin;
  assign zout = rd? est : 'hZ;
endmodule
```

- b. Considere el fragmento de *testbench* de abajo, construido para probar un contador síncrono ascendente con señales de *up* y de *reset* síncrona disparado por el flanco de subida de *clk*. Dibuje las señales *tb\_clk*, *tb\_reset* y *tb\_up*, y el valor del contador desde  $t=0$  a  $t=500ns$ :

```
`timescale 1ns / 1ps
module ...
... ..
always
  begin
    #25;
    tb_clk = ~tb_clk;
  end

initial
  begin
    tb_clk = 0;
    tb_up = 0;
    tb_reset = 0;
    @(negedge tb_clk)
    tb_reset = 1;
    repeat(2)
    @(negedge tb_clk);
    tb_reset = 0;
    tb_up = 1;
    repeat(6)
    @(negedge tb_clk);
    tb_up=0;
    repeat(6)
    @(posedge tb_clk);
    $finish;
  end
... ..
```

- c. Sea un procesador segmentado con DB[16] y un espacio de memoria de 16MB. Dispone de instrucciones como la instrucción **CLR.tam dir** donde **tam** puede ser B o W y **dir** es la dirección lógica del dato que borra (**Clear**). Su sistema de memoria posee, entre otras, una memoria M3 de 2 MB situada en los Bytes impares y que se selecciona cuando **A23=1** y **A22=0**. Si se quiere borrar la palabra M3(\$124) mediante CLR.B dir, obtenga el valor hexadecimal de “dir”.

Problema 2. Se quiere diseñar un sistema digital que reciba datos por un bus de entrada 8 bits Din para averiguar cuántos de ellos superan un número NUM. Todos los datos son números sin signo de 8 bits. El sistema operará de acuerdo con lo siguiente:

Tras recibir la señal de comienzo, el sistema capturará cada dato cuando se active una señal externa Dval (Dato válido). El primer dato que recibe es el número de referencia NUM y tras éste, recibirá otros 1023 datos. Las salidas SAL mostrarán el número de datos mayores que NUM.

- a. Dé la descripción estructural del sistema digital (esto es, dibujo con entradas y salidas).
- b. Proponga una unidad de datos específica para este sistema.
- c. Describa a nivel RT cada uno de los elementos de dicha unidad dimensionándolos adecuadamente.
- d. Indique la carta ASM de datos y de control.
- e. Realice la unidad de control mediante la técnica de 1 biestable por estado.



Problema 3. Para el **CS2010**:

- a. Desarrolle una subrutina llamada **Abs**, que realice  $R2 \leftarrow |R2|$ , siendo el contenido inicial de R2 un número de 8 bits con signo (Ca2). Además, escribirá en R3 un 0 si el número de R2 era positivo y 1 si era negativo.
  - a1. Explique el algoritmo mediante pseudocódigo o diagrama de flujo.
  - a2. Escriba la subrutina en código ensamblador.

A continuación, se dispone de una subrutina **MulU** que calcula el producto aritmético  $R1:R0 \leftarrow R1 \times R0$ , interpretando los datos de R1 y de R0 como números sin signo. El resultado del producto de 16 bits y en R1 se almacena el MSB del resultado y en R0, el LSB. Durante su ejecución se alteran los registros R2, R3 y R4.

- b. Escriba un programa denominado **MulS** para multiplicar 2 números de 8 bits con signo en Ca2 situados en R6 y R7. Para ello, tiene que usar las subrutinas Abs y MulU.
  - b1. Explique el algoritmo mediante pseudocódigo o diagrama de flujo.
  - b2. Escriba la subrutina en código ensamblador.

**Nota:** Se recuerda que, para cualquier número de bits de A, el Ca2(A) se puede obtener de una de las tres formas siguientes: 1) mediante  $0 - A$ ; 2) mediante  $Ca1(A)+1$ ; o 3) comenzando por el LSB, se mantiene todos los ceros y el primer 1 y se complementan los restantes.