



**FUNDAMENTOS DE COMPUTADORES**  
**Convocatoria extraordinaria - 29 de junio de 2018**

Nombre \_\_\_\_\_ DNI \_\_\_\_\_

1. (1,5 puntos) Dados los siguientes números: A=-43, B= +87

a) Exprese los dos números con el mínimo número de bits en representación en complemento a dos. Indique claramente cuántos bits necesita.

b) Efectúe las operaciones A-B y A+B en complemento a dos, representando los dos números con el mismo número de bits, indicando si existe o no desbordamiento o acarreo.

2. (3,5 puntos) Diseñar una máquina de estados con una entrada E y dos salidas X e Z.

X será 1 si E ha sido 1 en tres ciclos (no necesariamente consecutivos).

Z será 1 si E ha sido 1 en al menos los dos últimos ciclos consecutivos.

a) (1,5 puntos) Dibujar el diagrama de estados como máquina de Moore.

b) (1,5 puntos) Indicar las tablas de verdad que especifican las funciones de salida y transición de estados del sistema para una implementación con un registro contador y puertas.

c) (0,5 puntos) Dibujar el esquema de la implementación con un registro contador y puertas.

3. (3,5 puntos) La final de la liga de baloncesto NBA se ha jugado al mejor de siete partidos. Los resultados obtenidos se encuentran en un vector de enteros positivos (cada uno ocupa 32 bits) a partir de la posición de memoria RES. El formato es: primero la puntuación del primer equipo (Warriors), después la puntuación del segundo equipo (Cavaliers), después un cero, y a continuación los resultados de los siguientes partidos.

Se trata de analizar los resultados obtenidos y guardar en la variable WAR la suma de las puntuaciones de los Warriors; en la variable CAV la suma de las puntuaciones de los Cavaliers; y en la variable GAN guardar el valor 0x00000057 (W en ASCII) si han ganado los Warriors o 0x00000043 (C en ASCII) si han ganado los Cavaliers.

La estructura del programa es la siguiente:

.global start

.data

RES: .word 72, 83, 0, 102, 87, 0, 99, 87, 0, 87, 89, 0, 101, 102, 0, 99, 89, 0, 78, 77

.bss

WAR: .space 4

CAV: .space 4

GAN: .space 4

.text

start:

**Código a realizar**

FIN: B .

.end

a) (2 puntos) Codificar el programa en ensamblador ARM.

b) (0,5 puntos) Si la dirección de comienzo de la zona de datos (.data) es 0x0000C000, y las zonas de datos y de instrucciones en memoria son consecutivas indicar en qué dirección de memoria se encuentra la primera instrucción del programa y en qué dirección se encuentra la variable CAV, explicando porqué.

c) (0,5 puntos) Suponiendo que estamos utilizando una implementación multiciclo del ARM como la vista en clase indicar cuál es la frecuencia de uso para este programa de las instrucciones de:

- carga,
- almacenamiento,
- aritmético-lógicas
- salto

d) (0,5 puntos) Si la frecuencia de reloj es de 300 MHz determinar cuánto tarda en nanosegundos en ejecutarse el programa.

4. (1,5 puntos) Sobre la ruta de datos multiciclo de la figura marcar qué caminos de datos están activos al ejecutar la instrucción:

STR R2, [R3, #24] en su fase número 4.

Indicar el valor de todos los puntos de control para esta fase de ejecución.

