PROBLEMAS DE FUNDAMENTOS DE COMPUTADORES
Hoja 6

1) Pasa a notación binaria y hexadecimal las siguientes instrucciones escritas en lenguaje ensamblador
   a) ADD R0, R1, R2
   b) SUBI R4, #1, R5
   c) BGE 43
   d) BR 12
   e) AND R1, R7, R6
   f) ASR R4,R2

2) Indica cuál es el resultado de ejecutar las siguientes instrucciones, dando el contenido final de los registros y posiciones de memoria que se modifican y el valor de los indicadores de condición. Se supone para cada caso que el contenido inicial de los registros y posiciones de memoria es el siguiente:

   R1=2454h, R2=5656h, R3=FFFFh, R4=0000h, R5=0002h

   Memoria:

Dirección	contenido
00h	0339h
01h	63AFh
02h	EA00h
03h	3304h
04h	7834h
05h	54AAh

   a) SUB R1, R2, R5
   b) ADDI R3, #1, R1
   c) LOAD 3(R5), R1
   d) STORE R4, 1(R4)
   e) ASR R2,R7

3) El siguiente programa calcula el máximo común divisor de dos números “a” y “b” según el algoritmo de restas de Euclides:

```
Programa ejemplo1;
Var a=5, b=15, mcd: enteros;
Mientras (a≠b) hacer
    si (a>b) entonces a:=a-b;
          sino b:= b-a;
Fmientras
mcd:=a;
```

   a) Tradúcelo a lenguaje ensamblador de la MR
   b) Traduce el programa escrito en lenguaje ensamblador a lenguaje máquina, tanto en binario como en hexadecimal, suponiendo que se almacena a partir de la dirección 0Fh. Las variables “a” y “b” se almacenan en las posiciones 0Fh y 10h respectivamente, y la variable “mcd” en la posición 11h. La primera instrucción ejecutable del programa está en la dirección 12h.

4) Traduce el siguiente programa escrito en un lenguaje de alto nivel a lenguaje ensamblador de la MR, suponiendo que se almacena a partir de la dirección 00h. Las variables “a” y “b” se encuentran en las posiciones 00h y 01h respectivamente, y la primera instrucción ejecutable del programa está en la dirección 02h. La instrucción *swap a, b* intercambia los valores de las variables “a” y “b”.

```
Programa ejemplo2;
Var a, b: enteros;
a:=13;
b:=16;
Mientras (a>10) hacer
    a:=a-1;
    b:=b+2;
Fmientras
Si (a<b) entonces swap (a, b) sino b:= a-1; Fsi;
```

5) Traduce el siguiente programa escrito en un lenguaje de alto nivel a lenguaje ensamblador de la MR, suponiendo que se almacena a partir de la dirección 00h.

```
Programa ejemplo3;
Var a, b, c: enteros;
a:=13;
b:=1;
c:=0
Mientras (a>10) o (b<20) hacer
    Si (a<11) y (b>18) entonces c:= c+3; Fsi;
    a:=a- b;
Fmientras
```

6) La instrucción SWAP *a, b* está presente en algunos lenguajes de alto nivel. Esta instrucción intercambia los valores de las variables *a* y *b*. Escribe en lenguaje ensamblador de la MR un programa que se comporte como esta instrucción, suponiendo que las variables *a* y *b* se encuentran:

   a) La variable *a* en la posición de memoria 60h y la variable *b* en el registro R2 de la Unidad de Proceso.
   b) La variable *a* en la posición de memoria 25h y la variable *b* en la posición 1Dh

7) Escribe en lenguaje ensamblador de la MR un programa que realice la división por 2^n de un número NUM. El exponente *n* es un número positivo menor de 16. Utiliza desplazamientos para realizar la división, y supón que el número NUM se encuentra en la dirección 13h, el exponente *n* en la dirección 17h, y que las instrucciones se almacenan a partir de la dirección 2Ah.

8) Escribe en un lenguaje de alto nivel un programa que sume elemento a elemento dos vectores A y B, dejando el resultado en un tercer vector C (C[i]:=A[i] + B[i]). El tamaño de los vectores es de 10 elementos. Traduce el programa a lenguaje ensamblador de la MR, suponiendo que los vectores se almacenan a partir de la dirección de memoria 00h y el programa se almacena a partir de la dirección 80h.

9) Escribe un programa en lenguaje de alto nivel tal que dados dos vectores de 10 componentes (A(i), B(i), i = 1.. 10) calcule otros dos vectores (C(i), D(i), i = 1 .. 10) tales que:

   $C(i) = A(i)$, si $A(i) \leq B(i)$
   $C(i) = A(i)+B(i)$, en caso contrario

   y

   $D(i) = 2*C(i)$

   Escribe el programa en el lenguaje ensamblador de la MR, imponiendo que los datos se almacenan a partir de la dirección de memoria 00h y el programa a partir de la dirección de memoria 80h.