

- 1ª iteración de loop

$$R_0 = 0,1 \cdot 1000 \cdot 0,000$$

$R_1 = 0x0000 \quad C7F0$

Solo entra al bucle infinito si se carga el dato: 0x0000 DEAD en R1. Es decir, cuando R0 cargue la media palabra ubicada en la dirección de memoria: 0x1000 000A

Como antes de ejecutar el salto a "loop" suma 2 a RO, significa que el bucle se va a ejecutar 5 veces con normalidad, pero la 6ª entrará en el bucle infinito y allí se quedará

También hay que comprobar los dos saltos (que parecen if). En la primera iteración, ejecuta ambas instrucciones MOV. En las siguientes, la primera se va quedando con los valores más bajos, mientras que la segunda se queda con los más altos.

Por ello, como la última vez que se pueden realizar estos saltos es durante la 5ª iteración, comprobamos en ese momento cuáles son los valores más altos y más bajos.

Entre C7FD, GAD6, 233D, FB88 y 5E15, tendremos que 233D es el más pequeño y se almacenará en R2, mientras que FB88 se almacenará en R3.

QUESTION 1 (16 puntos)

Un microcontrolador 80C1768 contiene los siguientes datos en memoria RAM:

Decade / y	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
dec1000 000y	10	10	00	04	00	25	00	06	15	32	40	02	02	11	08	13
dec1000 001y	0	07	00	04	00	00	44	54	21	01	00	31	25	00	00	00
dec1000 002y	0	00	00	00	00	00	00	00	00	00	00	77	36	00	00	00

En estas circunstancias se ejecuta el siguiente código

$$R_1 = 0 \times 0000 \text{ GADC}$$

1.000	000	= 0x 0 00000000
1.000	001	= 0x 00000001
1.000	002	= 0x 00000002
1.000	003	= 0x 00000003

The diagram shows the following code structure with handwritten annotations:

```

loop:
  LDRB R1, [R0]
  CMP R1, #4
  BGT loop

forever:
  BQ forever
  ORP R2, R2, R1
  MOV R3, R1

if:
  JNE main
  BQ forever

if:
  JNE inc
  BQ forever

  ADDS R4, R4, R2
  BQ inc
  INC R3
  INC R4
  INC R5
  INC R6
  INC R7
  INC R8
  INC R9
  INC R10
  INC R11
  INC R12
  INC R13
  INC R14
  INC R15
  INC R16
  INC R17
  INC R18
  INC R19
  INC R20
  INC R21
  INC R22
  INC R23
  INC R24
  INC R25
  INC R26
  INC R27
  INC R28
  INC R29
  INC R30
  INC R31
  BQ forever

```

Handwritten annotations include:

- $R_1 = R_1$ with an arrow pointing to the `ORP R2, R2, R1` instruction.
- `forever` with an arrow pointing to the `BQ forever` instruction, labeled "back infinite".
- $R_2 \neq R_2$ with an arrow pointing to the `ORP R2, R2, R1` instruction.
- $R_1 \neq R_3$ with an arrow pointing to the `MOV R3, R1` instruction.

$$R_2 = C_7 F_0$$
$$R_3 = C7F0$$

Responda a los siguientes apartados

Apartado A. Complete la siguiente tabla con el contenido de los registros al finalizar la ejecución del código.

Registro	Contenido (hexadecimal)
R0	0x 1000 000 A
R1	0x 0000 DEAD
R2	0x 0000 2330
R3	0x 0000 FB88
R4	0x 0000 DEAD

R4 no es modificado durante la ejecución del programa.