

Ejercicio 4:

Dados los siguientes programas LEGv8:

- 4.1) Dar los valores finales de X0, teniendo en cuenta que inicialmente vale {X0=0, X1=10}.
- 4.2) Dada la asignación a X0, $X1 \leftrightarrow \text{acc}, i$, escribir el programa "C" equivalente donde todas las variables son de tipo long.
- 4.3) Dado que inicialmente {X1=N} ¿Cuántas instrucciones LEGv8 se ejecutan?
- 4.4) Para el programa de la derecha. Si reemplazamos B.LE done por B.MI done ¿Cuál es el valor final de X0 suponiendo que inicialmente {X0=0}?
- 4.5) Dada la asignación a X0, $X1 \leftrightarrow \text{acc}, i$, escribir el programa "C" equivalente del punto "4.4", donde todas las variables son de tipo long.

```
loop: ADDI X0, X0, #2          // X0 = X0 + 2
      SUBI X1, X1, #1          // X1 = X1 - 1
      CBNZ X1, loop           // si X1 ≠ 0, salto a loop
done:
```

4.1)
Si X0=0 y X1=10, al final de la ejecución X0 = 20

```
4.2)
do {
    acc += 2
    i--
} while (i != 0)
```

4.3)
Si X1=N, se ejecutan N*2 instrucciones LEGv8

```
loop: SUBIS X1, X1, #0          // FLAGS = X1 - 0
      B.LE done                // si X1 ≤ 0, salto a done
      SUBI X1, X1, #1          // X1 = X1 - 1
      ADDI X0, X0, #2          // X0 = X0 + 2
      B loop                   // salto incondicional a loop
done:
```

4.1)
Si X0=0 Y X1=10, al final de la ejecución X0 = 20

```
4.2)
for (long i=N; i ≤ 0; i--) {
    acc += 2
}
```

4.3)
Si X1=N, se ejecutan N*2 instrucciones LEGv8

4.4)
Si cambiamos B.LE por B.MI (branch on minus), entonces ejecutamos el loop hasta que X1 sea negativo, entonces al final de la ejecución X0 = 22

```
4.5)
for (long i=N; i < 0; i--) {
    acc += 2
}
```