

# AIC Ejercicio Tema 3

Carlos Cerdá Díaz  
{ccd8}@alu.ua.es

Universidad de Alicante 2011

## Problema

Para la suma de contenidos de un vector

```
void  
suma (int v[] , int vlength)  
{  
    int retorno = 0;  
    for (int i=0;i<vlength;i++)  
    {  
        retorno+=v[i];  
    }  
    return retorno;  
}
```

Se pide:

1. Traducir a código ensamblador
2. Desenrollar hasta cuatro iteraciones
3. Introducir la función en un procesador VLIW con dos slots de emisión y latencia 1

## 1. Asumimos que:

La dirección inicial de  $v$  se encuentra almacenada en  $r1$  y la dirección final en  $r2$ .

La dirección para almacenar el resultado se encuentra en  $r3$ .

```

                lw    r10, #0           ;retorno = 0
loop:          lw    r11, 0(r1)        ;obtenemos v[i]
                add   r10, r10, r11     ;sumamos
                addi  r1, r1, #4        ;incremento del puntero
                bne   r1, r2, loop      ;Si no hemos llegado al final saltamos
                sw    r3, r10          ;dejamos el resultado

```

## 2. Asumimos que la longitud del vector es múltiplo de 4.

```

                lw    r10, #0           ;retorno = 0
loop:          lw    r11, 0(r1)        ;obtenemos v[i]
                add   r10, r10, r11     ;sumamos
                lw    r12, 4(r1)        ;siguiente elemento desenrollado
                add   r10, r10, r12
                lw    r13, 8(r1)        ;siguiente elemento desenrollado
                add   r10, r10, r13
                lw    r14, 12(r1)       ;siguiente elemento desenrollado
                add   r10, r10, r14
                addi  r1, r1, #16        ;incrementamos el puntero en 4
                bne   r1, r2, loop      ;Si no hemos llegado al final saltamos
                sw    r3, r10          ;dejamos el resultado

```

3. La tabla de *SLOTS* para el VLIW es la siguiente asumiendo latencia 1

Ciclo	SLOT 1	SLOT 2
1	lw r10, #0	lw r11, 0(r1)
2	add r10, r10, r11	lw r12, 4(r1)
3	add r10, r10, r12	lw r13, 8(r1)
4	add r10, r10, r13	lw r14, 12(r1)
5	add r10, r10, r14	addi r1, r1, #16
6	bne r1, r2, loop	sw r3, r10