

1. **Arquitectura MIPS32** Se tiene una computadora MIPS32 en la que el CPI para los diferentes tipos de instrucción es la siguiente:

| Tipo                   | Cantidad de ciclos |
|------------------------|--------------------|
| Aritmético lógica      | 1                  |
| Carga o almacenamiento | 1,9                |
| Salto condicional      | 2                  |
| Salto incondicional    | 1,2                |

En esta computadora, tenemos el siguiente programa:

```
#include <stdio.h>
#define N 32
int main() {
    int suma;
    int a[N][N], b[N][N];
    suma = acum(a,b);
    printf ("Suma= %d\n", suma);
    return(suma);
}

int acum (int a[N][N], int b[N][N]) {
    register int i, j, ac;
    for (i=0;i<N;i++)
        for (j=0;j<N;j++)
            ac+=a[i][j]+b[j][i];
    return ac;
}
```

- Hacer el diagrama de stack de las funciones `acum` y `main`.
- Dar una versión en assembler de la función `acum`.
- Cuál es el CPI promedio la función `acum`?

## 2. Consideraciones de diseño de cachés

- En el diseño de un procesador, se tiene la opción de incluir un caché de 8 KB de dos o cuatro vías. El tiempo de hit para el caché de dos vías es 1.1 veces más rápido que el de cuatro vías, mientras que el miss rate para el caché de dos vías es de 4.9 % y para el de cuatro vías es de 4.4 %. Asumiendo que en los dos casos el tiempo de reloj se ajusta para que un hit tome un ciclo, que la penalidad de miss en L1 es de 10 ciclos por acceso a la caché L2 para el caso de la caché 2WSA, y que la caché L2 no tiene misses, ¿cuál de las dos opciones nos da menor tiempo promedio de acceso a memoria?
- Se tiene una máquina con una jerarquía de memoria que consta en un L1I y L1D separados y virtualmente direccionados, sin otros niveles de cache, de 8KB cada uno, no asociativos, con líneas de 32 bytes.  
Para las transferencias entre cache y memoria principal, la latencia es de 50 ns para lectura y escritura, y el ancho de banda es de 1 GB/s. Además, sólo puede haber una transferencia pendiente por desacierto.  
Encontrar una secuencia de accesos a memoria (words de 32 bits) expresada en direcciones de bytes, para la cual la tasa de desaciertos sea mayor para WB/WA que para WT/~WA.
- Con los parámetros del ejercicio anterior, encontrar una secuencia de escrituras a memoria exclusivamente para la cual el procesador pase menos tiempo detenido para el caso WT/~WA que para el caso WB/WA.

3. **Memoria virtual.** Supongamos un sistema de memoria virtual que usa una tabla lineal para traducir direcciones virtuales en físicas. Para cada una de las siguientes preguntas, responda qué sucedería al modificar uno los parámetros del sistema original.

- a) Si duplicamos la cantidad de bytes de memoria física, ¿cómo cambia la cantidad de bits en cada entrada de la tabla de paginación?
- b) Similarmente, ¿cómo afecta este cambio en la cantidad de entradas de la tabla de paginación?
- c) ¿Qué sucede con la cantidad de bits de cada entrada de la tabla, si duplicamos la cantidad de bytes de memoria virtual?
- d) ¿Cómo afecta este cambio, a la cantidad de entradas de la tabla?
- e) Al duplicar el tamaño de página, ¿cómo se ve afectada la cantidad de bits de cada entrada de la tabla?
- f) ¿Qué sucede con la cantidad de entradas de la tabla?
- g) El cuadro 1 muestra las primeras 8 entradas de la tabla de paginación. Si el tamaño de página es de 1KB, ¿cuál es la dirección física correspondiente a la dirección virtual 0xf74?
- h) Similarmente, ¿cuál es la dirección física correspondiente a la dirección virtual 0x1400?
- i) ¿Qué sucede con la dirección virtual 0x1818?

| Index | Valid | Present | Frame number |
|-------|-------|---------|--------------|
| 0     | 0     | 1       | 7            |
| 1     | 1     | 1       | 9            |
| 2     | 0     | 1       | 3            |
| 3     | 1     | 1       | 2            |
| 4     | 1     | 1       | 5            |
| 5     | 1     | 0       | 5            |
| 6     | 0     | 1       | 4            |
| 7     | 1     | 1       | 1            |

Cuadro 1: tabla de páginas del ejercicio 3.

4. **Performance.** Derivar la expresión de speedup dada por la Ley de Amdahl.