

Estructura de Computadores

Ejercicio Voluntario. Optimización de código MIPS

Propuesta de ejercicio

El objetivo del ejercicio es profundizar en los matices de la ejecución segmentada. Para ello, se deberá simular en WinMIPS64 una implementación del código C mostrado a continuación (similar al del ejercicio 7):

```
double out[100] = {0.0};
double filter[4] = {0.5, 0.3, 0.15, 0.05};
double input[104] = {0.0,1.0, 2.0, 3.0,4.0,5.0,6.0,7.0,8.0,9.0,
    0.0,1.0, 2.0, 3.0,4.0,5.0,6.0,7.0,8.0,9.0,
    0.0,1.0, 2.0, 3.0,4.0,5.0,6.0,7.0,8.0,9.0,
    0.0,1.0, 2.0, 3.0,4.0,5.0,6.0,7.0,8.0,9.0,
    0.0,1.0, 2.0, 3.0,4.0,5.0,6.0,7.0,8.0,9.0,
    0.0,1.0, 2.0, 3.0,4.0,5.0,6.0,7.0,8.0,9.0,
    0.0,1.0, 2.0, 3.0,4.0,5.0,6.0,7.0,8.0,9.0,
    0.0,1.0, 2.0, 3.0,4.0,5.0,6.0,7.0,8.0,9.0,
    0.0,1.0, 2.0, 3.0,4.0,5.0,6.0,7.0,8.0,9.0,
    9.0,9.0,9.0,9.0};

int main () {
    int n,k;
    for ( n = 0; n < 100; n++ ) {
        for ( k = 0; k < 4; k++ ) {
            out[n] += filter[k] * input[n+k];
        }
    }
}
```

Para realizar las simulaciones y documentar las estadísticas de simulación correspondientes, se utilizarán la siguiente configuración de WinMIPS64 (*Configure->Architecture*):

- *Code Address Bus* = 10
- *Data Address Bus* = 10
- *FP addition latency* = 2
- *Multiplier latency* = 4
- *Division latency* = 20

Para la correcta implementación, se sugiere consultar el fichero de texto *ISET* distribuido con el simulador, así como algunos de los ejemplos incorporados (*series.asm* es uno de los más aconsejables, pues incluye casi todos los elementos necesarios para este código).

Asimismo, se recomienda ejecutar el código C en cualquier plataforma para comprobar el resultado que debe

obtenerse tras la finalización del código (por ejemplo, los tres primeros elementos del resultado deberían ser `out[0]= 0.75`, `out[1]=1.75`, `out[2]=2.75`).

Optimizaciones

Una vez conseguida la implementación directa del código propuesto en el *ejercicio 7* de la hoja de problemas, y comprobado que el resultado final es el correcto, se procederá a la **optimización** del código (y de la ruta de datos), para tratar de minimizar el tiempo de ejecución total.

Optimizaciones sugeridas:

- Habilitar/deshabilitar tanto *Delay Slot* como *Forwarding*. No está permitido modificar las latencias de las unidades funcionales.
- Reordenación de código ensamblador
- Modificaciones de código a alto nivel: transformaciones de bucles en el código C. Se recomienda buscar informaciones sobre transformaciones habituales como desenrollado de bucles (*loop unrolling*) intercambio de bucles (*loop interchange*), *unroll and jam* y otras. **Se valorará especialmente este tipo de modificaciones**, pues son las que dan lugar a mayores mejoras de rendimiento.

Cuestiones

A través de la información del simulador, contesta a las siguientes cuestiones:

- ¿Cómo se representa el número *0.5* en punto flotante de doble precisión (64 bits)? ¿Dónde has encontrado dicha información?
- Si pudieras proponer modificaciones al repertorio de instrucciones, ¿hay alguna instrucción/es que permitiera mejorar el rendimiento de esta aplicación concreta?
- Realiza una estimación, a partir del código original, del porcentaje de ciclos dedicado a instrucciones relacionadas con el control del bucle (comparaciones, saltos, actualización de contadores...)

Entrega: documentación

Para realizar la entrega, se subirá al Campus Virtual un fichero PDF que incluya la siguiente información:

- Respuestas las cuestiones planteadas en la sección anterior
- Un grafo con las dependencias (LDE - EDE - EDL) del código ensamblador inicial (el correspondiente al ejercicio 7)
- El contenido del fichero *asm* inicial, junto con las estadísticas obtenidas tras la simulación (número de ciclos, de instrucciones, CPI, motivo de las paradas del *pipeline* y tamaño del código).
- Propuestas de modificación de código, incluyendo el código final (tanto versión C como ensamblador) generado, así como las opciones de simulador (*Delay Slot*...) utilizadas. Nuevamente se incluirán las estadísticas obtenidas tras la simulación
- Discusión acerca de la idoneidad de la transformación planteada: ¿por qué tiene impacto positivo? ¿Hay aún margen de mejora?