Quiz Vector Processing/SIMD — Arquitectura de Computadores II
Prof: Ronald García — Alunmo: Jonathan Guzmán

Quiz Vector Processing / SIMD

Pregunta 1

Considere el siguiente código de ensamblador basado en RV64V que va ser ejecutado en una implementación vectorial con n = 64 y 1,25GHz de clk con un solo LANE. Determine:

a) Identifique los convoys encontrados en el extracto mostrado.
   En el codigo del extracto proporcionado podemos encontrar 4 convoys.
b) Calcule los chimes necesarios para la secuencia de instrucciones.
   - Convoy 1: 3 chimes
   - Convoy2: 1 chime
   - Convoy 3: 2 chimes
   - Convoy 4: 2 chimes
   - Total: 8 chimes
c) Calcule el Te.
   $T_e = n \times m \, cycles$
   $T_e = 64 \times 8 \, chimes$
   $T_e = 512$
d) Calcule el Total_FLOPS.
   Para este caso tenemos que solamente hay 3 operaciones con punto flotante, entonces tenemos que:
   $Total \, Flops = 3 \times 512$
   $Total \, Flops = 1536$
e) Si se desea cambiar el número de LANE = 4 que características tiene que tener la implementación para permitir una mejora en el desempeño, como puede esto influir las decisiones de diseño y programación.
   Si se realiza este cambio debería contemplarse las siguientes decisiones en cuanto a diseño y programación:
   - Procesamienton en paralelo
   - Ancho de banda de memoria
   - Latencia y sincronización
   - Optimización de código y/o algoritmos

Pregunta 3

Comparación de desempeño Intrinsics vs naive implementation.

a) Implemente en C++/Rust/C# que permita obtener el producto punto de dos arreglos tipo float de tamaño N donde N es múltiplo 4 (N > 500), mida el tiempo de ejecución sin optimización de compilador.
b) Mediante el uso de instrinsics x86 vectorice el mismo algoritmo y verifique el valor obtenido en es igual al valor obtenido en b. (indique que extensiones está empleando AVX1,2,512.., etc).
   El código utiliza intrínsecos SSE para aprovechar las instrucciones SIMD y mejorar el rendimiento del cálculo del producto punto entre dos vectores.
c) Que diferencias observa en el tiempo de ejecución, que relación existen entre el desempeño y el alineamiento de los datos en memoria.

```
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD$
 g++ -o pregunta31 pregunta3.1.cpp
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD$
 ./pregunta31
Resultado: 252.017
Tiempo promedio de ejecución sin optimización: 6.99271e-06 segundos
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD$
 ./pregunta31
Resultado: 252.017
Tiempo promedio de ejecución sin optimización: 5.90699e-06 segundos
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD$
 ./pregunta31
Resultado: 252.017
Tiempo promedio de ejecución sin optimización: 1.28631e-05 segundos
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD$
 ./pregunta31
Resultado: 252.017
Tiempo promedio de ejecución sin optimización: 7.24579e-06 segundos
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD$
```

```
Resultado: 244.392
Tiempo de ejecución con vectorización SSE: 2.763e-06 segundos
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD
$ ./pregunta32
Resultado: 251.582
Tiempo de ejecución con vectorización SSE: 5.021e-06 segundos
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD
$ ./pregunta32
Resultado: 254.849
Tiempo de ejecución con vectorización SSE: 5.375e-06 segundos
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD
$ ./pregunta32
Resultado: 251.723
Tiempo de ejecución con vectorización SSE: 4.832e-06 segundos
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD
$ ./pregunta32
Resultado: 238.676
Tiempo de ejecución con vectorización SSE: 8.597e-06 segundos
jonathan@Jonathan-MSiU:~/Documents/ArquitecturaComputadores/Quiz-VectorProcessingSIMD
$
```

Los resultados muestran que la vectorización SSE ha mejorado el rendimiento en términos de tiempo de ejecución en comparación con la versión no vectorizada.
La alineación de datos en memoria es crucial para la eficiencia de la vectorización, los intrínsecos SSE requieren que los datos estén alineados en memoria, y si no lo están, puede haber una penalización de rendimiento. En este caso, parece que los datos están alineados correctamente, ya que se obtiene un mejor rendimiento que cuando no.

Respecto a las gráficas:

- Primera: el punto máximo indica que el programa está alcanzando un rendimiento máximo de aproximadamente 204.8 GFLOPs/s cuando la tasa de transferencia de datos es de 0.1 FLOPS/BYTE, esto sugiere que, en esta configuración particular, el programa está ejecutando operaciones en paralelo de manera eficiente, lo que resulta en un alto rendimiento de GFLOPs/s.
- Segunda: en esta gráfica, el punto máximo se encuentra en una tasa de transferencia de datos de 0.2 FLOPS/BYTE, con un rendimiento de 400 GFLOPs/s, esto indica un rendimiento aún mayor en comparación con la primera gráfica.