**Christian Gabriel Lara López**

**5.4**

a) El registro %xmm0 tiene diferentes funciones en los códigos. En el compilado con -01 es usado para guardar un valor auxiliar que lee en cada iteración de dest y luego se multiplica por el siguiente valor del arreglo y guarda ese valor obtenido en dest en cada iteración. En el código compilado con -02 el registro %xmm0 es usado a modo de variable para ir acumulando el valor obtenido del producto realizado en cada iteración para al terminar las el ciclo guardar este valor en dest.

b) La versión optimizada podemos comprobar que sí se comportará igual a la función combine3 analizando su código en ensamblador, incluso usando el aliasing de memoria.

c) La versión optimizada se comportará de igual manera ya que podemos comprobar que el valor de dest va a ser igual al valor del registro %xmm0 en la iteración anterior, excepto en la primera iteración.

**5.5**

a) En este código habrán 2\*n multiplicaciones y n sumas para un polinomio de grado n.

b) Podemos comprobar en la tabla 5.12 que la latencia para la multiplicación de números de punto flotante es de 5 y que la suma es de 3. La operación que limita la velocidad por lo tanto es esta multiplicación que da esta CPE de 5.00, dado que el otro ciclo no puede empezar hasta no realizarse la misma.

**7.1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Símbolo** | **Entrada en .symtab?** | **Tipo** | **Módulo** | **Sección** |
| buf | Sí | Externo | m.o | .data |
| bufp0 | Sí | Global | swap.o | .data |
| bufp1 | Sí | Global | swap.o | COMMON |
| swap | Sí | Global | swap.o | .text |
| temp | No |  |  |  |

**7.2**

a) REF(main.1) -> DEF(main.1)

REF(main.1) -> DEF(main.1)

El símbolo fuerte lo encontramos en el módulo 1 dado que main es una función

b) Esto daría error dado que tenemos 2 símbolos fuertes de igual nombre, una función y una variable global inicializada.

c) REF(x.1) -> DEF(x.2)

REF(x.2) -> DEF(x.2)

El símbolo fuerte está en el módulo 2 que es una variable global inicializada

**7.3**

**$>**gcc p.o libx.a

**$>**gcc p.o libx.a liby.a

**$>**gcc p.o libx.a liby.a libx.a