

PRÁCTICA 1, GRUPO 3, PABLO MENDOZA

1. Rendimiento de mi equipo (PC personal, no el de laboratorio)

a. Características generales:

- i. Fabricante: OEM (A medida, por piezas)
- ii. Modelo: N/A
- iii. Sistema: Windows 10 x64
- iv. Memoria RAM: 16 GB
- v. Procesador: Intel I7 3770 Ivy Bridge @3.4 GHz
- vi. Gráficas: AMD Radeon RX 480 8GB VRAM

b. Información arrojada por QwikMark:

- i. Modelo de procesador: Intel Core i7-3770 CPU @ 3.40 GHz
- ii. Conjuntos de instrucciones soportados:
 - 1. MMX
 - 2. SSE
 - 3. SSE2
 - 4. SSE3
 - 5. SSSE3
 - 6. SSSE4
 - 7. SSSE4.1
 - 8. SSSE4.2
- iii. Velocidad teórica : 3.40 GHz (3.4E9 ciclos por segundo)
- iv. Núcleos de CPU: 8
- v. RAM: 16GB
- vi. Memoria de video 3072 Mb

c. Test de rendimiento QwikMark

- i. Velocidad medida: 3.4033489 GHz
- ii. CPU FLOPS: 83 GBFLOPS (FLOPS: Cantidad de cálculos de coma flotante por segundo)
- iii. Ancho de banda con memoria principal: 12GB/s (Referido con la memoria RAM. Este resultado expresa la cantidad de información que se puede transmitir como máximo entre la CPU y la RAM)
- iv. Transferencia con disco HDD: 132MB/s

2. Simulador de CPU

- a. Identificación de elementos:
 - i. Contador de programa: PC
 - ii. Memoria RAM: Tabla de almacenamiento de programa y datos
 - iii. Buses: Conexiones entre elementos internos
 - iv. Registro de instrucciones: IR
 - v. Unidad Aritmético-Lógica: ALU
 - vi. Registro de datos W, X, Y, Z
- b. Descripción del programa de prueba:
 - i. LOD #3: El contador de programa selecciona la orden a ejecutar. La orden se carga en el registro de instrucciones. La primera parte de esta va al decodificador y al MUX, que dice al ALU el tipo de operación a seguir. EL dígito viaja a MUX, se carga en el ALU, y termina en el acumulador. EL contador de programa se suma dos a sí mismo.
 - ii. MUL #2: El contador selecciona la instrucción, que viaja al registro de instrucciones. El operando pasa por el decodificador y el MUX y se carga en el ALU. EL 3 del acumulador se retroalimenta al ALU en el primer espacio reservado a enteros. El dígito viaja al ALU. El ALU multiplica ambos y el resultado se carga en el acumulador. El contador se suma 2
 - iii. ADD #5: El contador selecciona la instrucción, que viaja al registro de instrucciones. EL operando viaja al decodificador y al MUX y se carga en el ALU. EL 6 del acumulador se retroalimenta al ALU, en el primer espacio de enteros. EL dígito se carga en el ALU, cuya suma se almacena en el acumulador. El contador se suma 2
 - iv. STO Y: El contador selecciona la instrucción, que viaja al registro. El operando viaja al decodificador. El dígito selecciona la entrada Y de la memoria, donde el 11 del acumulador viaja para copiarse. El contador se suma 2
 - v. HLT: El contador ejecuta esta orden, que viaja al registro, donde se emite directamente un pulso al contador que se suma dos. Fin del programa.
- c. Programa ejercicio: LOD #10 / STO Z / DIV #2 / STO Y / HLT

3. Explicación de los términos del glosario:

- a. LOD: Carga un valor entero cualquiera o alguno almacenado en memoria al acumulador.
- b. STO: Almacena el valor del acumulador a una casilla de memoria a voluntad
- c. NOP: Pasar a la siguiente operación
- d. HLT: Fin de programa
- e. ADD: Suma al acumulador el valor de un registro o un entero cualquiera. El resultado va al acumulador
- f. SUB: *Ídem*. En vez de sumar, resta
- g. NOT: Niega de forma binaria el valor del acumulador.
- h. JMP: Salta a una posición de memoria a voluntad.
- i. JNZ: Si el acumulador es 0, salta como JMP, si no, se ejecuta la siguiente instrucción.
- j. CPZ: Niega el contenido de un entero o de un registro al acumulador
- k. CPL: Si el valor de un entero o de un registro es negativo, el acumulador lo pone a uno, si no, a 0.