Informe Técnico Grupal
Segundo Parcial Organización de Computadores

1. Integrantes:
    - Isabella Hernández Posada
    - Laura Andrea Castrillón Fajardo
    - Samuel Llano Madrigal
    - Samuel Martinez Arteaga

2. Descripción del algoritmo elegido para la realización de la raíz cuadrada.

Nuestra implementación de la raíz cuadrada consiste en que para todo número no negativo N, iremos probando k= 0,1,2, … Posible parte entera de N, para ello en cada iteración calculamos el valor de k² y lo compararemos con la resta diferencia= N-k^2, a partir de ello tendremos los siguientes casos:

- diferencia > 0, aún no tenemos un resultado concluyente, así que k + 1 y realizamos otra iteración
- diferencia = 0, N tiene de raíz exacta k, por ende la parte entera es la misma k
- diferencia < 0, k^2 supera N lo que significa que tomamos un valor más grande, así que la parte entera es k - 1

Observación: usamos la multiplicación únicamente para obtener k² y compararlo con **N**; no necesitamos el valor del producto para nada más que esa comparación.

Decidimos hacerlo de esta forma ya que nos inspiramos en la siguiente matriz donde la diagonal son cuadrados perfectos que junto a sus índices nos da la información con la que partimos a solucionar el problema.

0	1	2	3 …
1	1	2	3
2	2	4	6
3 …	3	6	9

Quisimos implementar que el resultado no solo se mostrará en el registro 19 sino también en la pantalla como número si el resultado es un número entre 0-9 o con el símbolo + si es mayor, para ello se itera sobre resultado-1 y lo dirige al apuntador respectivo a cada número:

- Si resultado - 1 = 0, salte al apuntador/dirección de ese número
- Si resultado - 1 !=-1 0, no es el número de la respuesta, realizar iteración. Si tras realizar 10 iteración (de 0-9) resultado != 0, el resultado es mayor, por lo que se dibuja un +

3. Diseño gráfico de las iniciales: ¿cómo las dibujaron? ¿qué efectos usaron? ¿Cuál es el proceso para mostrarlas en pantalla?

Para el diseño de las iniciales se usaron imágenes de boceto que luego se realizaron en bitmap (Nand2Tetris) y obtuvieron el código en jack, las imágenes son creativas, sin embargo no tienen algún tipo de efecto.

Al iniciar el programa todas las letras son mostradas en la pantalla. Para ubicar en la misma fila la primera tipografía de las letras que representa cada nombre (S, M, L, I) se manipula el registro R12 (puntero base en SCREEN) de tal manera que al dibujar una nueva letra R12 se incremente en 4 (R12=R12+4) moviendola en la misma fila. Una vez terminada la impresión de la primera fila se define R12 = 4136 para volver al comienzo de la fila y repetir el proceso de dibujar la segunda tipografía. En cada iteración se ejecuta un retraso saltando a la puntero llamado delay donde a partir de un bucle o iteraciones de conteo se genera un efecto de animación visible antes de dibujar la letra siguiente.

Cada vez que el programa ingresa una letra por teclado, se almacena en la memoria su valor ASCII. Para identificar a qué letra corresponde, se realizan comparaciones usando la diferencia: letra = letra_ingresada − código_ASCII_letra:

- Si letra = 0, la letra coincide con alguna de las definidas (S, M, L, I) y se salta a la etiqueta correspondiente. La letra también puede ser el código ASCII del espacio " "; de ser así, se mostrarán nuevamente todas las letras.
- Si letra ≠ 0, no coincide con esa letra y se prueba con la siguiente.

Al saltar a la letra correspondiente, se borra la pantalla y se dibujan ambas tipografías una debajo de la otra. Para ello, es necesario ajustar el registro R12 de modo que la segunda letra inicie debajo de la primera (R12 = 4136).

4. Dificultades técnicas y cómo las superaron.

Para implementar el retraso visual, tuve un problema: después de saltar a la etiqueta delay (que se invoca tras imprimir cada letra), el programa no sabía a qué instrucción debía continuar. La decisión de “qué sigue” se tomaba antes del salto, pero esa información no estaba disponible dentro de delay, así que al finalizar el tiempo de espera no había un destino claro. Definí una variable (un espacio de memoria) para guardar, justo antes de llamar a delay, el puntero a la “siguiente letra” o instrucción. Al terminar, delay lee ese puntero y salta a esa dirección.

El borrado fue un punto un poco complejo de abordar, debido a que es algo que se hace registro a registro y no lo habíamos entendido muy bien. Así mismo, cuando intentamos hacer que una letra se ponga debajo de otra fue un poco complejo. Lo logramos superar entendiendo mejor el posicionamiento de los caracteres en la pantalla utilizando el registro 12.

El simulador no trae directamente un “print” de caracteres, sino que hay que traducir el código ASCII a la dirección correspondiente en la memoria de pantalla y mapearlo a la tabla de bits de la fuente

5. Distribución del trabajo y aprendizajes clave.

- Isabella Hernández Posada: Fui responsable del diseño de las letras y números que se usaron en los programas. Me enfoqué en escoger una tipografía clara y estética, de manera que las iniciales (I, L, S, M) y los números fueran visibles y fáciles de interpretar en pantalla. Aprendí que el diseño visual en un entorno como Nand2Tetris requiere pensar en legibilidad y usabilidad, no solo en estética. También descubrí cómo trasladar una idea gráfica a coordenadas y pixeles en la memoria de la máquina, integrando la parte creativa con la lógica de programación en ensamblador.

- Laura Andrea Castrillón: Fui responsable de la lógica de raíz cuadrada y delay al aparecer todas las letras. Aprendí sobre el flujo que es necesario seguir al programar en ASM y la realización e integración de operaciones como la resta y la multiplicación que fueron necesarias. Así mismo aprendí sobre el manejo del registro 12 y el uso de las etiquetas y los saltos.

- Samuel Llano Madrigal: Fui responsable de la función que al presionar una tecla aparezca en la pantalla. Aprendí que el teclado no solo entrega letras, sino códigos numéricos, igualmente la pantalla, que es una región de memoria, y que para imprimirlos hay que relacionar esos valores con la memoria de pantalla.

- Samuel Martinez Arteaga: Fui responsable de mostrar los números del resultado de la raíz cuadrada e integración del mostrado de todas las letras al inicio y tras presionar el espacio. Aprendí sobre el borrado y actualización de pantalla, conociendo mejor el manejo del registro 12, y el control de las etiquetas y los saltos.

6. Video:
   https://drive.google.com/file/d/1oDNKOIPDLF2jt3ip_HsgD4ISL8QkfdkQ/view?usp=sharing

7. Repositorio: https://github.com/SamuelMarti22/P2_Organizaci-n_RaizyLetras
   El repositorio solo fue usado para almacenar los archivos por lo que solo hay un commit para subir los archivos y no el proceso y el trabajo de todo el equipo.