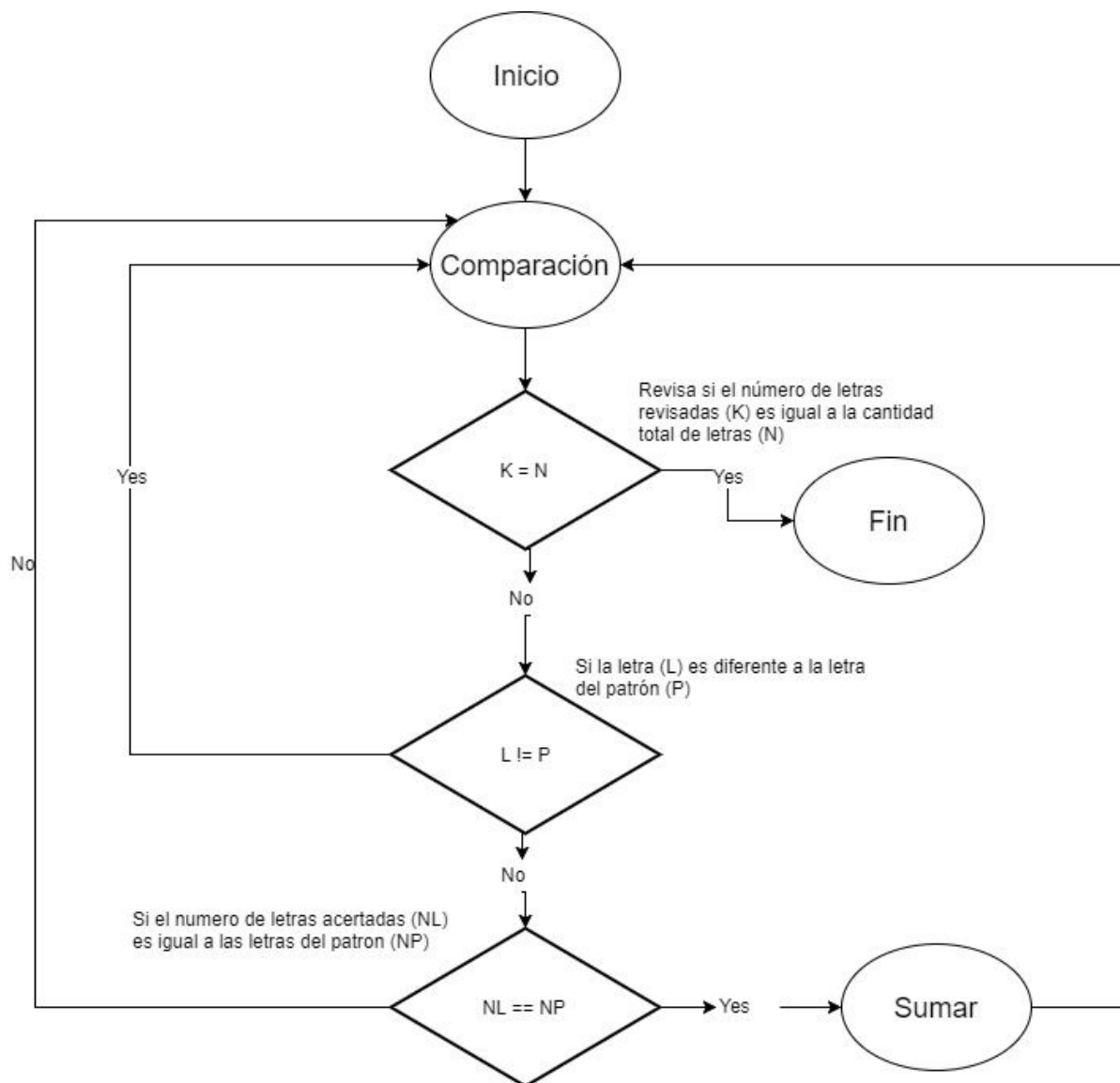


Algoritmo Fuerza Bruta



Descripción:

Fuerza bruta es un algoritmo que evalúa todas las posibilidades una a una hasta encontrar lo deseado, en este caso se implementó para el procesamiento de texto y encontrar cuántas veces aparece un patrón dentro del texto. Se implementó comparando letra con letra, en caso de que todas la letras coincidan aumenta en una unidad el número de patrones encontrados en el texto y de desplaza a la siguiente letra para evaluar para evaluar el patrón con las siguientes 11 letras. En caso que las 11 letras no coinciden solo se desplaza una letra para comparar con las 11 siguientes.

Dado un patrón y un texto de M y N caracteres respectivamente, compara el patrón con cada subcadena de texto de longitud M; tiene una complejidad de $O(MN)$.

Tabla 1. Registros Utilizados

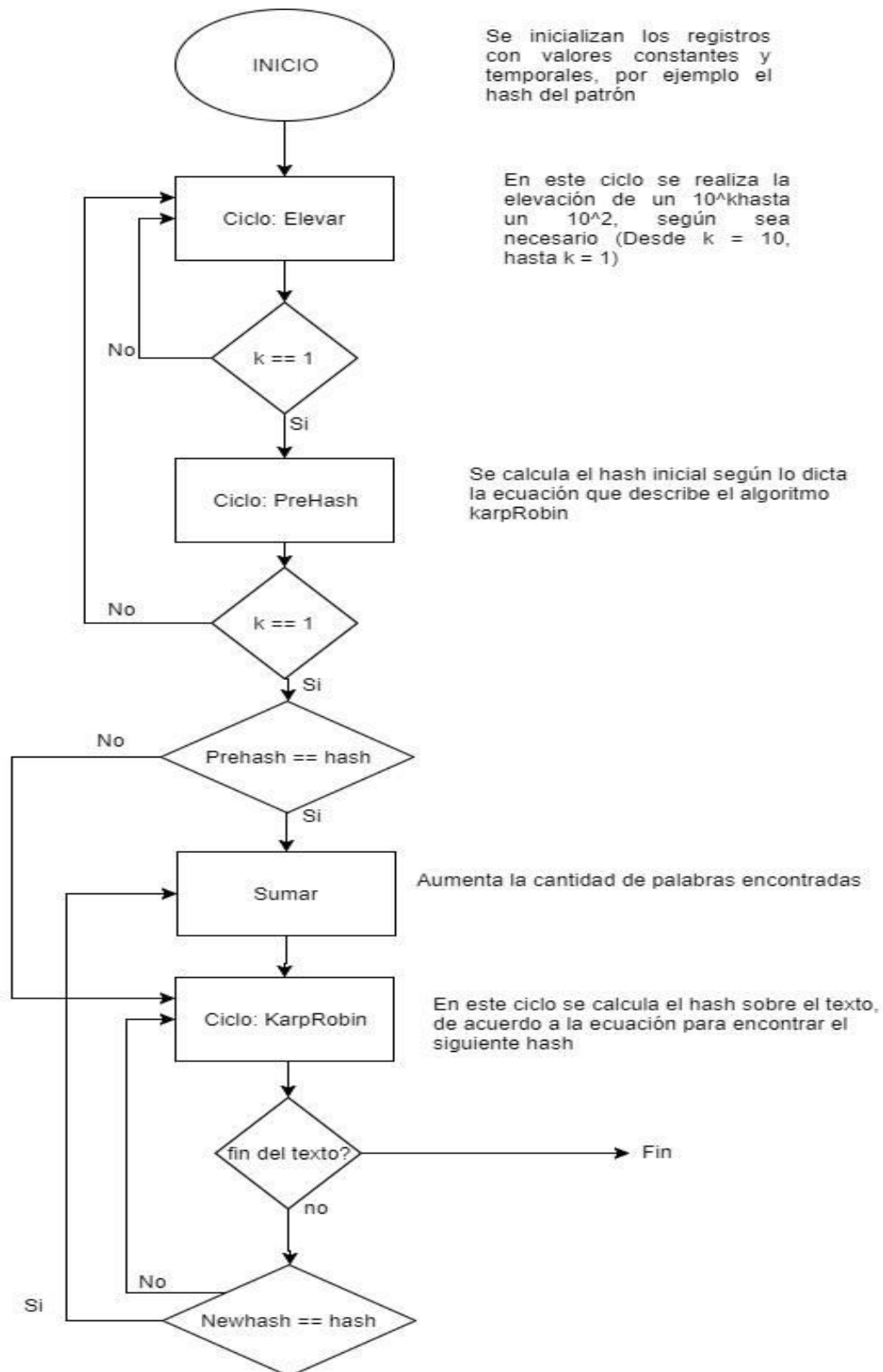
Registro	Función	Valor Inicial
a0	Contador de letras acertadas	0
a1	Contador de bytes del texto leído	0
a2	Contador de palabras	0
a3	Cantidad de letras del patrón	11
t8	Cantidad de letras en el texto	m
s5	Accesar al primer espacio de memoria a evaluar	digitado
s6	Posición inicial de memoria en el cual está almacenado el patrón	0x10010000

Tabla 2. Instrucciones Utilizadas

Bloque	Instrucción	Número de Veces Utilizadas
Inicio	addi	6
	sub	1
	add	1
Comparación	beq	2
	lb	2
	addi	4
	bne	1
	j	1
Sumar	addi	2
	andi	2
	j	1

Instrucciones utilizadas: 24

Karpt Robin



Descripción:

Karp Robin es un algoritmo utilizado para la búsqueda de patrones en un texto. Se caracteriza por tomar un grupo de caracteres del texto, este grupo debe ser de igual tamaño que el patrón, y sobre cada grupo en el texto aplicarle una función hash, y si este valor coincide con el hash del patrón, se ha encontrado una coincidencia. Lo que se considera es que el patrón tiene un tamaño de n y el texto de m , este algoritmo se ejecuta si $m > n$.

Su función de hash se da por las siguientes ecuaciones:

$$Hash_k = \sum_{i=k}^{i=n} \text{texto}[i] * 3^{\hat{m}} \quad (1)$$

$$Hash_{k+1} = 3 * (Hash_k - \text{texto}[k] * 3^{n-1}) + \text{texto}[k+1] \quad (2)$$

Tabla 1. Registros Utilizados

Registro	Función	Valor Inicial
S7	Contador de palabras	0
A0	Guarda el hash del patrón Lorem Ipsum	7741639
A1	Almacena tamaño del patrón n	11
A2	Almacena el valor de la base	3
A3	Almacena la base a la $n-1$ (3^{10})	59049
T0	Offset a partir de donde se inicia a leer	0
T1	Dirección de memoria en la cual comienza a leer	0
T2	Tamaño del texto	m
T3	Temporal de la dirección inicial de memoria	0
T4	Temporal para iterar en preHash ($n-1$)	10
T5	Temporal para iterar en elevar ($n-1$)	10
T6	Temporal utilizado en elevar, es la multiplicación de la base	3

Tabla 2. Instrucciones utilizadas

Bloque	Instrucción	Número de veces utilizada
Inicio	addi	13
	subi	3
	sub	1
Elevar	beq	1
	mul	1
	subi	1
	j	1
PreHash	lb	3
	addi	5
	mul	2
	add	3
	subi	1
	bne	1
	beq	1
KarpRobin	lb	2
	addi	2
	mul	2
	add	1
	subi	1
	Sub	1
	bne	1
	beq	1
Sumar	addi	1
	j	1

Instrucciones utilizadas: 50.

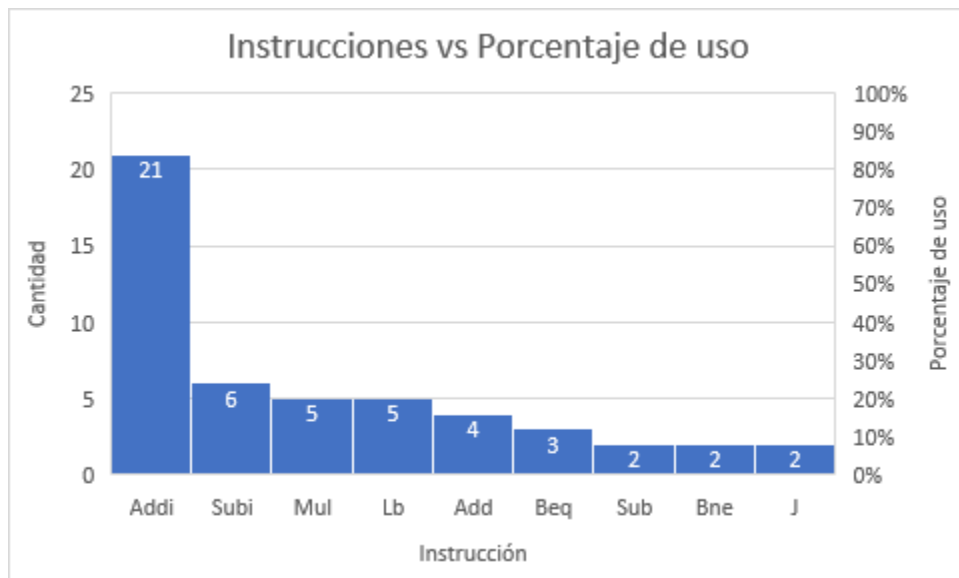


Figura 1. Instrucción vs porcentaje de uso

Bitácora

09/09/2017

Revisión del documento para entenderlo.

Análisis e investigación de cómo funcionan los algoritmos de reconocimiento de patrones.

10/09/2017

Investigación del algoritmo Kart Robin para entender cómo emplearlo en en lenguaje ensamblador. Se propuso un dos ideas para resolver el

13/10/2017

Se le presenta a Fabian la idea de cómo desarrollar el algoritmo de fuerza bruta, que al parecer no cumplía con los los requerimientos. porque se desea que compare caracter a caracter.

16/09/2017

Desarrollo de los algoritmos en mars

18/09/2017

Se continua con el desarrollo de los algoritmos, al parecer presenta un error.

21/09/2017

Se empieza a usar al FPGA por lo que hay que investigar cómo utilizarla, cuales son los pines de la misma, utilizar los 7 segmentos y los "switches".

Documentación del proyecto y continua con pruebas de cómo utilizar la FPA, mediante la creación de mux y sumadores

23/09/2017

Inicio de la modificación de la arquitectura para ajustarla a las necesidades de este proyecto

30/09/2017 al 10/10/2017

Continúa con el acople de la FPGA a la arquitectura del mismo

09/09/2017

Cambio en lo algoritmos porque se presentó un bug con los existentes.

Referencias:

1. Slideplayer.es. (n.d.). *Cadenas de Caracteres y Emparejamiento de Patrones - ppt descargar*. [online] Available at: <http://slideplayer.es/slide/3173781/> [Accessed 16 Oct. 2017].