Proyecto Final (Fase #2)

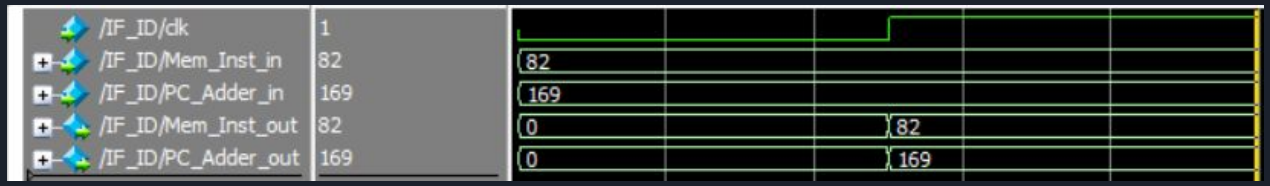
Equipo #1 :

Abraham Magaña Hernández - 220791217
Damián Guevara - 220791063
Samuel Espinoza Sucilla - 214017739

Seminario de Arquitectura en Computación.
López Arce Delgado Jorge Ernesto.
D13 Martes y Jueves 7:00 AM a 9:00 AM.

Código de Verilog

En las primeras reuniones se habló de cómo se estructuraban los buffers, empezamos el código de los mismos, no hubo un problema, los buffers funcionan por sí solos, no había fallas en sus conexiones y todo funcionaba también de acuerdo al clk.

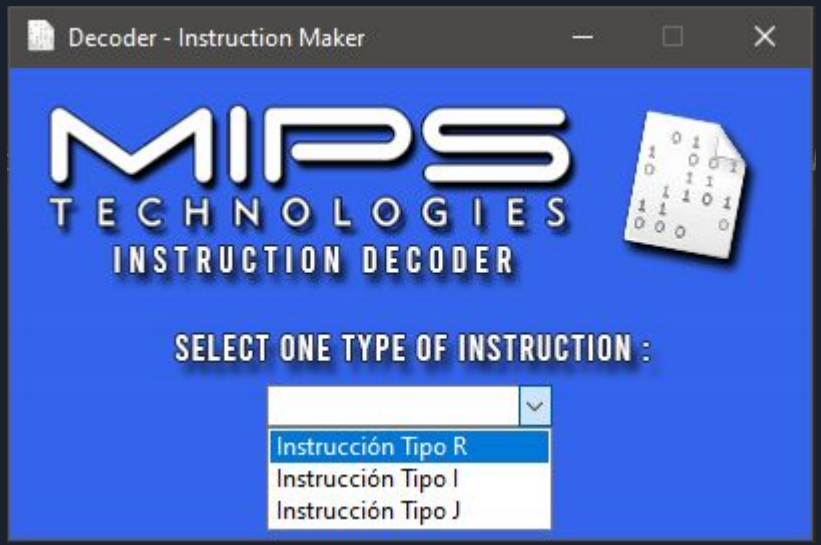



```
//Addi
6'b001000:
begin
  RegDst = 0;
  Branch = 0;
  MemRead = 0;
  MemToReg = 0;
  AluOp = 3'b000;
  MemToWrite = 0;
  AluSrc = 1;
  RegWrite = 1;
end
```

También se agregaron los case para la Unidad de Control y como esta debía tomar las instrucciones de Tipo I, esto se habló en varias reuniones y fue tema de varios errores al momento de hacer la simulación sin los buffers, pero al final se comprobó el output de cada instrucción, consiguiendo que todas funcionaran como deberían por separado.

Decodificador de Instrucciones - GUI

El decodificador está programado en Python 3.9.2 y cumple con la función de dar output al momento de Instrucciones de Tipo R y Tipo I.

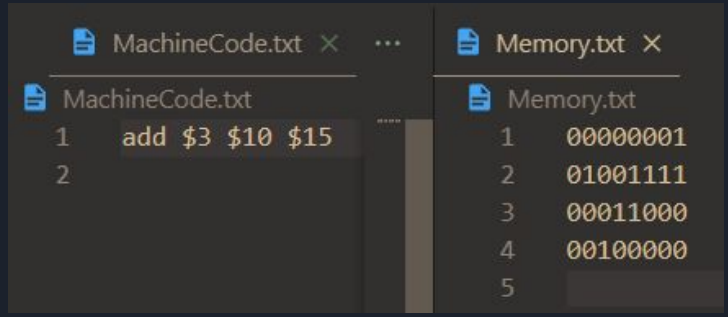


Es importante denotar que esta interfaz gráfica recibe los datos en decimal, nos da una previsualización de la instrucción en su formato de bits, bloquea las instrucciones erróneas, resetea archivos de memoria y también el archivo que contiene las instrucciones escritas en su formato de ensamblador.

Escritura de Archivo de Memoria y Ensamblador

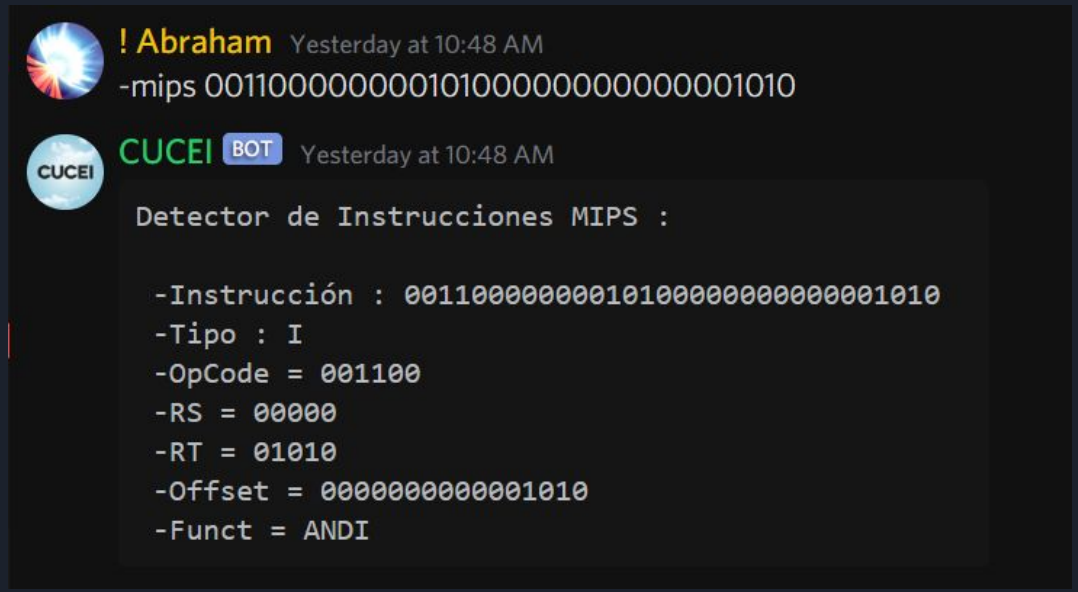
El archivo de la derecha es el output de la instrucción detectada como una válida escrita en el formato para que la segunda fase de este proyecto funcione.

Por otro lado, el archivo de la izquierda es la misma instrucción pero escrita en ensamblador, ambos archivos se pueden resetear simultáneamente.



Verificador de Instrucciones

Esta es más una herramienta usada también para mejorar el procedimiento para la creación de la propuesta en ensamblador, el decoder funciona para crear las instrucciones, pero lo que hace especial esta herramienta es que no hay que ejecutar algo como tal, solamente es un comando de discord, como tenemos un servidor de en esta app para llevar el trabajo de todo el proyecto tener un bot que con un comando haga un manejo de datos de todo lo que necesitamos, agiliza la forma de trabajar.



-mips -hexa
-bi -clear