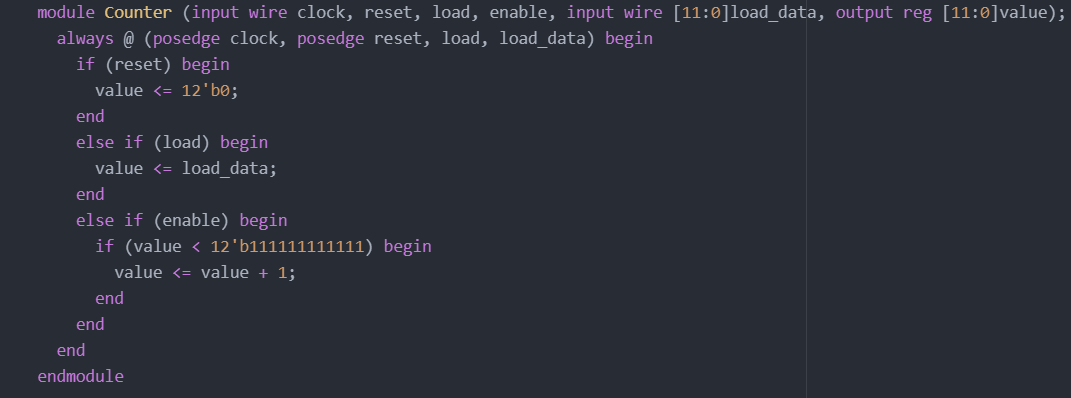
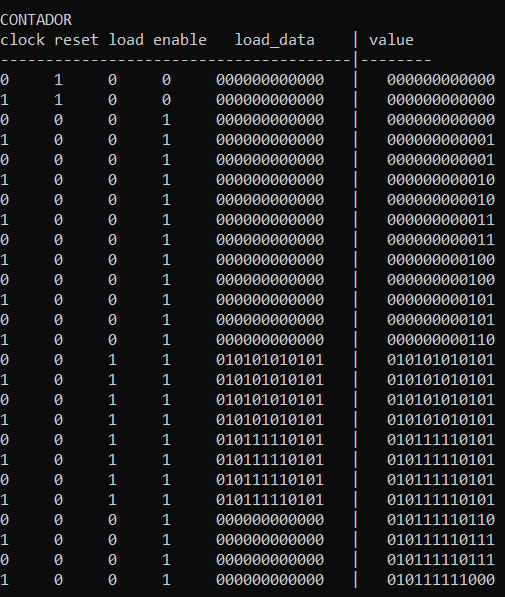
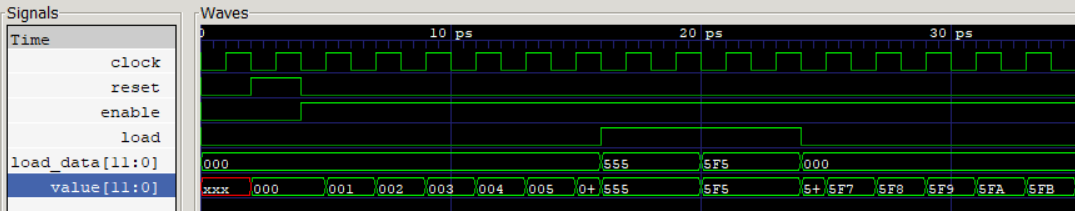
Laboratorio 8

Ejercicio 1:

Para la creación del contador se utilizaron bloques always para que pueda funcionar con el cambio del clock, el reset, el load o algún cambio en el dato que se está cargando. Por esto, como puede verse, se utilizaron output reg. Al mismo tiempo, como se observa, tanto el dato para cargar como el valor de salida son de 12 bits de ancho. Finalmente, se diseñó para que el valor se coloque en 0 si hay un reset, para que tome el valor del dato de carga cuando el load está activado y para que cuente 1 cada flanco de reloj si el enable está activado. Estas funciones pueden verse en el diagrama de timing de GTKWave y la tabla mostrada en la terminal.

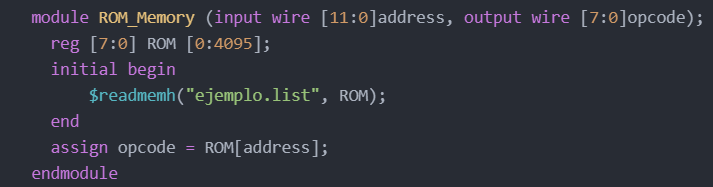


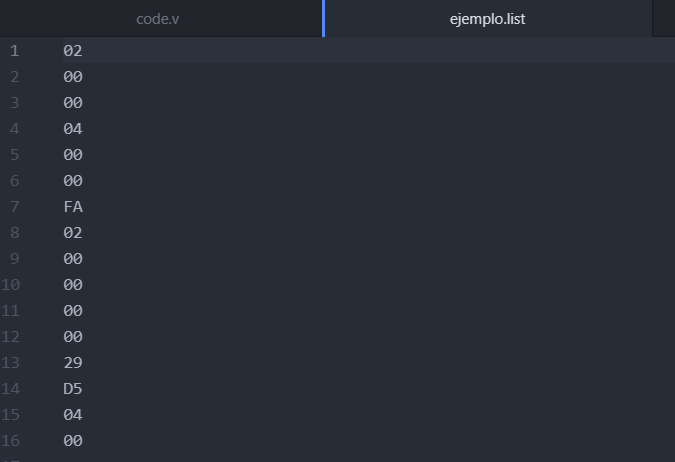


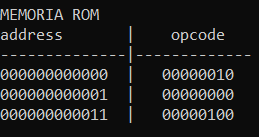


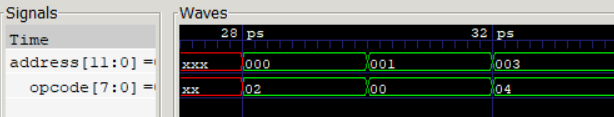
Ejercicio 2:

Para crear la memoria ROM se utilizó el arreglo de dos dimensiones de verilog y la función readmemh, la cual toma un documento con datos en hexadecimal y los coloca en la variable del arreglo 2D. Al módulo se le colocó un input que representa a la dirección que escoge qué byte se lee de la memoria y un output que representa la instrucción u opcode, el cual muestra el byte seleccionado. Como puede verse en la tabla de la terminal y en el diagrama de timing, al introducir las direcciones 0, 1 y 3, se muestran los datos que corresponden a estas direcciones en el documento ejemplo.list.









Ejercicio 3:

Para la creación de la ALU se utilizó un bloque condicional case, el cual muestra todos los escenarios posibles que puede tomar una variable. Este bloque se coloca en un bloque always para que pueda ejecutarse a cada momento. En las condiciones se colocaron todas las opciones que muestra la ALU presentada en el libro.

