**MEMORIA ROM**

ROM significa Memoria de sólo lectura en español, es un tipo de memoria de computadora que solo puede ser leída, pero no escrita ni cambiada por el usuario. Se utiliza para almacenar datos permanentes o semipermanentes.

Los datos en ROM se programan en la memoria durante el proceso de fabricación, y no se pueden modificar ni borrar por el usuario. La ROM es no volátil, lo que significa que los datos almacenados en ella no se pierden cuando se apaga la energía. Esto la hace útil para almacenar información crítica del sistema que necesita ser accesible de forma constante.

En nuestro caso utilizaremos la memoria ROM para almacenar las instrucciones que realizara nuestro procesador, para esto crearemos un nuevo modulo en ISE-DESIGN que llamaremos MROM y necesitáremos 2 entradas y 1 salida:

* ENTRADAS
  + Clk: esta entrada será el reloj de nuestro componente.
  + Addres: con esta entrada de 4 bits podremos decidir que instrucción mandaremos en su salida.
* SALIDAS
  + Data\_out: salida de 32 bits donde madaremos nuestras instrucciones.

Captura de pantalla de computadora

Descripción generada automáticamente

**CODIGO**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.numeric\_std.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity MROM is

Port (clk: in STD\_LOGIC;

address: in STD\_LOGIC\_VECTOR (3 downto 0);

data\_out: out STD\_LOGIC\_VECTOR (31 downto 0));

end MROM;

architecture Behavioral of MROM is

constant mem\_size: natural :=2\*\*10;

--0: Suma

--1: resta

--2: and

--3: or

--4: NegB

--5: comp2B

--6: beq

--7: bne

--8: j

--9: Si es otro valor regresa a 0.

type mem\_type is array (9 downto 0) of STD\_LOGIC\_VECTOR (31 downto 0);

constant mem : mem\_type := (0 =>x"00430801", 1=>x"00A62002", 2=>x"01093803", 3=>x"016C5004", 4=>x"039ED005",

5=>x"01B79006", 6=>x"04240006", 7=>x"08EA0006", 8=>x"0C000000", others=>x"00000000");

begin

rom: process (clk)

begin

if rising\_edge (clk) then

data\_out<= mem(to\_integer(unsigned(address)));

end if;

end process rom;

end Behavioral;

PC

El contador del programa, también conocido como Program Counter (PC), es un registro en la CPU (Unidad Central de Procesamiento) de una computadora que contiene la dirección de memoria de la próxima instrucción que se ejecutará en la secuencia de instrucciones del programa.

El funcionamiento de nuestro PC es simplemente de pasar el contenido de su entrada a su salida cada vez que detecta un flanco de subida en el reloj.

Para crear este componente en ISE-DESING necesitaremos 2 entradas:

* Clk: esta entrada será el reloj de nuestro componente.
* Ent: una entrada de 4 bits que será proporcionada por el sumador para posteriormente enviarlo a la MROM.

También necesitamos una salida:

* Rout: salida de 4 bits que esta conectada a la MROMy enviara la señal en “Ent” cada flanco de subida del reloj.

Una captura de pantalla de una computadora

Descripción generada automáticamente con confianza media

CODIGO

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity PC is

Port ( clk : in STD\_LOGIC;

Ent : in STD\_LOGIC\_VECTOR (3 downto 0);

Rout : out STD\_LOGIC\_VECTOR (3 downto 0));

end PC;

architecture Behavioral of PC is

begin

process (clk, Ent)

begin

if (clk'event and clk ='1') then

Rout <= Ent;

end if;

end process;

end Behavioral;