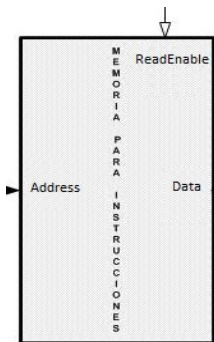


Identificación	Nombre	Símbolo	Descripción	Tabla de verdad	Funciones Lógicas	Código	Nombre del archivo						
2	Memoria para instrucciones		Unidad de memoria que nos permite almacenar datos que se ingresan con la entrada address que es de 32 bits y provoca una salida de 32 bits	<p>Tabla Utilizada:</p> <table><tr><th>Read_Enable</th><th>Data</th></tr><tr><td>0</td><td>32'b0</td></tr><tr><td>1</td><td>mem[address]</td></tr></table> <p>En la tabla se presenta la situación si el read enable está activado entonces la salida será el dato almacenado en la dirección ingresada.</p> <p>Si no está activado entonces la salida serán ceros.</p>	Read_Enable	Data	0	32'b0	1	mem[address]	Las funciones lógicas utilizadas en este módulo son únicamente Cuando read_enable esté activado entonces la salida se activa, Si no entonces la salida serán ceros.	<pre>module MemoriaInstrucciones(address , // Direccion de entrada data , // Datos de salida read_en , // Read Enable); input [31:0] address; output [31:0] data; input read_en; reg [31:0] data ; // Bloque combinacional para la ROM o memoria de instrucciones always @(ce or read_en or address) begin //Si se da el chip enable y el read enable entonces da los datos de salida if(read_en) begin case (address) //Datos predeterminados de la Rom 0 : data = 10; 1 : data = 55; 2 : data = 244; 3 : data = 0; 4 : data = 1; 5 : data = 8'hff; 6 : data = 8'h11; 7 : data = 8'h1; 8 : data = 8'h10; 9 : data = 8'h0; 10 : data = 8'h10; 11 : data = 8'h15; 12 : data = 8'h60; 13 : data = 8'h90; 14 : data = 8'h70; 2183648 : data = 8'h90; default: data = 0; endcase end else data = 0; end endmodule</pre>	Memoria_Instrucciones_V1_ChavarriaOrtegaOrtiz Villalobos.v
Read_Enable	Data												
0	32'b0												
1	mem[address]												

