**Laboratuvar Çalışması 0x5**

Hafıza

Bu labın amacı

* Hafıza oluşturup, veri okumak,
* Tasarlanan devreleri gerçekleyip test edebilmek.

### Problem 1 - Hafıza oluşturma ve okuma

Bu problemde sizden istenen oluşturduğunuz küçük bir hafızadan verilen adres değerine göre, o adreste bulunan verileri çekmektir. Oluşturacağınız hafıza registerları tutacaktır. Toplam register sayısını 32 olarak alınız. Devrenin temel çalışma prensibi, rs1 ve rs2 değerlerini adres gibi alıp, hafızada içinde barındırdıkları 32-bit dataları **rs1\_data** ve **rs2\_data** portlarına aktarması istenmektedir. Okuma işlemini kombinasyonel tasarlayınız. Basit bir hafıza kodu örneği ekte verilmiştir.

Bu problem için sadece okumayı test eden bir testbench örneğini githubdaki repoda bulabilirsiniz.

module p1 (

input logic clk, reset,

// yazma portları

input logic we,

input logic [4:0] waddr,

input logic [31:0] wbdata,

// okuma portları

input logic [4:0] rs1,

input logic [4:0] rs2,

output logic [31:0] rs1\_data,

output logic [31:0] rs1\_data

);

## Ekler

### EK A - Hafıza oluşturma

|  |
| --- |
| /\* memory.sv  \*  \* iki portlu hafiza devresi ornegi  \* birinci port sadece yazmak icin,  \* ikinci port sadece okumak icin kullaniliyor.  \*/  module memory (  input logic clk, reset,  input logic we,  input logic [2:0] waddr, addr,  input logic [15:0] din,  output logic [15:0] dout  );  // mem adında 16-bit uzunluğunda 8 elemanlı bir hafiza olustur.  // mem[0], mem[1], ... mem[7] olacak  // ve her biri 16 bit uzunlugunda olacak  logic [15:0] mem [0:7];  // baslangicta hafizayi varsa reg\_image dosyasindaki degerlerle doldur  // Ornek reg\_image dosyasini misc/ klasoru altindan indirip kullanabilirsiniz.  // ModelSim proje klasorunun altina kopyalamaniz gerekir.  // Calismazsa, butun memory okumalarinda X gorebilirsiniz.  initial begin  $readmemh("reg\_image.mem", mem);  end  // write portu  // rising edge clk geldiğinde eger we biti aktif ise  // din i mem dizisinin waddr inci elemanına aktar.  always\_ff @(posedge clk)  if (we) mem[waddr] <= din;  // read portu  // combinational olarak kullan  // extra read portu buraya eklenebilir  assign dout = mem[addr];  endmodule |