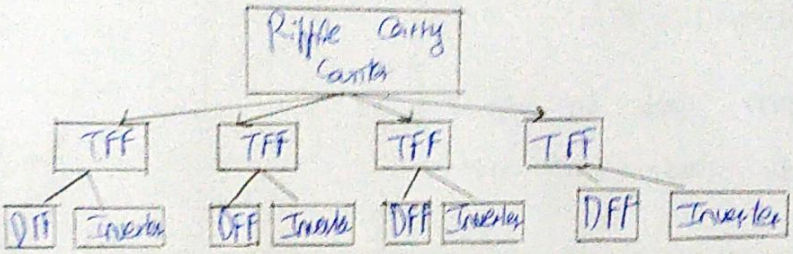
VLSI Lab

Experiment - I

I Pre-lab Questions

1 List the types of design methodologies for digital design with an example?

Sol: There are two basic types of digital design methodologies
i) Top-down design methodology
ii) Bottom-up design methodology



2 Give the difference between module and module instance?

Sol: When a module is invoked, Verilog creates a unique object from the template. Each object has its own name, variable parameters, and I/O interface. The process of creating objects from a module template is called instantiation and the objects are called instances.

3 What are built in gate primitives?

Sol: Verilog provides a standard set of primitives, such as and, nand, or, nor etc as a part of the language. These are also commonly known as built in primitives.

4 Give the use of net, reg and wire data types

Sol: NETS - the nets variable represents the physical connection between structural entities. These variables do not store values (except trireg). They have the value of their driver. Changes consistently by the driving circuit.

reg - Can be used to create registers in procedural blocks. They can store values.

A wire represents a network of connection with each connection either driving a value (or) responding to the resolved value. Easy when on the list

5. Declare the following variables in Verilog

a) An 8 bit reg not called a_in

Soln wire [7:0] a_in;

b) An integer called Cout

Soln integer Cout;

c) An array called delays, Contains integer type values.

Soln integer delays [0:19];

d) A parameter Cache_size equal to 512

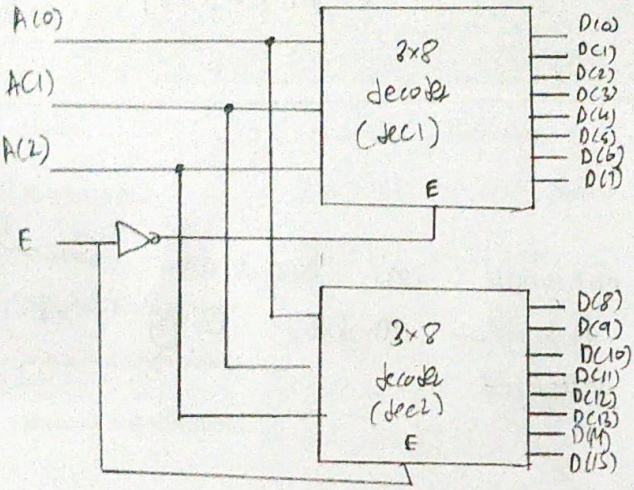
Soln parameter Cache_size = 512;

II Post-Lab Questions

1. Draw the block diagram of 4×16 decoder. Use 3×8 decoder to design 4×16 decoder using gate level modelling.

Soln.

Diagram



Gate level Modelling program

```
Module Dec 3to8 ( A, E, D);
    input [2:0] A;
    input E;
    output [7:0] D;
    assign D[0] = E & ~A[2] & ~A[1] & ~A[0];
    assign D[1] = E & ~A[2] & ~A[1] & A[0];
    assign D[2] = E & ~A[2] & A[1] & ~A[0];
    assign D[3] = E & ~A[2] & A[1] & A[0];
    assign D[4] = E & A[2] & ~A[1] & ~A[0];
    assign D[5] = E & A[2] & ~A[1] & A[0];
    assign D[6] = E & A[2] & A[1] & ~A[0];
    assign D[7] = E & A[2] & A[1] & A[0];
endmodule;
```

```
module Dec4to16 (A, D);
    input [3:0] A
    output [15:0] D;
    Dec3to8 dec1 (A(A[2:0], E(AC3)), D(D[15:8]));
    Dec3to8 dec2 (A(A[2:0], E(A[3]), D(D[7:0]));
end module;
```

RESULT:

Realization of Combinational and Sequential Circuits using Gate-Level and Dataflow modelling using Modelsim is done and output is verified.

LATE

14/2/25