VIS : Technology Transfer Course

University of California, Berkeley

# Lab # 4 : Verification by Simulation

Yuji Kukimoto

05/15/96

## 1 Introduction

The main goal of this lab is to familiarize yourself with the notion of nondeterminism. After modeling the client module using non-determinism, we will verify the entire system by simulation.

## 2 Hands on Experience

**Problem #1** Draw a state transition diagram of the client module using nondeterminism. The functionality of the client module should be as follows:

> The client module takes a global clock `clk` and an acknowledgment signal `ack` as inputs and generates a request signal `req` as an output. In the initial state, the request line is set to zero, i.e. the client has no request. The client module then sets the request line high nondeterministically and, once it is set, keeps it high until an acknowledgment is given. The client module sets the request line low at the next clock tick after an acknowledgment is given and goes back to the initial state.

Repeat the same question for the case where the client module sets the request line low *nondeterministically* after an acknowledgment is given.

**Problem #2** Write the two types of the client module in Verilog. You can use the files "ex1a.v" and "ex1b.v" respectively, which contain the system description from "ex0.v" and a template for writing the client.

**Problem #3** What properties should the entire system satisfy? Write them down in English.

**Problem #4** Verify by random simulation whether the entire system satisfies the properties you came up with for each type of the client modules. The command sequence of VIS would be

```
read_blif_mv foo.mv
init_verify
simulate -n 20
```

The last argument of the simulate command is the number of cycles to be simulated. If the system does not satisfy the properties, what is wrong with the design?

| | Property: English description: | Did it simulate right? | If no, what is wrong in design? |
|---|---|---|---|
| ex1a | | | |
| ex1b | | | |