國 立 中 興 大 學

資 訊 工 程 學 系

Computer-Aided Design – Fall, 2023

Assignment #3

Due Dec. 13

The focus of the second assignment is MIS: A multi-level combinational logic optimizer. MisII is a powerful logic synthesis tool, and is widely used in industry and academia. It is included in SIS later.

1. Using sis

1.1. Input Specification

There are several ways to input logic descriptions or circuits to misII (and sis). Input description for espresso is also applicable to misII.

- Boolean networks: These are specified using the blif format. This is nothing but a netlist description of the Boolean network. The documentation for *blif* may be found along with the source code.:
- Mapped Networks: These are netlists of library gates. These are specified in *blif* format.
- Logic Equations: misII can also accept a set of logic equations describing the network. There is no documentation for this; the following example should be adequate to describe this:

$$f = x1' + x2;$$
$$g = f + x2 * x3;$$

- Espresso format: Essentially each row in an espresso expression is a product term of a PLA; or equivalently, each row specifies a cube. The document can be viewed with the instruction “octman -5 espresso”

1.2. Running sis

Sis can be sun in either interactive or batch mode. In the interactive mode commands are provided for input-output, network manipulation, and other miscellaneous functions. Sequences of commands that are known to be effective with specific cost functions (area and delay) are stored in script files. These are available in ~cad/octtools/sun4/lib/misII/lib. The script files may be executed by using the *source* command. In the interactive mode there is an on-line help command.

Cell libraries are specified using a *genlib* format. Some of the available libraries are in the library (lib) directory. Read in a library and then map the circuit for some optimization metric.

2. Things to Do

Try to play with sis. Please create a four-bit ripper adder in any readable format; for example: “adder.blif”.

Use the script whose name is ‘script’ to do the optimization. Compare the results with the unoptimized circuit. A simple discussion on the usage of misII can be found in pp. 129-132 of the book: *Contemporary Logic Design*, by R. H. Katz.

(1) Read the original circuit; this can be done by issuing “rl adder.blif” (this is an alias, the entire command should be “read_blif adder.blif”). Executing the following commands:

```
misII> rl adder.blif
misII> pf
misII> ps
misII> rlib msu.genlib
misII> map
misII> pg
misII> pat
```

The pf (“print_factor”) command in the second line prints the circuit in factored form. The ps (“print_stats”) in the third line prints the status of the current network, including number of nodes, number of literals in both factored-form and sum-of-product form. The fourth line reads in a library of basic cells, in this case the MSU general library (see pp. 132). The fifth line maps the Boolean expression into actual gates (in the cell library). The sixth line prints of the area occupied by the gates, while the seventh line prints the signal arrival time at each node.

(2) Implement the optimized circuit by performing some logic operations. Executing the following commands:

```
sis> rl adder.blif
sis> source script
sis> pf
sis> ps
sis> rlib msu.genlib
sis> map
sis> pg
sis> pat
```

The fourth line optimizes the Boolean expression with a standard script called “script”. The following commands are essentially the same as those in (1); they will show you whether your result is good in this case.

Record the results in both runs, compare the results. Also draw the final circuits correspond to the result in (1) and (2).