

## Problem 2: State Minimization

Source: SpringSoft Inc.

Dec. 20, 2000

### 1. Introduction

Sequential circuit optimization has been the subject of research for several decades. One problem is the state minimization problem. It can be defined informally as deriving a finite-state machine with the same behavior but with a minimum number of states. The reduction in states correlates to a reduction in transitions, and hence usually leading to a reduction of logic gates.

### 2. Input

A state transition file in BLIF [1] will be used as the input. You will need to implement a simple BLIF reader for this problem.

The format of an FSM description in BLIF is:

```
.module <module-name>
.input <names of inputs>
.output <names of outputs>
.start_kiss
.i <num-inputs>
.o <num-outputs>
[.p <num-terms>]
[.s <num-states>]
[.r <reset-state>]
<input> <current-state> <next-state> <output>
...
<input> <current-state> <next-state> <output>
.end_kiss
```

where

<num-inputs> is the number of inputs to the FSM, which should agree with the number of inputs in the .inputs construct for the current model.

<num-outputs> is the number of outputs of the FSM, which should agree with the number of outputs in the .outputs construct for the current model.

<num-terms> is the number of state transitions in the format of 4-tuples "<input> <current-state> <next-state> <output>" that follow in the FSM description.

<num-states> is the number of distinct states that appear in "<current-state>" and "<next-state>" columns.

<reset-state> is the symbolic name for the reset state for the FSM; it should appear somewhere in the "<current-state>" column.

<input> is a sequence of num-inputs members of {0,1,-}.

<output> is a sequence of num-inputs members of {0,1,-}.

<current-state> and <next-state> are symbolic names for the current state and next state transitions of the FSM.

### **3. Output**

The reduced finite-state machine in BLIF.

### **4. Examples**

Please see Part B of the problem description. The example is a 5-states FSM. After state minimization, it becomes a 4-state FSM.

### **5. Language/Platform**

1. Language: C or C++.
2. Platform: SUN OS/Solaris or PC Windows.

### **6. Evaluation**

The evaluation will be based on correctness, CPU time, and memory usages.

### **7. Questions**

Please report any question regarding this problem to [cad@cis.nctu.edu.tw](mailto:cad@cis.nctu.edu.tw) with the email subject "CAD Contest: Problem 2." Your question(s) will be answered in two weeks, and the Q&A's will be posted at the contest web site

### **Reference**

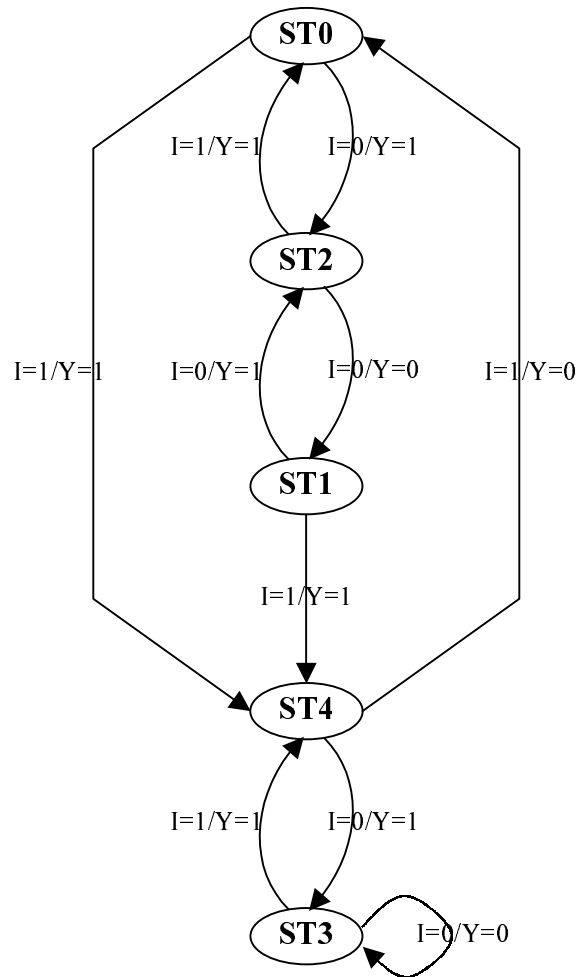
[1] Berkeley Logic Interchange Format (BLIF),  
<http://www-cad.eecs.berkeley.edu/Respep/Research/vis/blif.ps>

## B · Original State Table & State Diagram

```

.model FSM
.inputs I
.outputs Y
.start_kiss
.i 1
.o 1
.p 10
.s 5
0 ST0 ST2 1
1 ST0 ST4 1
0 ST1 ST2 1
1 ST1 ST4 1
0 ST2 ST1 0
1 ST2 ST0 1
0 ST3 ST3 0
1 ST3 ST4 1
0 ST4 ST3 1
1 ST4 ST0 0
.end_kiss
.end

```



## C · Reduced State Table & State Diagram

```

.model FSM
.inputs I
.outputs Y
.start_kiss
.i 1
.o 1
.p 8
.s 4
0 ST0_ST1 ST2      1
1 ST0_ST1 ST4      1
0 ST2      ST0_ST1  0
1 ST2      ST0_ST1  1
0 ST3      ST3      0
1 ST3      ST4      1
0 ST4      ST3      1
1 ST4      ST0_ST1  0
.end_kiss
.end

```

