CS 6132

Advanced Switching Theory

Final Project

Objective:

Timing-Driven Technology Mapping considering Area Overhead

In this project, you are going to implement a timing-driven technology mapping algorithm based on the dynamic programming method taught in class. Since the method taught in class always maps each node with optimal mapping scheme in terms of timing before critical paths are determined, the slacks of the nodes on non-critical paths may be large. This means the area overhead of the nodes on non-critical paths can be reduced by utilizing these slacks without affecting the timing of critical paths. The flow or algorithm of your program is not restricted, but the mapping algorithm must be thoroughly implemented in your program. That is, you cannot use tools like SIS or Design Compiler to do technology mapping for you. There are many research papers discussing the problem of this project. You can reference these papers to improve your results.

Input Files:

Cell Library Format (cell.lib)

The cells used for technology mapping are defined in “cell.lib”. Each cell is specified in the following format.

GATE <cell-name>

<cell-logic-function>

<cell-area>

<cell-delay>

<*cell-name*> is the name of the cell in the cell library.

<*cell-logic-function*>is an equation written in conventional algebraic notation using the operators ’+’ for OR, ’*’ or nothing (space) for AND, ’!’ for NOT, and parentheses for grouping. The names of the literals in the equation define the input pin

names for the cell; the name on the left hand side of the equation defines the output of the cell. The equation terminates with a semicolon. Only single-output cells may be specified.

<*cell-area*> defines the area of the cell.

<*cell-delay*> defines the cell delay of the cell. In this project, only cell delay is concerned. Net delay is neglected.

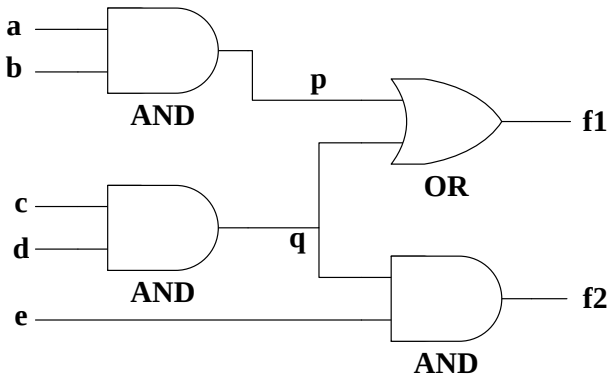
For example, an aoi22 gate is specified as follows.

```
GATE aoi22_x1
O =    ! ((a1 * a2) + (b1 * b2));
2320.00
3.9638
```

"aoi22_x1" is the cell-name. "O = ! ((a1 * a2) + (b1 * b2));" represents the cell's logic function where "O", "a1", "a2", "b1" and "b2" are its input/output pins. "2320.00" is the cell's area and "3.9638" is the cell's delay.

EQN Circuit Format (*.eqn)

The circuits to be mapped are specified in EQN format. This is a standard format of SIS and is technology independent. Consider the following example and its EQN circuit.




```
INORDER = a   b   c   d   e;
OUTORDER = f1   f2;
p = a * b;
q = c * d;
f1 = p + q;
f2 = q * e;
```

The line "INORDER = a b c d e;" specifies that "a", "b", "c", "d", "e" are connected to the primary inputs of circuit. Similarly, the line "OUTORDER = f1 f2;" specifies that "f1" and "f2" are connected to the primary outputs of the circuit. The next four lines describe the logic relations among the nets. For example, "q = c * d;" describes that "q" should have the value equals to "c" AND "D".

More information about this format can be found in internet if the above information

is insufficient for you.

Output Files & Information:

BDNET Format (*.bd)

The output file of your program is the mapped circuit in BDNET format. This is a technology dependent format used in SIS. An example with comment is given below.

```
MODEL “TEST.AST”;  // circuit name

INPUT          // primary input
   “ID0(0)” : “ID0(0)”          // primary input name and its corresponding net
   “ID1(1)” : “ID1(1)”;         // primary input name and its corresponding net

OUTPUT       // primary output
   “OD0(242)” : “[2330]”;     // primary output name and its corresponding net

INSTANCE    “nand2_lvt”:”pysical”    // the mapping cell from cell lib
   “a”      :      “[2998]”;                  // input a and its corresponding net
   “b”      :      “[924]”;                   // input b and its corresponding net
   “O”      :      “[1670]”;                  // output name and its corresponding net
…
…
…
ENDMODEL;
```

This format is also a standard format. Therefore, you can find more formal definition of this format from internet if the above example is insufficient.

Your program should also report the total cell area and the worst delay (the latest arrival time of all primary outputs). It should be stored in a file with format shown below.

area <total-area>

delay <worst-delay>

An Example:

C17.eqn + cell.lib => C17.bd + C17.report

An example named C17 is given in the file "2008final.rar" which can be downloaded from the course website. Note that the file "C17.bd" is only a mapped circuit without any optimization in terms of area or timing. It simply presented to help you to understand the file formats used in this project. In this mapped circuit, the worst delay of this circuit is 7.98 and the total area is 11136.

Verification:

In this project, you also need to provide a verification flow to make sure that your technology mapping program does not alter the functionality of the circuits. That is, you have to show that your program is correct. For this part, you are allowed to SIS. (*Hint: There are some commands in SIS can do Boolean equivalence checking… However, you may need to transform your circuit to another format…*)

Grading:

Timing Behavior: 50%
Area Cost: 30%
Verification Flow: 10%
Demonstration: 10%

The score of "Timing Behavior" and "Area Cost" will be determined by comparing your result with the average result.

Due Date & Demonstration:

Due Date: 2009/1/13 6:00AM
Demonstration: 2009/1/13~2009/1/14

If you have any question, please contact to TA: 謝昂志
E-mail: achsieh@cs.nthu.edu.tw Ext. 33573