University of Toronto, Faculty of Applied Science and Engineering
Department of Electrical and Computer Engineering

ELE 1387F -CAD for Digital Circuit Synthesis and Layout Handout #15

November 1999 J. Rose

Assignment #3: Dynamic Programming Solution to the Technology Mapping Problem

Assignment Date: November 17

Due Date: December 1

Late Penalty: **-1 per day late, with total marks = 10**

You are to write an implementation of the dynamic programming approach to technology mapping of a boolean network as described in class, and in handouts #11 and #14. The problem is a simplified version of the general case.

Assume that the input subject network is a tree that has already been decomposed into two-input NAND gates and inverters. The input library of gates will also be given as an already decomposed network of 2-input NANDs and inverters. You **do not** need to consider the different permutations of the library gates, as discussed in class, but just match them with the topology as it is given in the input.

The output of your program should be a network of gates with minimum total cost.

Your program should display the subject network and the progress of the algorithm as various matches are achieved. Use the X11-based graphics package, as in Assignments 1 and 2. **Your graphics display should illustrate the progress of the algorithm, not simply the final answer.**

Your program should use the following input format, which describes subject network and library elements (i.e. gates): The first line of the file describes the **record** type. There are two kinds of records - one which describes the subject network, and a library record which describes the gates. There is only one subject network record. The record description line indicates the type of record, and the inputs and output. All fields are integers.

RecordType [GateType] [Cost] In1 In2 ... InN -1 Out

RecordType is 1 for the subject network, and 2 for a library (gate) element. GateType is a positive number that is the number of the gate in the case of a library record. Each gate has a unique number, and is used in the output netlist described below.

You should assume that the subject network is a tree (i.e.it has no fanout).

Cost is area cost of the gate, in the case of library record. Neither GateType nor Cost appear in a subject network record descriptor. The In# give the number of the nets that are the inputs to either the network or the gate, and Out is the output number of the network or gate.

The list of record descriptors is terminated by a RecordType of -1.

Following each record descriptor is a set of lines that give the tree describing either the subject network or

the gate network. Each line has the following form:

NodeNum NodeType NetNum1 NetNum2 [NetNum3]

NodeNum is a unique number of a node. Note that this number is unique only within the network, and is not unique between the subject network and gate networks (I re-use node numbers and net numbers alot, so don't rely on uniqueness). NodeType is one of 1 (for an inverter) or 2 (for a 2-input NAND gate). For the inverter NetNum1 is the number of the net attached to the input NetNum2 is the number of the net attached to the output and NetNum3 is not present. For the 2-input NAND Gate NetNum1 and NetNum2 are the numbers of the nets attached to the inputs and NetNum3 is the number of the net attached to the output.

The list of Nodes is terminated with a NodeNum of -1.

Here is an example input file taken from the example used in class:

File Contents	*Meaning*
1 1 2 3 4 -1 9	**Subject Network (SN), inputs 1,2,3,4 output 9**
1 2 1 2 5	**SN - nand2 - inputs 1,2 output 5**
2 2 3 4 6	**SN - nand2 - inputs 3,4 output 6**
3 1 5 7	**SN - inverter - input 5, output 7**
4 1 6 8	**SN - inverter - input 6, output 8**
5 2 7 8 9	**SN - nand2 - inputs 7,8 output 9**
-1	**End of subject network**
2 1 5 1 -1 2	**Gate Network (GN), (inverter) Gate #1, cost=5, input 1, output 2;**
1 1 1 2	**GN - inverter input 1 output 2**
-1	**End of gate network**
2 2 10 1 2 -1 3	**New GN, (nand2) Gate #2, cost=10, inputs 1,2 output 3;**
1 2 1 2 3	**GN - nand2 inputs 1, 2 output 3**
-1	**End of gate network**
2 3 12 1 2 -1 4	**New GN, (and2) Gate #3, cost=12, inputs 1,2 output 4;**
1 2 1 2 3	**GN - nand2 inputs 1, 2 output 3**
2 1 3 4	**GN - inverter input 3 output 4**
-1	**End of gate network**
2 4 12 1 2 -1 5	**New GN, (or2) Gate #4, cost=12, inputs 1,2 output 5;**
1 1 1 3	**GN - inverter input 1 output 3**
2 1 2 4	**GN - inverter input 2 output 4**

3 2 3 4 5	**GN - nand2 inputs 3, 4 output 5**
-1	**End of gate network**
-1	**End of record descriptors (hence file)**

The output format should be given in a similar manner to the network descriptor files:

NodeNum GateType NetNum1 Netnum2 .... NetNumN

Where **NodeNum** is a unique Gate identifier, and **GateType** is one of the gate numbers given in the gate record descriptor above (e.g. or2 is GateType 4) and NetNum1 ... NetNum(N-1)are the inputs to the gate and NetNumN is the output. For example, one possible mapping of the network above is:

1 3 1 2 5	**Node 1 of output: #3 (and2), inputs 1, 2 output 5**
2 3 3 4 6	**Node 2 of output: #3 (and2), inputs 3, 4 output 6**
3 2 5 6 7	**Node 3 of output: #2 (nand2), inputs 5, 6 output 7**

You should test your program on the following five test files in the directory ~jayar/1387/a3 on the eecg and ECF systems, and at www.eecg.toronto.edu/~jayar/courses/ece1387/a3/testfiles/:

net1, net2, net3, net4, and net5.

What to hand in:

1. Your program, and give the location of the executable, which should be on an eecg machine. Indicate how to run the program.

2. Include a short description of the flow of your program, assuming that I already have a basic knowledge of the dynamic programming method. Indicate any enhancements to the basic algorithm that you have made.

3. You should also hand in the solution to the five problems, in the format described above, along with the cost of the minimum solution. Please include the print outs of the graphics display of the solutions.