Technology-Dependent Logic Optimization
Gate Sizing

PA3

Introduction

- Technology mapping is a step of
  - taking a logic netlist as the input and
  - expressing the netlist using a set of gates from a technology library

- In the technology library, a gate may have different configurations, resulting in different area, delay, and power

- In this assignment, you are asked to write a program to determine the gate configurations to minimize the area subject to a delay constraint

Problem Formulation

- Inputs
  - A netlist in the AIG format
  - A technology library
- Output
  - A functionally equivalent netlist implemented with gates in the technology library
- Objective
  - Optimize the total area while meeting the timing constraint (initial delay)

Input 1 – Netlist (.blif)

- Netlist are given in the blif format
- Use ABC to parse the netlist and transfer it into the AIG format
- Don’t optimize the given netlist
  - Io_ReadBlifAsAig() # in main.c
  - Simply use ”strash” after reading the blif file

Input 2 – Technology Library (.lib)

- **INV**

Pattern Type	Timing (n:#FOs)	Area
INV1	3.38+1.03*n	0.044
INV2	3.28+0.47*n	0.058
INV3	2.82+0.42*n	0.073
INV4	3.30+0.24*n	0.087
INV5	3.13+0.22*n	0.102

- **NAND**

Pattern Type	Timing (n:#FOs)	Area
NAND1	4.60+0.74*n	0.058
NAND2	4.34+0.30*n	0.087
NAND3	4.32+0.15*n	0.146

Output – Netlist (.mbench)

- Output the mapped netlist in the modified bench format
- You cannot modify PI and PO names

```
# Initial delay: xxx
# Original area: xxx
# Optimized area: xxx
INPUT(x1)
INPUT(x2)
..
OUTPUT(y1)
OUTPUT(y2)
…
G1 = INV1(x1)            # slack
G2 = NAND2(x2, G1)       # slack
…
```

Technology Mapping

- An AND node is mapped to an NAND + an INV
- A complemented fanin (phase==1) is mapped as an INV
- Two INVs on a wire are redundant and need to be removed

- Initial delay
  - Circuit delay under which all the nodes are mapped to the fastest gates
    - Circuit delay is defined as the arrival time of the PO with the maximum arrival time
  - An AND node with *n* fanouts is mapped as an NAND + *n* INVs
    - Each fanout has one INV
  - Topological timing analysis
  - Note: the arrival time of each PO after optimization should not exceed the initial delay

Delivery & Due Date

- A zip file including
  - Your source code and a ReadMe describing how to compile and run your program
  - Your mapping results (.mbench) of ISCAS’85 benchmarks

    c432 ~ c7552

- Due on 2024/6/9 (Sun)