Principles Of
Digital Design

Technology Mapping

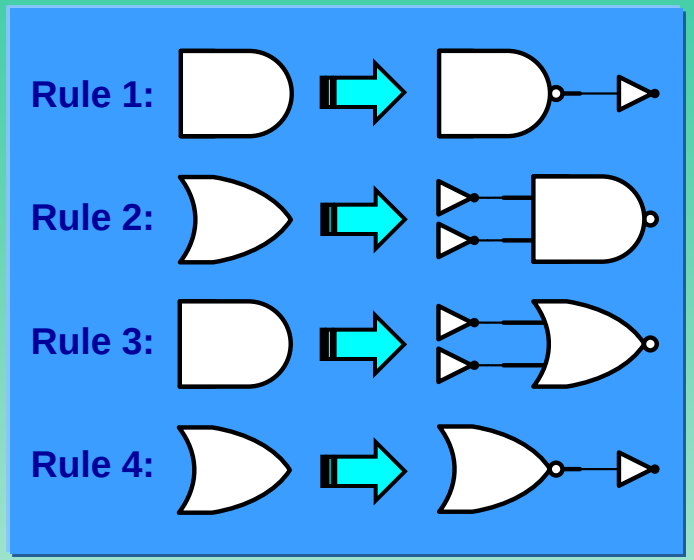
Mapping Boolean Expressions to Gates

- *Mapping to NAND and NOR Gates*
- *Delay Minimization*

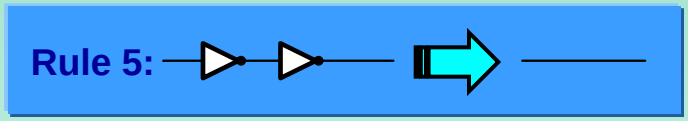
Technology Mapping for Gates

- **Some implementation technology usually have a library with only one type of gate (such as 3-input NOR, or 3-input NAND)**
- **Technology mapping is a transformation of Boolean expressions into a logic schematic containing only given type(s) of gate(s)**
- **Technology mapping consist of three tasks**
  - **Conversion replaces each operator with an operator representing the gate function given in the gate array**
  - **Optimization eliminates unnecessary inverters**
  - **Decomposition replaces a *n*-input gate with an *m*-input gate available in the library**

Conversion and Optimization



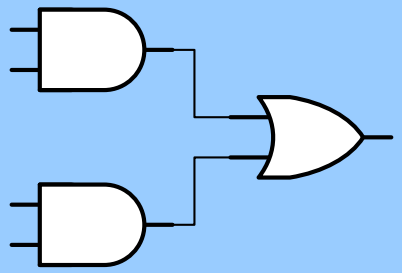
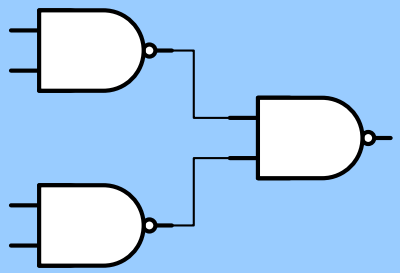
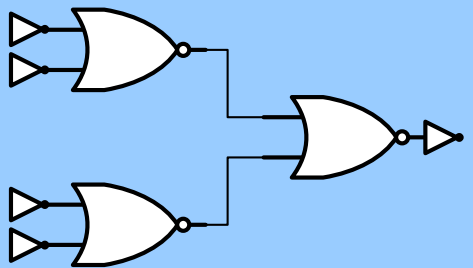
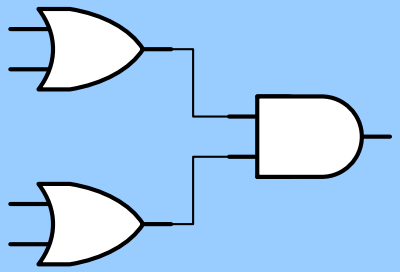
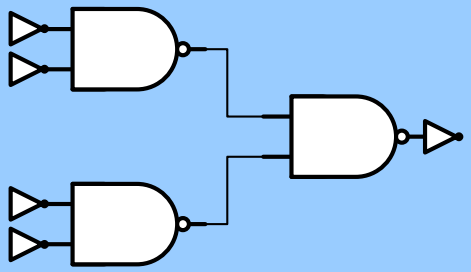
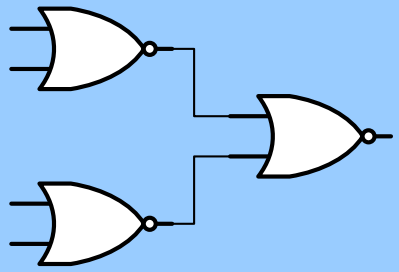
Conversion Rules



Optimization Rules

- **Conversion Procedure:**
  - Replace AND and OR gates with NAND or NOR gates by using Rules 1 – 4, and eliminate double inverters whenever possible

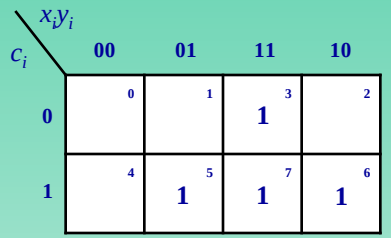
Translation of Standard Terms to NAND and NOR Schematics

Form Type	Standard Form Implementation	NAND Implementation	NOR Implementation
Sum of products			
Product of sums			

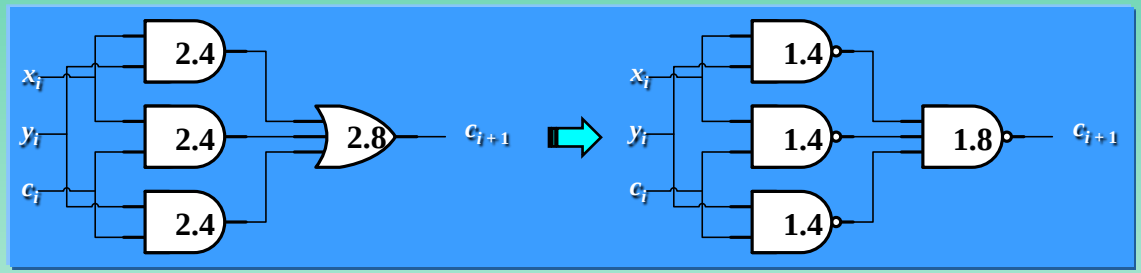
Conversion to NAND (NOR) Gates

Example: Conversion to NAND (NOR) gates

Problem: Derive the NAND and NOR implementations of the carry function



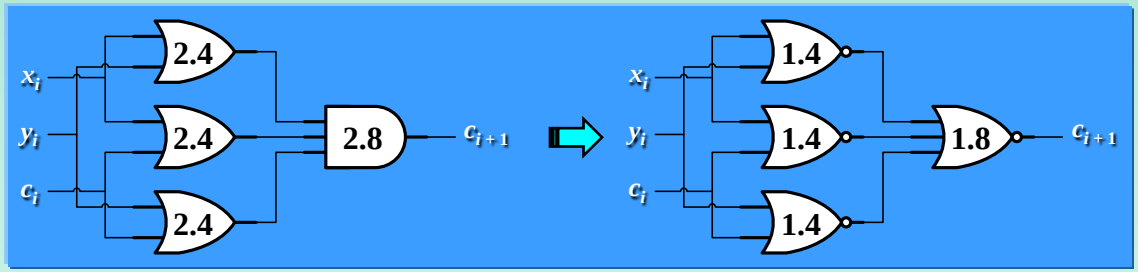
Map Definition Carry Function c_{i+1}



NAND Implementation

$$c_{i+1} = x_iy_i + x_ic_i + y_ic_i$$
$$c_{i+1} = (x_i + y_i)(x_i + c_i)(y_i + c_i)$$

Standard Forms

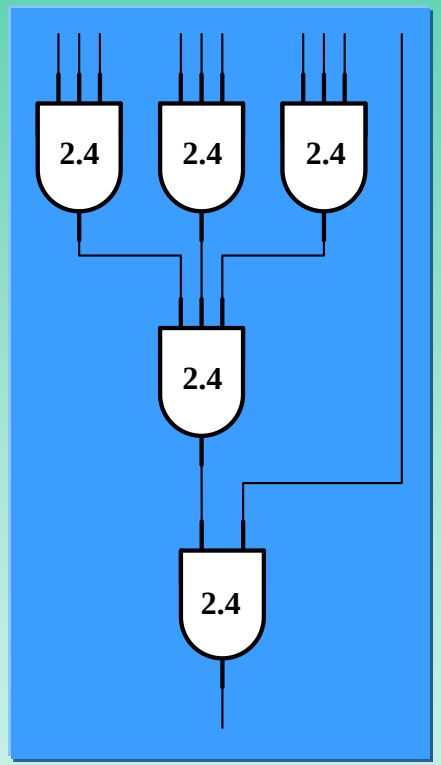


NOR Implementation

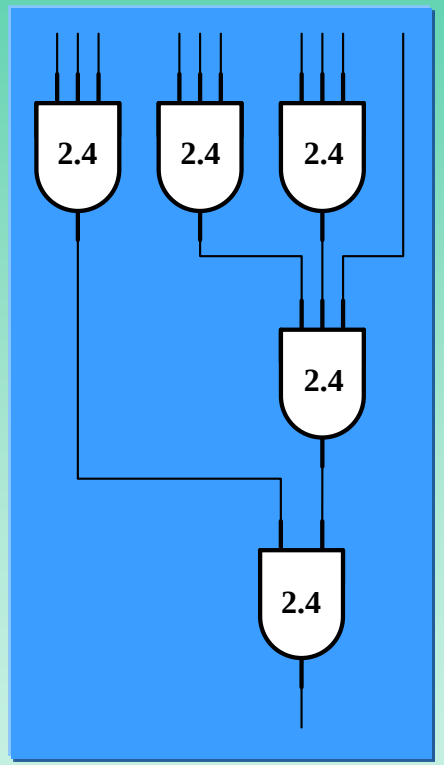
Decomposition of 10–input into 3–input Gates

Level Number	Number of Inputs	Number of Gates
1	10	[10 / 3] = 3
2	3 + (10 – 3([10 / 3])) = 4	[4 / 3] = 1
3	1 + (4 – 3([4 / 3])) = 2	[2 / 3] = 1

Input and Gate Computation on Each Level



One Possible Decomposition



Alternative Decomposition

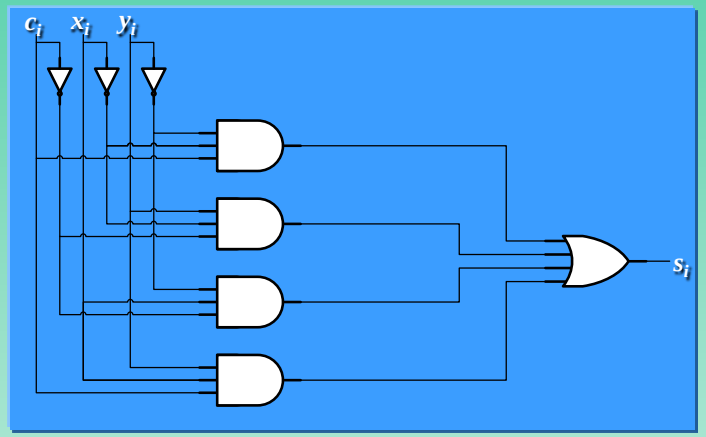
Technology Mapping for Gates

Example: Technology mapping for gates

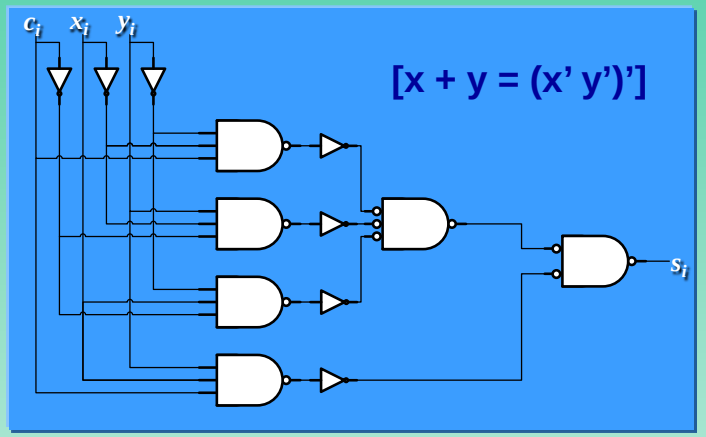
Problem: Implement the sum function with 3–input NAND gates

c_i \ $x_i y_i$	00	01	11	10
0	0	1	3	1
1	1	5	1	6

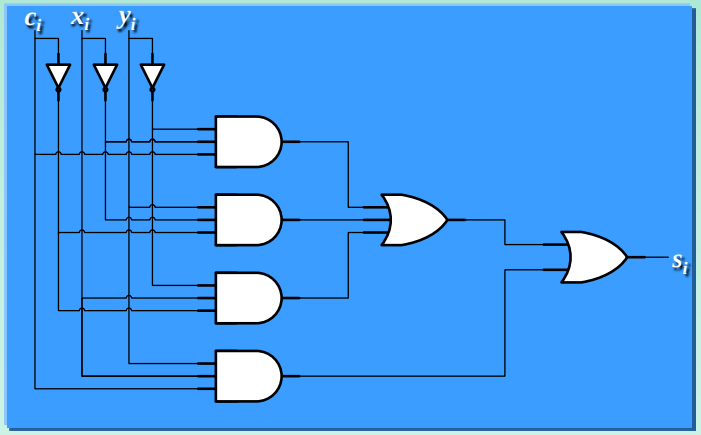
Map for Sum Function s_i



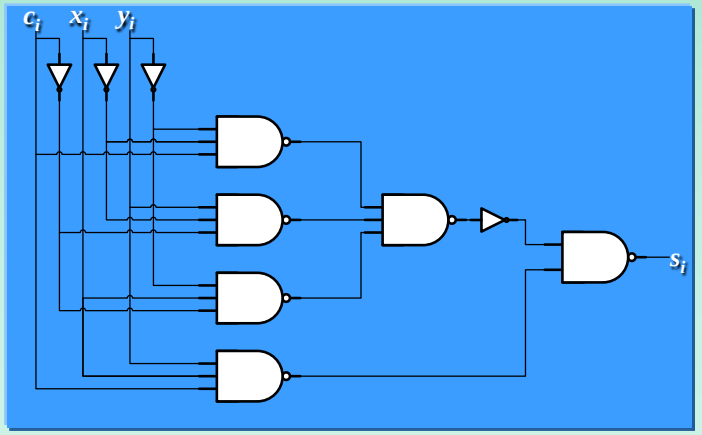
AND–OR Implementation



Conversion to NAND Network



OR Gate Decomposition

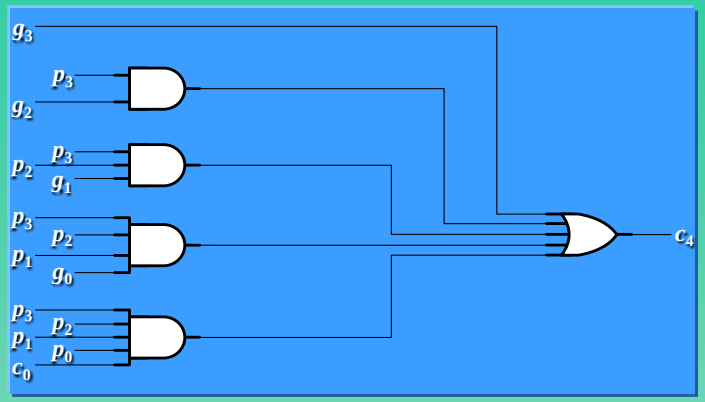


Optimized NAND Network

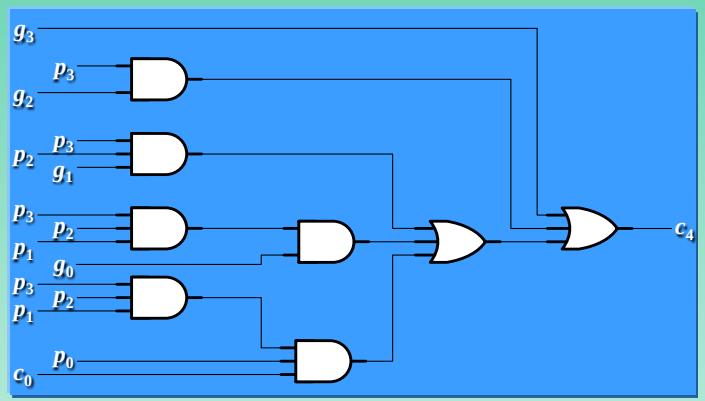
Design Retiming

Example: Design retiming

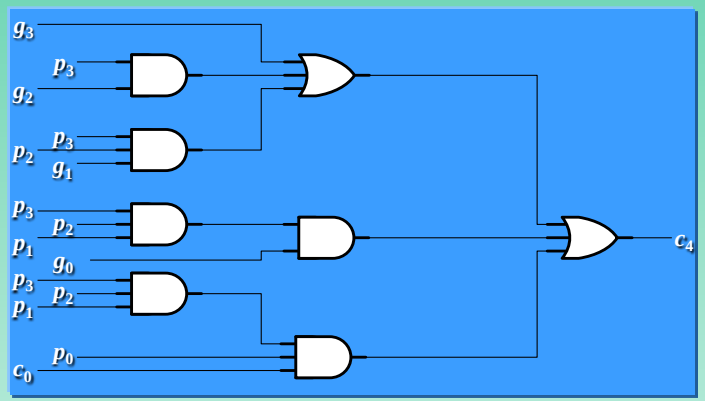
Problem: Implement 4–bit carry-look-ahead function using 3–input NAND gates



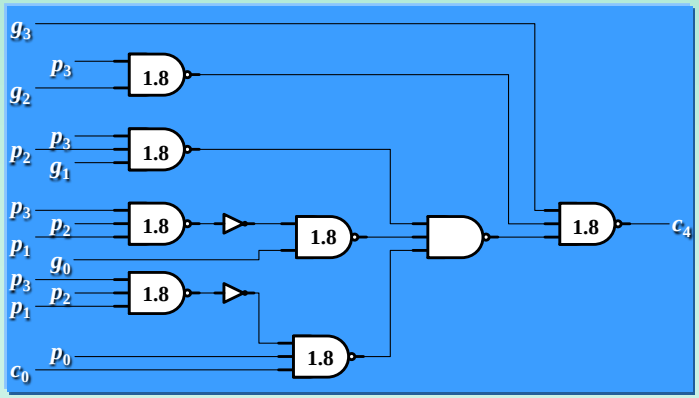
AND-OR Implementation



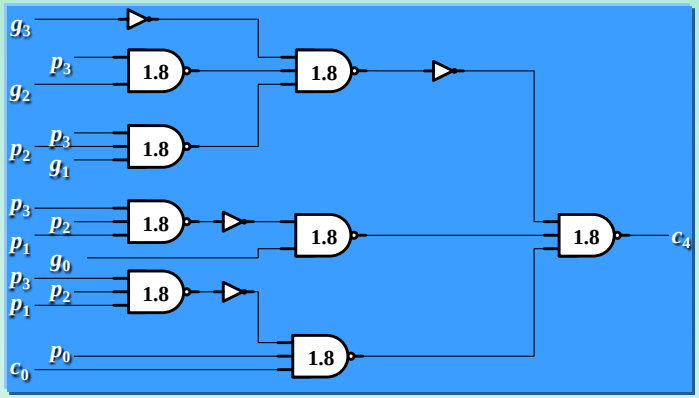
Decomposition of AND-OR Implementation



Performance Optimized Decomposition

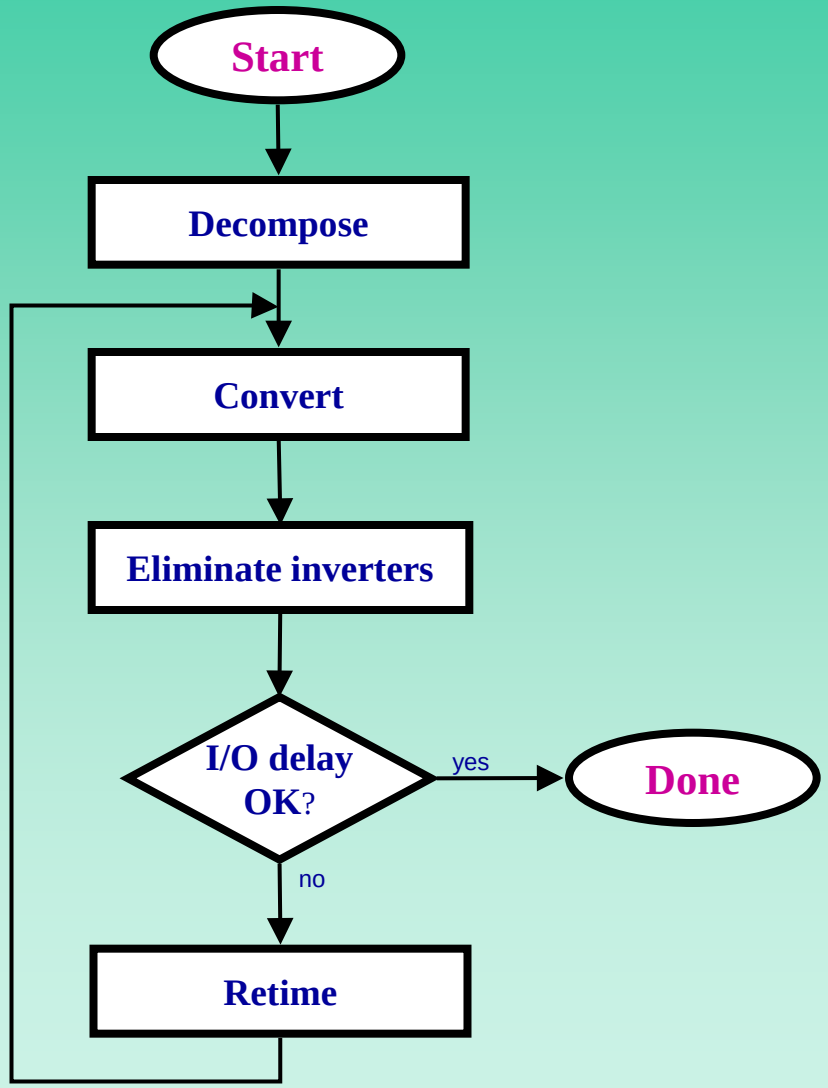


NAND Implementation of Above, Delay = 8.2ns



Performance Optimized NAND Implementation, Delay = 6.4ns

Technology Mapping Procedure



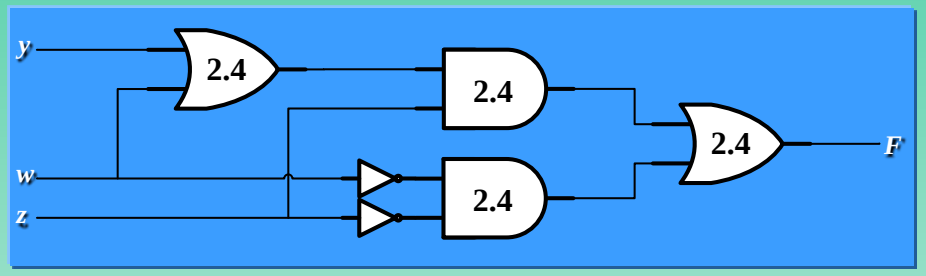
Technology Mapping for Custom Libraries

- Libraries contain gates with different functions and different delays
- Technology mapping means covering schematic with library gates
- Primary goal: Minimize delay on critical paths
- Secondary goal: Minimize cost on non-critical paths

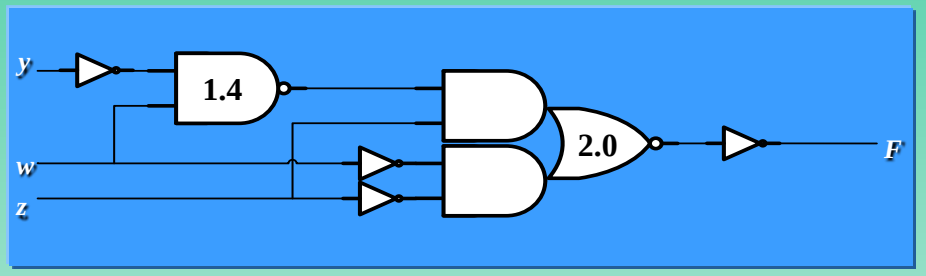
Mapping for Custom Libraries

Example: Technology mapping for custom libraries

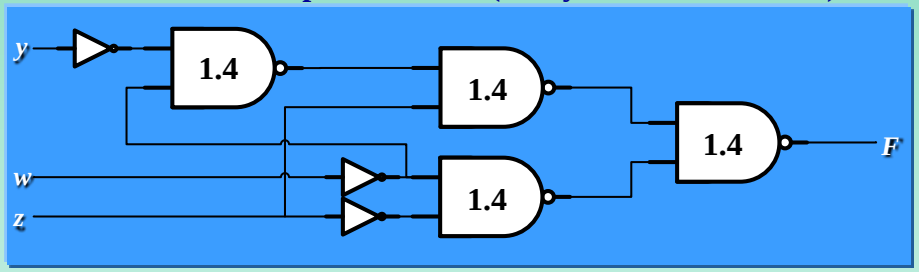
Problem: Implement the expression $w'z' + z(w + y)$ with the logic gates defined earlier



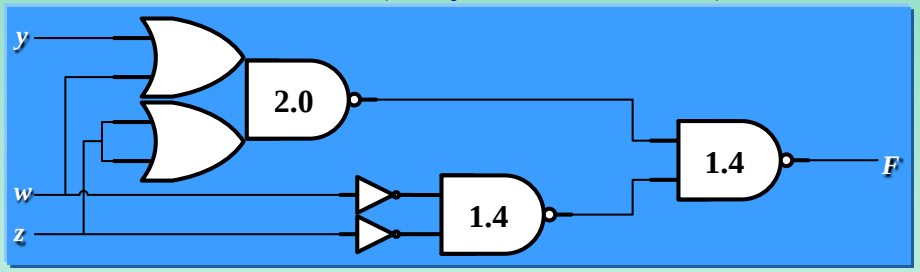
AND–OR Implementation (Delay = 7.2ns, Cost = 28)



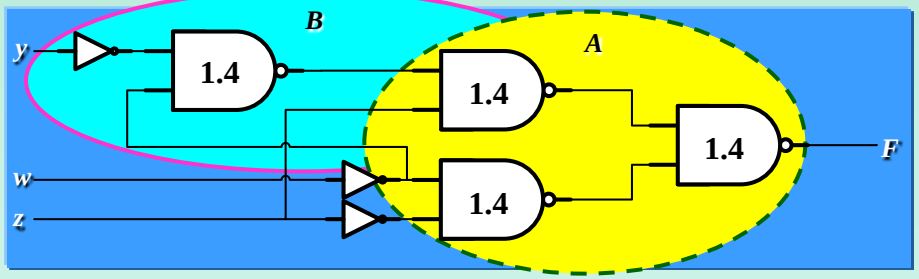
Alternative A (Delay = 5.4ns, Cost = 20)



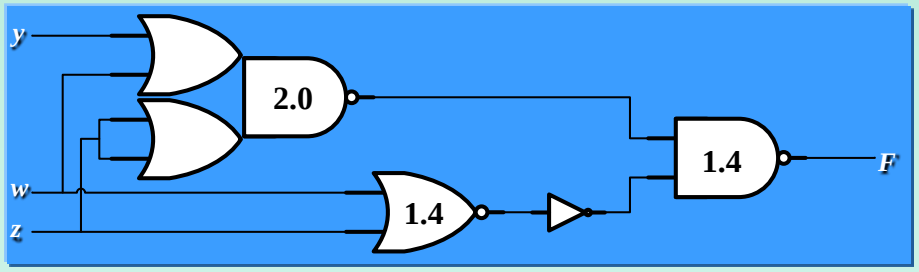
NAND Implementation (Delay =5.2ns, Cost = 22)



Alternative B (Delay = 3.8ns, Cost = 20)

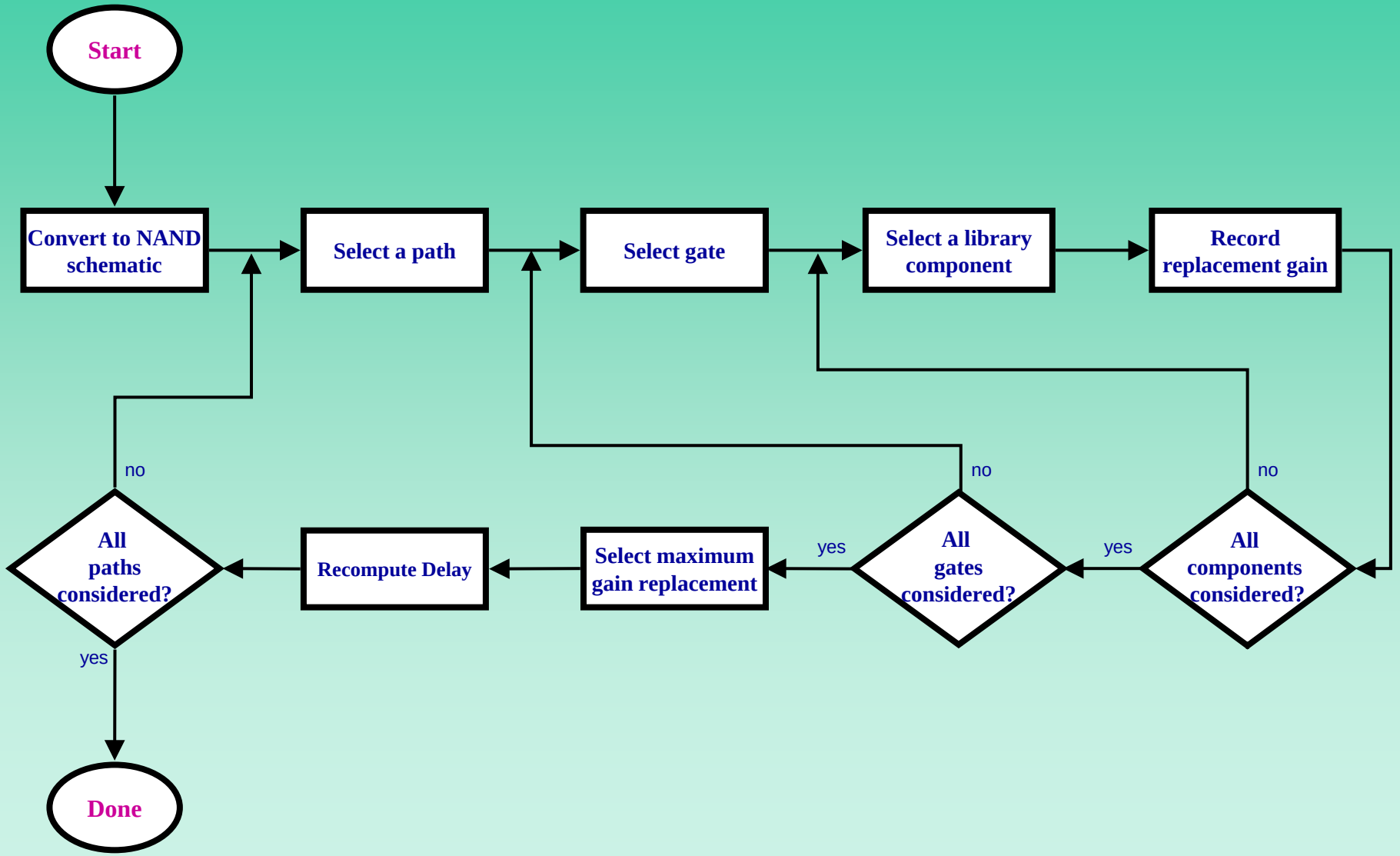


Two Possible Conversions



Cost Optimized Alternative B (Delay = 3.8ns, Cost = 18)

Conversion Procedure for Custom Libraries



Summary

- Technology mapping for NAND/NOR gates
  - Decomposition
  - Conversion
  - Optimization
  - Retiming
- Technology mapping for custom libraries by schematic covering with complex gates with
  - Time optimization on circuit paths
  - Cost optimization on non-critical paths