Quine-McCluskey Procedure
(Class 4.1 – 2/5/2013)

CSE 2441 – Introduction to Digital Logic

Spring 2013

Instructor – Bill Carroll, Professor of CSE

Today's Topics

- Quine-McCluskey procedure
- Combinational circuit analysis

Quine-McCluskey Minimization Method

- Advantages over K-maps
  - Can be computerized
  - Can handle functions of more than six variables
  - Can be adapted to minimize multiple functions
- Overview of the method
  - Given the minterms of a function
  - Find all prime implicants (steps 1 and 2)
    - Partition minterms into groups according to the number of 1's
    - Exhaustively search for prime implicants
  - Find a minimum prime implicant cover (steps 3 and 4)
    - Construct a prime implicant chart
    - Select the minimum number of prime implicants
  - Note – the method can also be described for maxterms and implicates

Example 3.24 -- Use the Q-M method to find the MSOP of the function

$f(A,B,C,D) = \sum m(2,4,6,8,9,10,12,13,15)$

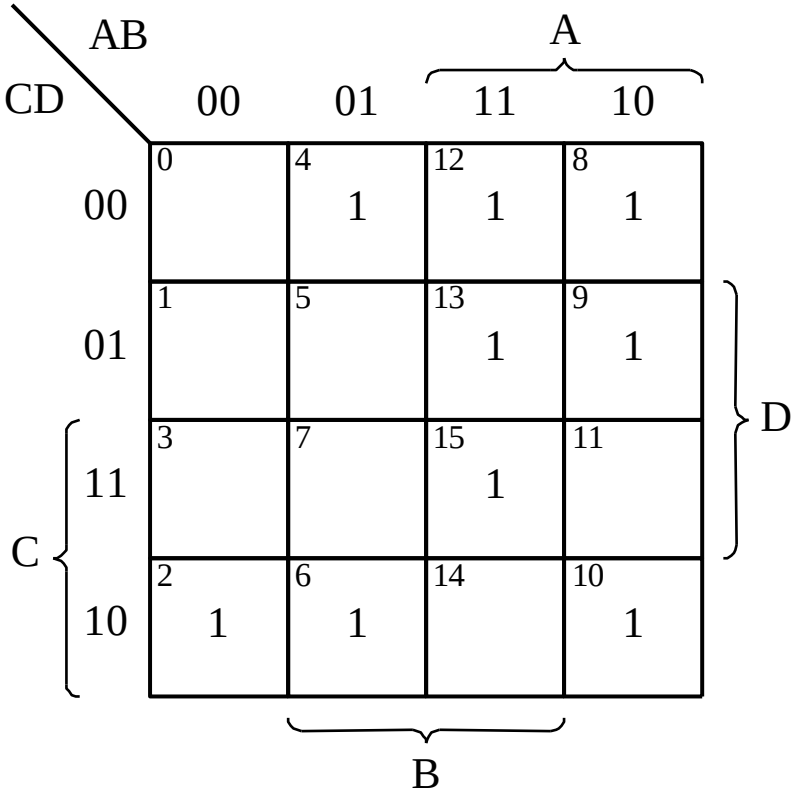


Figure 3.32 K-map for example 3.30.

Step 1 -- List Prime Implicants in Groups (Example 3.24)

Minterms	$ABCD$	
2	0010	
4	0100	Group 1 (a single 1)
8	1000	
6	0110	
9	1001	Group 2 (two 1's)
10	1010	
12	1100	
13	1101	Group 3 (three 1's)
15	1111	Group 4 (four 1's)

Step 2 -- Generate Prime Implicants (Example 3.24)

List 1			List 2			List 3		
Minterm	*ABCD*		Minterms	*ABCD*		Minterms	*ABCD*	
2	0010	√	2, 6	0-10	PI_2	8, 9, 12, 13	1-0-	PI_1
4	0100	√	2, 10	-010	PI_3			
8	1000	√	4, 6	01-0	PI_4			
6	0110	√	4, 12	-100	PI_5			
9	1001	√	8, 9	100-	√			
10	1010	√	8, 10	10-0	PI_6			
12	1100	√	8, 12	1-00	√			
13	1101	√	9, 13	1-01	√			
15	1111	√	12, 13	110-	√			
			13, 15	11-1	PI_7			

Step 3 -- Prime Implicant Chart (Example 3.24)

	2	4	6	√ 8	√ 9	10	√ 12	√ 13	√ 15
* * PI_1				×	⊗		×	×	
PI_2	×		×						
PI_3	×					×			
PI_4		×	×						
PI_5		×					×		
PI_6				×		×			
* * PI_7								×	⊗

Step 4 -- Reduced Prime Implicant Chart (Example 3.24)

	√ 2	√ 4	√ 6	√ 10
PI_2	×		×	
$*PI_3$	×			×
$*PI_4$		×	×	
PI_5		×		
PI_6				×

The Resulting Minimal Realization of *f*

$$f(A,B,C,D) = \mathrm{PI}_1 + \mathrm{PI}_3 + \mathrm{PI}_4 + \mathrm{PI}_7$$

$$= 1\text{-}0\text{-} + \text{-}010 + 01\text{-}0 + 11\text{-}1$$

$$= AC' + B'CD' + A'BD' + ABD$$

How the Q-M Results Look on a K-map

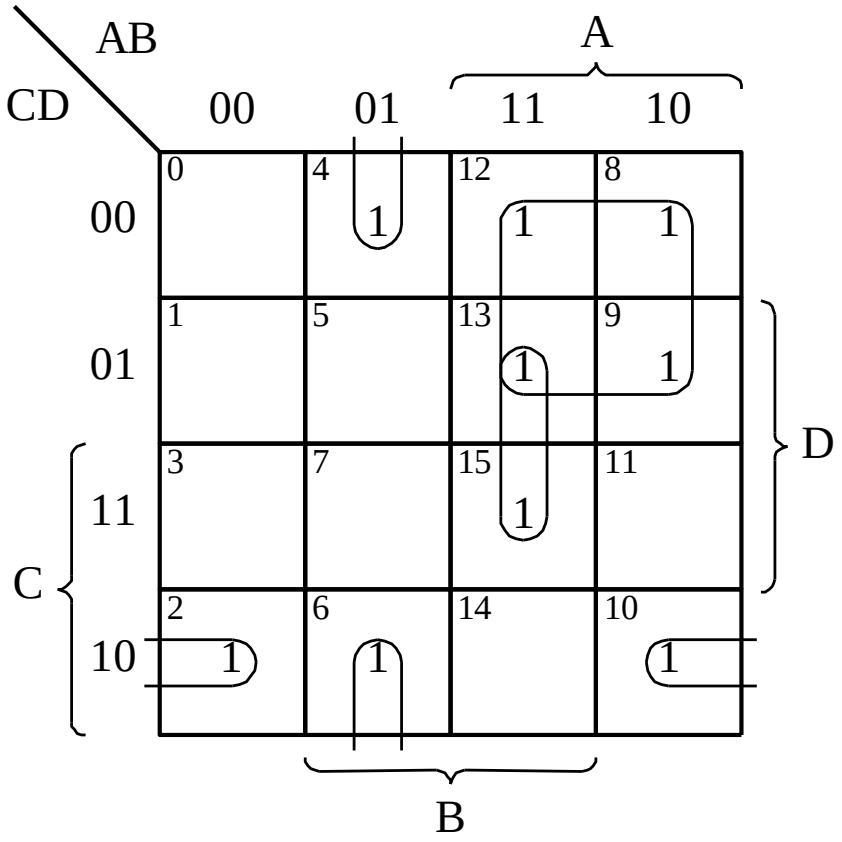


Figure 3.33 Grouping of terms.

Covering Procedure

Step 1 -- Identify any minterms covered by only one PI. Select these PIs for the cover.

Step 2 -- Remove rows covered by the PIs identified in step 1. Remove minterms covered by the removed rows.

Step 3 -- If a cyclic chart results from step 2, go to step 5. Otherwise, apply the reduction procedure of steps 1 and 2.

Step 4 -- If a cyclic chart results from step 3, go to step 5. Otherwise return to step 1.

Step 5 -- Apply the cyclic chart procedure. Repeat step 5 until a void chart or noncyclic chart chart is produced. In the latter case, return to step 1.

Coverage Example

$f(A,B,C,D) = \sum m(0,1,5,6,7,8,9,10,11,13,14,15)$

	√ 0	√ 1	5	√ 6	√ 7	√ 8	√ 9	10	11	13	√ 14	√ 15
* * PI_1	⊗	×				×	×					
PI_2		×	×				×			×		
PI_3			×		×					×		×
PI_4						×	×	×	×			
PI_5							×		×	×		×
PI_6								×	×		×	×
* * PI_7				⊗	×						×	×

Reduced PI Charts

	5	10	11	13
PI_2	×			×
PI_3	×			×
PI_4		×	×	
PI_5			×	×
PI_6		×	×	

	√ 5	√ 10
* PI_2	×	
*PI_4		×

Cyclic PI Charts

1. No essential PIs.
2. No row or column coverage.

	√ 1	2	√ 3	4	5	6
*PI_1	×		×			
PI_2		×	×			
PI_3		×				×
PI_4				×		×
PI_5				×	×	
PI_6	×				×	

	2	4	5	6
PI_2	×			
PI_3	×			×
PI_4		×		×
PI_5		×	×	
PI_6			×	

	√ 2	√ 4	√ 5	√ 6
*PI_3	×			×
PI_4		×		×
*PI_5		×	×	

Using the Q-M Method with Incompletely Specified Functions

1. Use minterms and don't cares when generating prime implicants

2. Use only minterms when finding a minimal cover

Example 3.25 -- Find a minimal sum of products of the following function using the Quine-McCluskey procedure.

$$f(A,B,C,D,E) = m(2,3,7,10,12,15,27) + d(5,18,19,21,23)$$

Minimizing Table for Example 3.25

List 1			List 2			List 3		
Minterm	$ABCDE$		Minterms	$ABCDE$		Minterms	$ABCDE$	
2	00010	√	2, 3	0001-	√	2, 3, 18, 19	-001-	PI_1
3	00011	√	2, 10	0-010	PI_4	3, 7, 19, 23	-0-11	PI_2
5	00101	√	2, 18	-0010	√	5, 7, 21, 23	-01-1	PI_3
10	01010	√	3, 7	00-11	√			
12	01100	PI_7	3, 19	-0011	√			
18	10010	√	5, 7	001-1	√			
7	00111	√	5, 21	-0101	√			
19	10011	√	18, 19	1001-	√			
21	10101	√	7, 15	0-111	PI_5			
15	01111	√	7, 23	-0111	√			
23	10111	√	19, 23	10-11	√			
27	11011	√	19, 27	1-011	PI_6			
			21, 23	101-1	√			

PI Chart for Example 3.25

	√ 2	3	√ 7	√ 10	√ 12	√ 15	√ 27
PI_1	×	×					
PI_2		×	×				
PI_3			×				
* * PI_4	×			⊗			
* * PI_5			×			⊗	
* * PI_6							⊗
* * PI_7					⊗		

Results of Minimization for Example 3.25

$$f(A,B,C,D,E) = PI_1 + PI_4 + PI_5 + PI_6 + PI_7 \ \textbf{OR}$$

$$= PI_2 + PI_4 + PI_5 + PI_6 + PI_7$$

Basic Knowledge Self Assessment

- 2's complement number systems
- Alphanumeric codes – BCD, ASCII
- Basic logic gates
- Logic functions
  - Truth tables
  - Algebraic expressions
  - Canonical forms
    - SOP
    - POS
    - Minterm/maxterm lists
  - Minimum forms
    - MSOP
    - MPOS
  - Minimization procedures
    - Boolean algebra
    - Karnaugh maps
    - Quine-McCluskey

Analysis of Combinational Circuits (1)

- Digital Circuit ***Design***:
  - Word description of a function
    $\Rightarrow$ a set of switching equations
    $\Rightarrow$ hardware realization (gates, programmable logic devices, etc.)

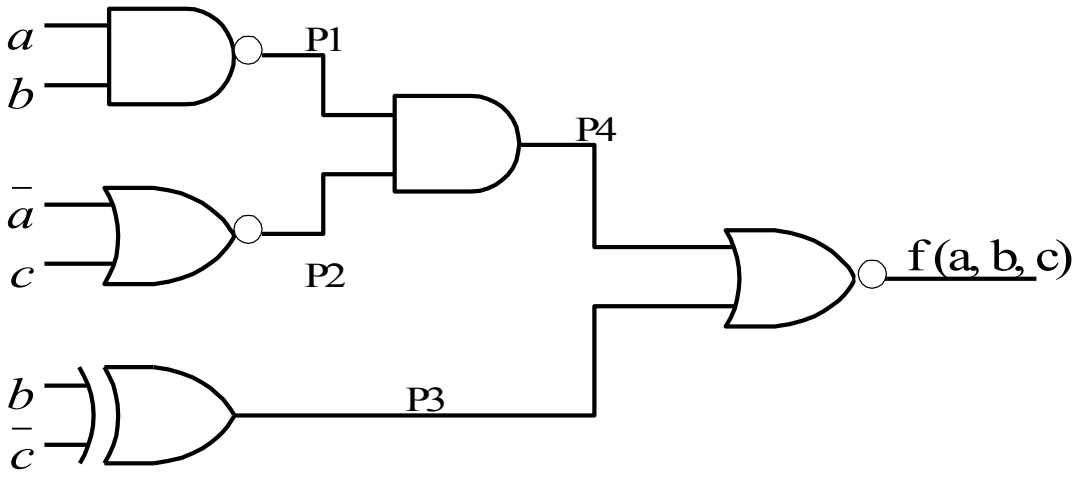
- Digital Circuit ***Analysis***:
  - Hardware realization
    $\Rightarrow$ switching expressions, truth tables, timing diagrams, etc.

- Analysis is used
  - To determine the behavior of the circuit
  - To verify the correctness of the circuit
  - To assist in converting the circuit to a different form.

Analysis of Combinational Circuits (2)

- ***Algebraic Method***: Use switching algebra to derive a desired form.

- ***Example 2.33***: Find a simplified switching expression and logic circuit for the following circuit (Fig. 2.21a).



(a)

Analysis of Combinational Circuits (3)

- Write switching expression for each gate output:

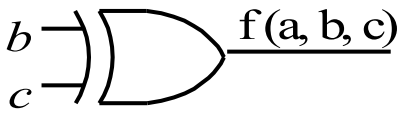
$$P_1 = \overline{ab}, \quad P_2 = \overline{a+c}, \quad P_3 = b \oplus \bar{c}, \quad P_4 = P_1 \cdot P_2 = \overline{ab} \cdot \overline{(\bar{a}+c)}$$

- The output is: $f(a,b,c) = \overline{P_3 + P_4} = \overline{(b \oplus \bar{c}) + \overline{ab} \cdot \overline{(\bar{a}+c)}}$
- Simplify the output function using switching algebra:

$$\begin{aligned} \bar{f}(a,b,c) &= (b \oplus \bar{c}) + \overline{ab} \cdot \overline{\bar{a}+c} \\ &= bc + \bar{b}\bar{c} + \overline{ab} \cdot \overline{\bar{a}+c} && \text{[Eq. 2.24]} \\ &= bc + \bar{b}\bar{c} + (\bar{a}+\bar{b})a\bar{c} && \text{[T8]} \\ &= bc + \bar{b}\bar{c} + a\bar{b}\bar{c} && \text{[T5(b)]} \\ &= bc + \bar{b}\bar{c} && \text{[T4(a)]} \end{aligned}$$

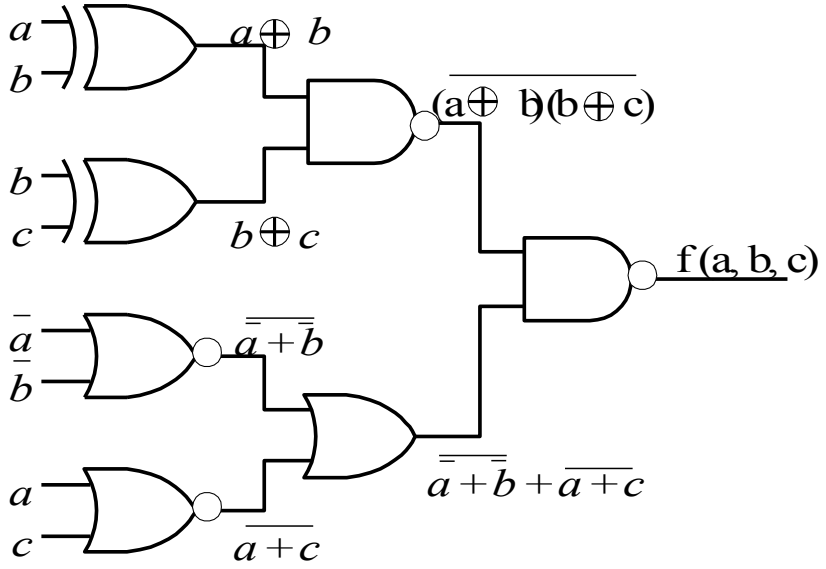
$\bar{f}(a,b,c) = b \odot c$ [Eq. 2.32]

Therefore, $f(a,b,c) = (b \odot c)' = b \oplus c$



Analysis of Combinational Circuits (4)

- ***Example 2.34***: Find a simplified switching expression and logic network for the following logic circuit (Fig. 2.22).



Given circuit

Analysis of Combinational Circuits (5)

- Derive the output expression:

$f(a,b,c)$

$$= \overline{\overline{(a \oplus b)(b \oplus c)} \cdot \overline{(\bar{a} + \bar{b} + \overline{a + c})}}$$

$$= \overline{(a \oplus b)(b \oplus c)} + \overline{\bar{a} + \bar{b} + \overline{a + c}}) \quad \text{[T8(b)]}$$

$$= (a \oplus b)(b \oplus c) + (\bar{a} + \bar{b})(a + c) \quad \text{[T8(a)]}$$

$$= (a\bar{b} + \bar{a}b)(b\bar{c} + \bar{b}c) + (\bar{a} + \bar{b})(a + c) \quad \text{[Eq. 2.24]}$$

$$= a\bar{b}b\bar{c} + a\bar{b}\bar{b}c + \bar{a}bb\bar{c} + \bar{a}b\bar{b}c + \bar{a}a + \bar{a}c + a\bar{b} + \bar{b}c \quad \text{[P5(b)]}$$

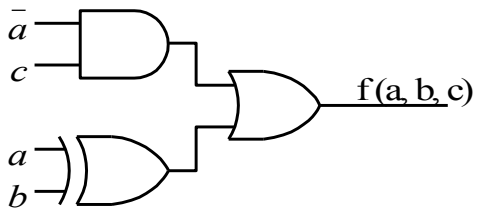
$$= a\bar{b}c + \bar{a}b\bar{c} + \bar{a}c + a\bar{b} + \bar{b}c \quad \text{[P6(b), T4(a)]}$$

$$= \bar{a}b\bar{c} + \bar{a}c + a\bar{b} + \bar{b}c \quad \text{[T4(a)]}$$

$$= \bar{a}b\bar{c} + \bar{a}c + a\bar{b} \quad \text{[T9(a)]}$$

$$= \bar{a}b + \bar{a}c + a\bar{b} \quad \text{[T7(a)]}$$

$$= \bar{a}c + a \oplus b \quad \text{[Eq. 2.24]}$$



Simplified circuit

Analysis of Combinational Circuits (6)

- ***Truth Table Method***: Derive the truth table one gate at a time.
- The truth table for Example 2.34:

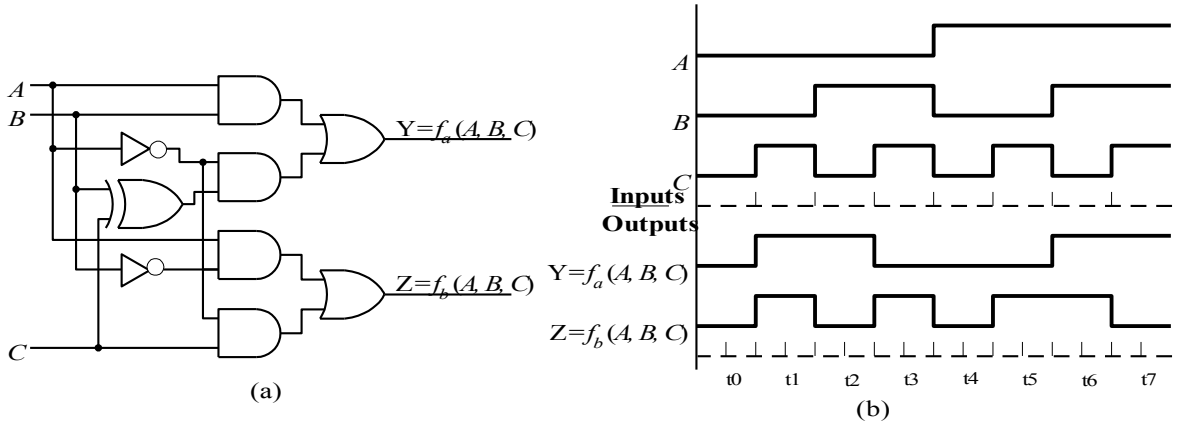
abc	$\bar{a}c$	$a \oplus b$	$f(a,b,c)$
000	0	0	0
001	1	0	1
010	0	1	1
011	1	1	1
100	0	1	1
101	0	1	1
110	0	0	0
111	0	0	0

Analysis of Combinational Circuits (7)

- ***Analysis of Timing Diagrams***
  - The *Timing diagram* is a graphical representation of input and output signal relationships over time.
  - Timing diagrams may show intermediate signals and propagation delays.

Analysis of Combinational Circuits (8)

- ***Example 2.35***: Derivation of truth table from a timing diagram



Time	Inputs *ABC*	Outputs $f_a(A, B, C)$	Outputs $f_b(A, B, C)$
t0	0 0 0	0	0
t1	0 0 1	1	1
t2	0 1 0	1	0
t3	0 1 1	0	1
t4	1 0 0	0	0
t5	1 0 1	0	1
t6	1 1 0	1	1
t7	1 1 1	1	0

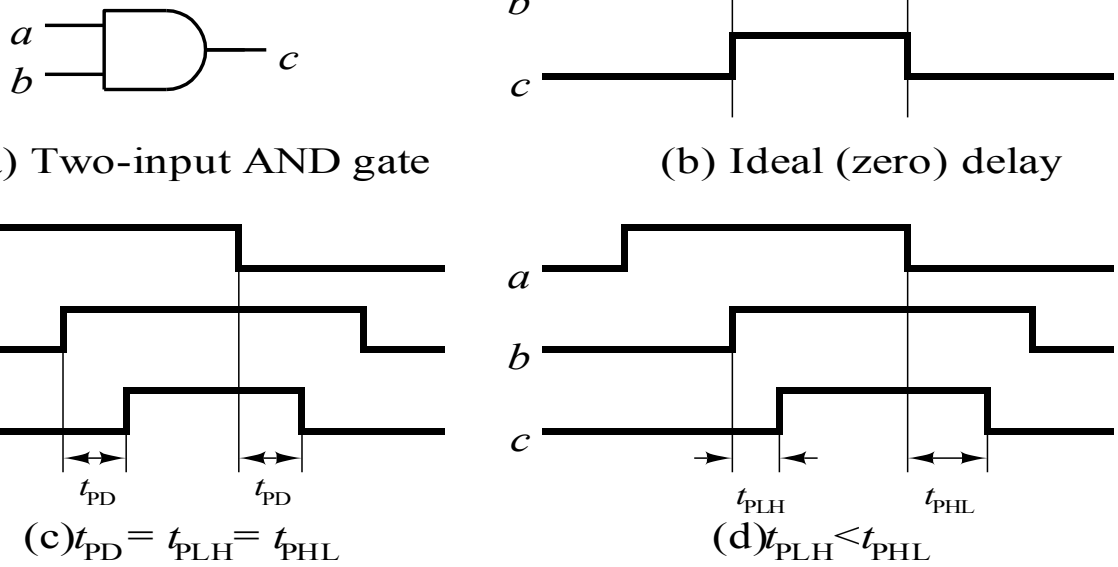
(c)

Analysis of Combinational Circuits (9)

- ***Propagation Delay***
  - Physical characteristics of a logic circuit to be considered:
    - Propagation delays
    - Gate fan-in and fan-out restrictions
    - Power consumption
    - Size and weight
  - *Propagation delay*: The delay between the time of an input change and the corresponding output change.
  - Typical two propagation delay parameters:
    - t_{PLH} = propagation delay time, low-to-high-level output
    - t_{PHL} = propagation delay time, high-to-low-level output
  - Approximation: $$t_{PD} = \frac{t_{PLH} + t_{PHL}}{2}$$

Analysis of Combinational Circuits (10)

- Propagation delay through a logic gate



Analysis of Combinational Circuits (11)

- Power dissipation and propagation delays for several logic families (Table 2.7)

Logic Family	Propagation Delay t_{PD}(ns)	Power Dissipation Per Gate (mW)	Technology
7400	10	10	Standard TTL
74H00	6	22	High-speed TTL
74L00	33	1	Low-power TTL
74LS00	9.5	2	Low-power Schottky TTL
74S00	3	19	Schottky TTL
74ALS00	3.5	1.3	Advanced low-power Schottky TTL
74AS00	3	8	Advanced Schottky TTL
74HC00	8	0.17	High-speed CMOS

Analysis of Combinational Circuits (12)

- Propagation delays of primitive 74LS series gates (Table 2.8)

Chip	Function	t_{PLH}		t_{PHL}	
		Typical	Maximum	Typical	Maximum
74LS04	NOT	9	15	10	15
74LS00	NAND	9	15	10	15
74LS02	NOR	10	15	10	15
74LS08	AND	8	15	10	20
74LS32	OR	14	22	14	22