Logic and Computer Design Fundamentals

Term Review

Ming Cai
cm@zju.edu.cn
College of Computer Science and Technology
Zhejiang University

Items of Final Exam

1. Fill in the blank
2. Chose the best Choice
3. Verilog
4. Kaunaugh Map and Optimization
5. Circuit Analysis
   - Combinational circuit
   - Sequential circuit
6. Logic Design
   - Combinational circuit
   - Sequential circuit

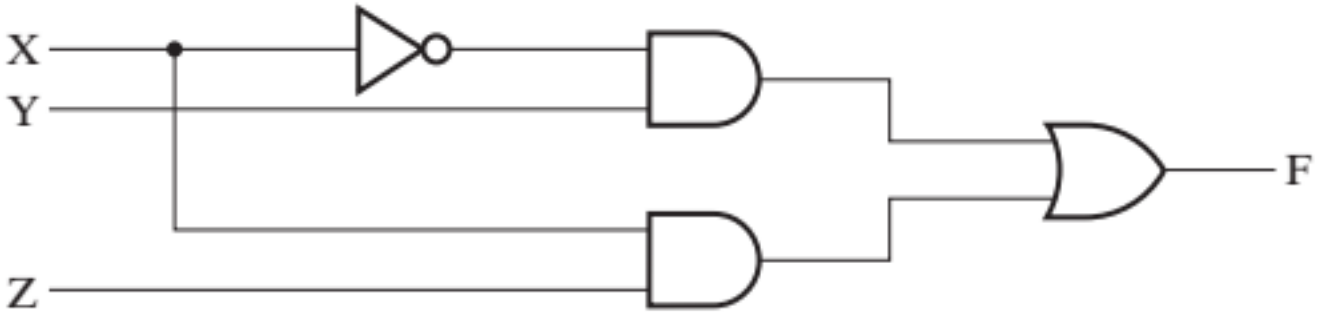
Some Tips for Final Exam

1. Need to understand the terminology!

2. 答题纸内容，一定要字迹整洁、清晰，不然自己吃亏！

Verilog Example

- Please use Verilog to describe the following circuit. (No simplification is needed!)




```
module  test(X, Y, Z, F)
input   X, Y, Z;
output F;

assign F = ____~X&Y | X&Z____;

endmodule
```

Course Review

HIGHLIGHTS & PROBLEMS

Chapter 1

- Conversion between number systems
  - Binary number, hexadecimal number
    - Eg. $(1101\ 1111)_2 = (DF)_{16}$

Decimal	Binary	Octal	Hexadecimal
369.3125	101110001.0101	561.24	171.5
189.625	10111101.101	275.5	BD.A
214.625	11010110.101	326.5	D6.A
62407.625	1111001111000111.101	171707.5	F3C7.A

- Conversion between binary number and decimal code
  - BCD (binary-coded decimal)
    - Eg. $(1001\ 0101)_{BCD} \rightarrow (95)_{10} = (0101\ 1111)_2 = (5F)_{16}$
  - Parity Bit
    - 100 0001 → 0100 0001 (with even parity) → 1100 0001 (with odd parity)
    - How to generate odd parity bit P for any 5-bit binary number $D_4D_3D_2D_1D_0$?
- Gray Code & ASCII Character Code

Chapter 2

- Boolean algebra
  - DeMorgan's law: $\overline{(X + Y)} = \overline{X}\,\overline{Y}$ and $\overline{(XY)} = \overline{X} + \overline{Y}$
  - Distributive laws: $X+YZ=(X+Y)(X+Z)$
  - Dual of an algebraic expression: OR ←→AND, 0's ←→ 1's
  - Consensus theorem: $XY+\overline{X}Z+YZ = XY+\overline{X}Z$, $(X+Y)(\overline{X}+Z)(Y+Z)=(X+Y)(\overline{X}+Z)$
- Complement of a function
  - OR ←→AND, 0's ←→ 1's, $X \longleftrightarrow \overline{X}$
- Canonical forms, Standard forms
  - Minterms, Maxterms, Canonical forms (SOM, POM)
  - Product terms, sum terms, SOP, POS
  - Relationship between SOM and POM? SOM and SOP?
- Two-level circuit optimization
  - Cost criteria: gate input cost
  - Karnaugh map (K-map), Prime Implicants, Essential Prime Implicants
  - Simplifying in SOP form (with don't care conditions)

Chapter 2

- Other gates
  - NAND/NOR, XOR/XNOR, Odd/Even Function, Buffer, 3-state buffer
- Exclusive-OR operator and gates
  - Identities of XOR operation:
    - $X \oplus \overline{Y} = \overline{X} \oplus Y = \overline{(X \oplus Y)}$ $X \oplus \overline{(Y \oplus Z)} = \overline{(X \oplus Y)} \oplus Z = \overline{(X \oplus Y \oplus Z)}$
  - Odd function and even function
    - Use odd function to generate even parity bit
    - Use even function to generate odd parity bit
    - The even function is obtained by replacing the output gate with an XNOR gate. $P_{odd} = X \oplus Y \oplus Z$, $P_{even} = \overline{P_{odd}} = \overline{(X \oplus Y \oplus Z)}$
- High-impedance outputs
  - 3-state buffer

EN	IN	OUT
0	X	Hi-Z
1	0	0
1	1	1

Chapter 2

- Problems:

1. The dual of an algebraic expression is obtained by 1) interchanging OR and AND operations and 2) replacing 1's by 0's and 0's by 1's.
2. Use DeMorgan's Theorem to complement a function: 1) interchange AND and OR operators and 2) complement each constant value and literal
3. Four variables odd function has ____C____ "1" squares in its corresponding K-Map.

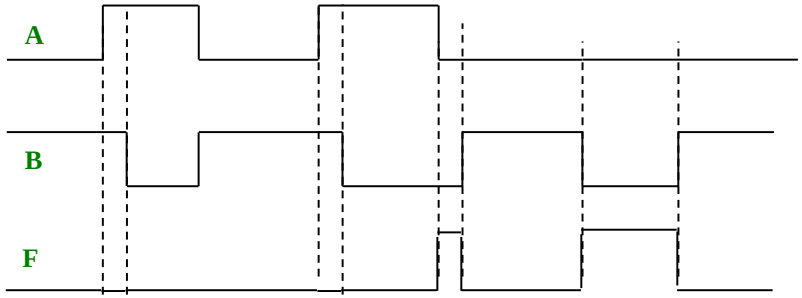
   A. 4 B. 7 C. 8 D. 14
4. The gate input cost G of function $F=AB(C+D)+C(\bar{B}D+\bar{A}D)$ is __A__.

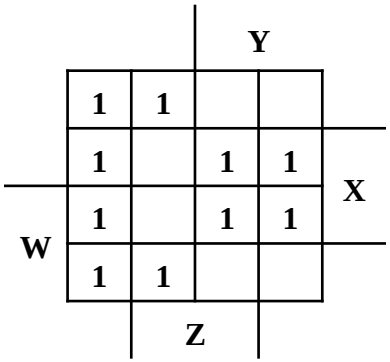
   A. 15 B. 14 C. 13 D. 12
5. Which of the following logical gates can be used as a controllable inverter? ____B____.

   A. AND gate B. XOR gate C. Buffer gate D. OR gate
6. The Essential Prime Implicants in the K-Map given below are ___B___.

   A. Y'Z', XZ' B. X'Y', XY C. XY, XZ' D. Y'Z', X'Y'
7. Given below are the waveforms of input A, B and output F of a logic device. Then the device is a ___A___ gate.

   A. NAND B. NOR C. XOR D. OR

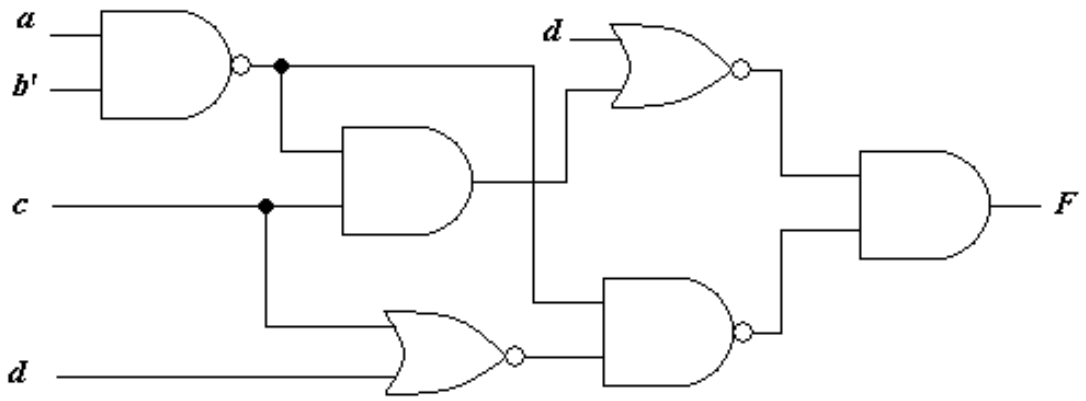




Chapter 2

- Problems:

1. According to the following logic circuit diagram, write down the corresponding Boolean function and optimize it to the form of SOP



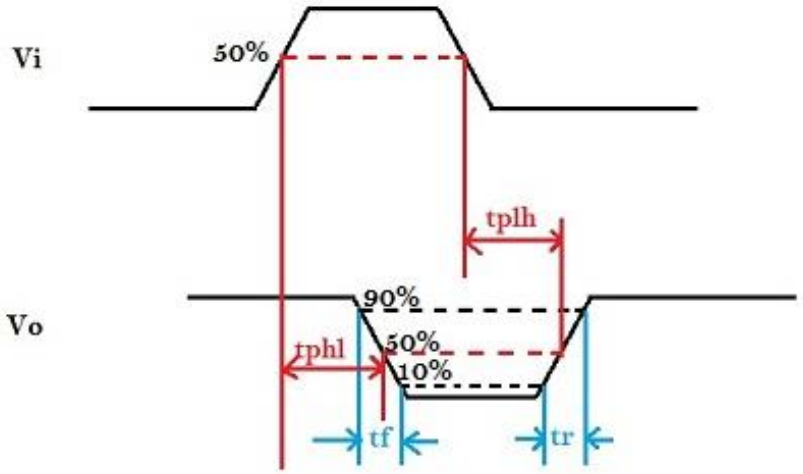
2. Given a Boolean function

$$F(W, X, Y, Z) = \sum m(4, 6, 7, 8, 12, 15) + \sum d(2, 3, 5, 10, 11)$$

Optimize F together with the don't-care conditions d using a K-map

Chapter 3

- Technology parameters
  - fan-in, fan-out, noise margin, cost, transition time, propagation delay, power dissipation

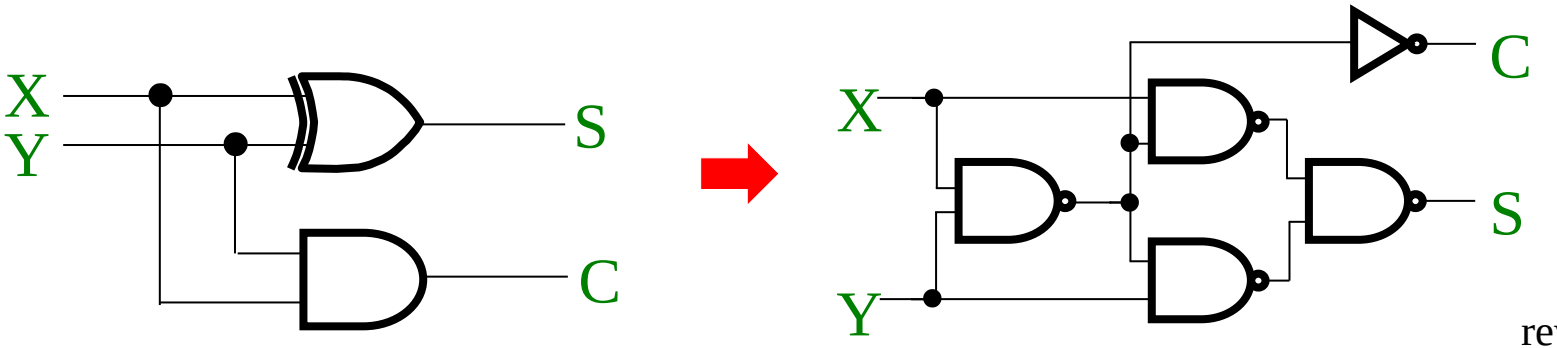


tr = Rise transition time
tf = Fall transition time
tphl = Propagation delay high-low
tplh = Propagation delay low-high

  - Delay Model: transport delay, inertial delay, rejection time
  - How to calculate gate delay based on fan-out?

Chapter 3

- Methods of Describing Logic Events
  - Truth Table, Timing Diagram, Boolean Function, Karnaugh Maps, Logic Circuit
- Design procedure: specification, formulation, optimization, technology mapping, verification
  - Hierarchical Design
- Seven-segment display
  - How to design a BCD-to-Seven-Segment decoder? example 3-2
- Technology mapping
  - How to implement a Boolean function with NAND gates?



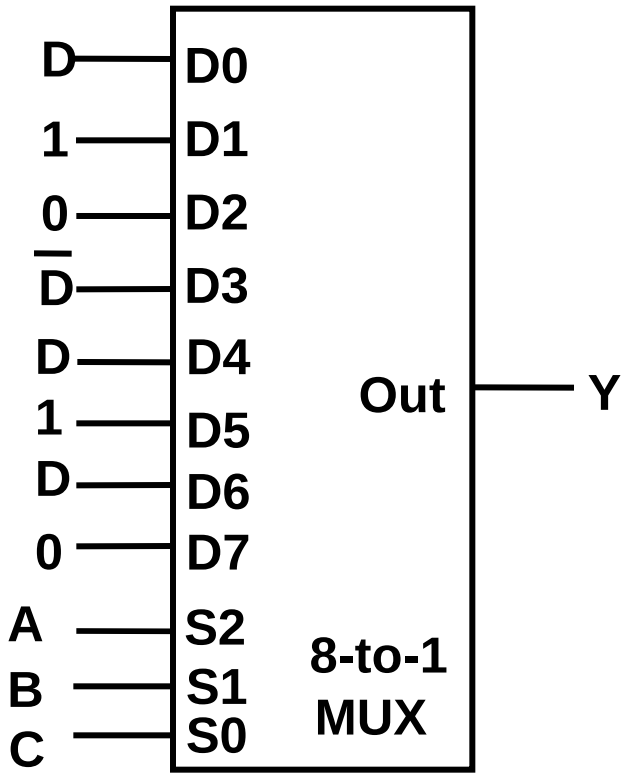
Chapter 3

- n-to-m-Line Decoder
  - n inputs and m outputs with $n \leq m \leq 2^n$
- m-to-n-Line Encoder
  - m inputs and n outputs with $n \leq m \leq 2^n$
- Multiplexer
  - n control inputs (selection inputs), m inputs and one output with $m < 2^n$
- Demultiplexer: Decoder with Enable
- Combinational Function Implementation
  - Decoders and OR gates
  - Multiplexers
  - ROMs
  - PALs: doesn't provide full decoding of the variables, so it doesn't generate all the minterms
  - PLAs: similar to the PALs
  - Lookup Tables

Chapter 3

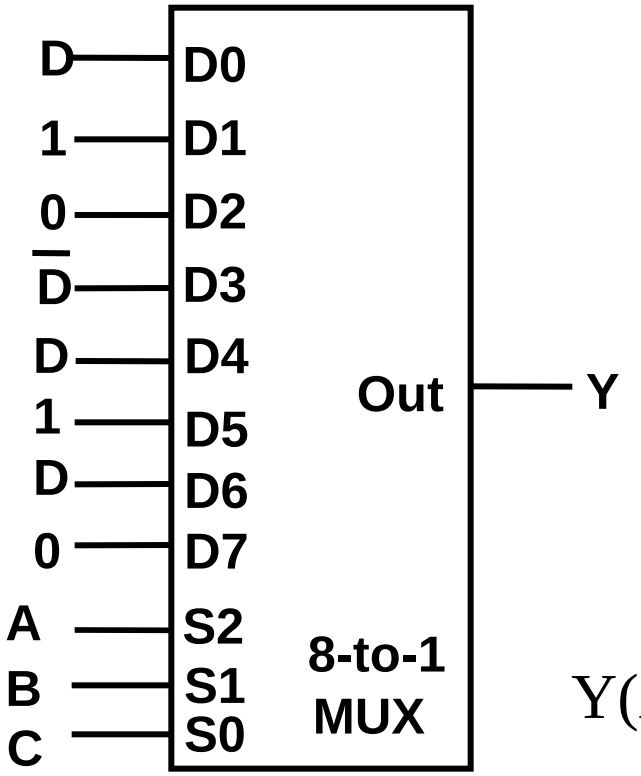
- Problem: Give the canonical sum of product expression for the function which is implemented using the following circuit.

A	B	C	D	Y	
0	0	0	0	0	D
0	0	0	1	1	
0	0	1	0	1	1
0	0	1	1	1	
0	1	0	0	0	0
0	1	0	1	0	
0	1	1	0	1	$\overline{D}$
0	1	1	1	0	
1	0	0	0	0	D
1	0	0	1	1	
1	0	1	0	1	1
1	0	1	1	1	
1	1	0	0	0	D
1	1	0	1	1	
1	1	1	0	0	0
1	1	1	1	0	



$Y(A, B, C, D) = \Sigma_m(1, 2, 3, 6, 9, 10, 11, 13)$

- Problem: Give the canonical sum of product expression for the function which is implemented using the following circuit.



- approach 2

$$Y=\overline{A}\overline{B}\overline{C}D+\overline{A}\overline{B}C1+\overline{A}B\overline{C}0+\overline{A}BC\overline{D}+$$
$$A\overline{B}\overline{C}D+A\overline{B}C1+AB\overline{C}D+ABC0$$
$$=\overline{A}\overline{B}\overline{C}D+\overline{A}\overline{B}C\overline{D}+\overline{A}\overline{B}CD+\overline{A}BC\overline{D}+$$
$$A\overline{B}\overline{C}D+A\overline{B}C\overline{D}+A\overline{B}CD+AB\overline{C}D$$

$$Y(A, B, C, D) = \Sigma_m(1, 2, 3, 6, 9, 10, 11, 13)$$

Chapter 3

- Problem: Find the logic expression of output F in the figure.

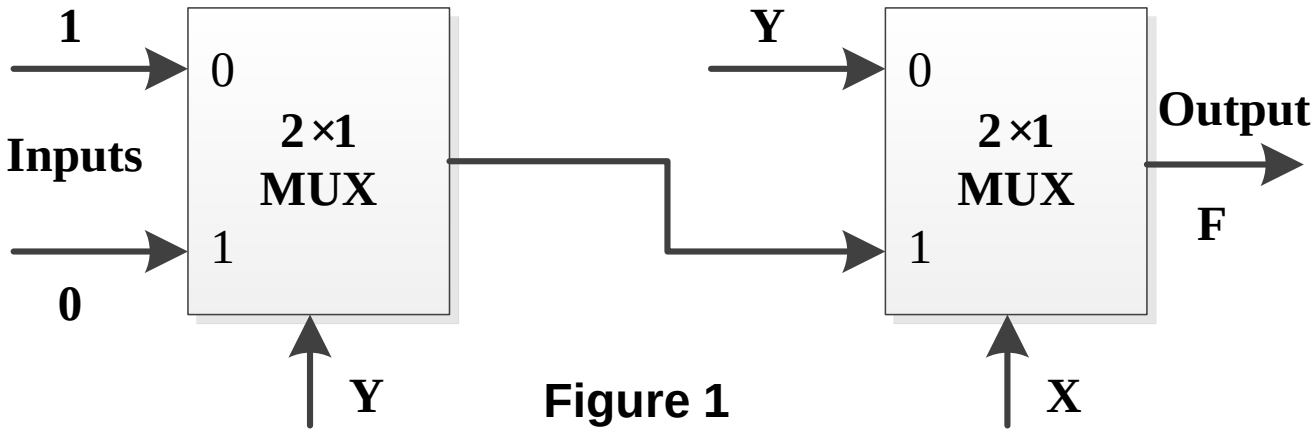


Figure 1

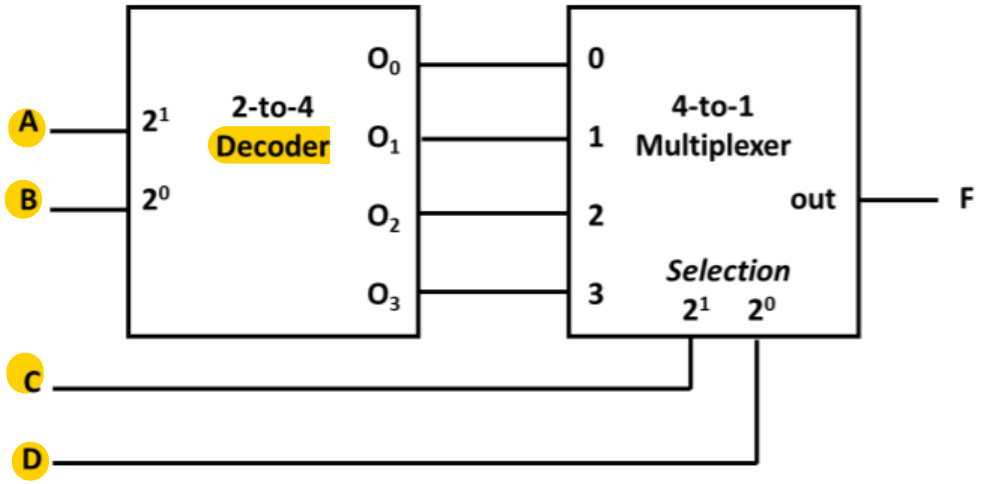
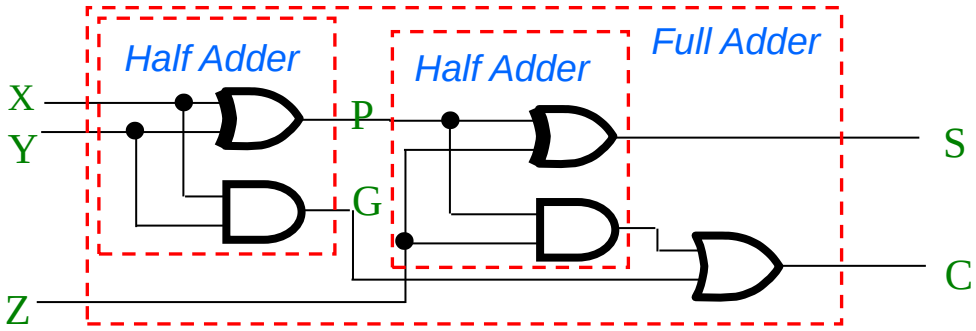
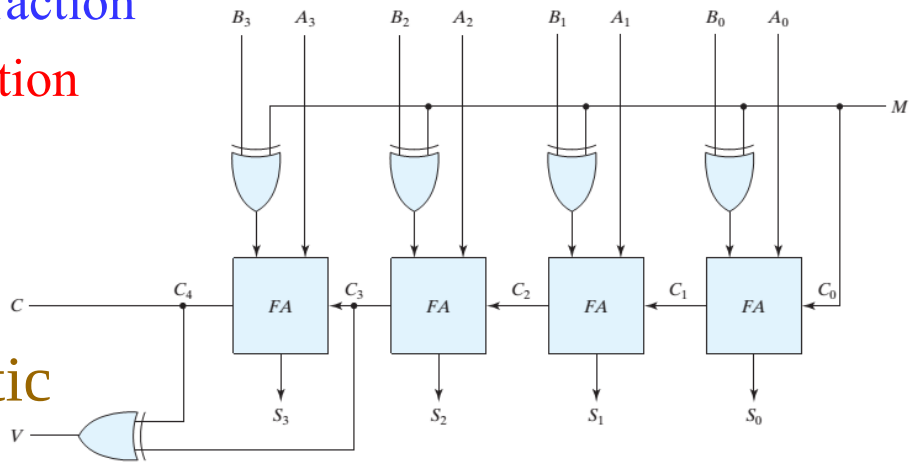


Figure 2

Chapter 3

- Half Adder
- Full Adder
  - *carry generate:* X ·Y
  - *carry propagate:* X⊕Y
- Binary Ripple Carry Adder
- Carry Lookahead Adder
- Binary subtraction
  - Unsigned 2's Complement subtraction
  - Signed 2's Complement subtraction
    - 1100 - 0011 = 1001
    - 0011 - 1100 = 0111
- Signed-Magnitude Arithmetic
- Signed-Complement Arithmetic



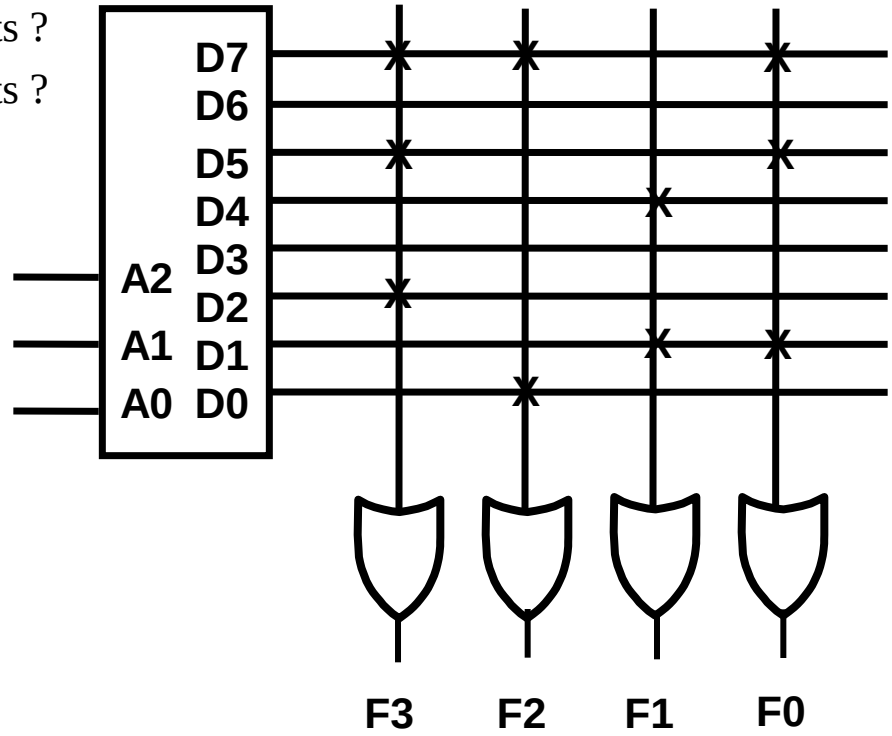


Chapter 5

- Programmable implementation technologies
  - PROM, PAL, PLA, FPGA

AND	OR	DEVICE
Fixed	Fixed	Not Programmable
Fixed	Programmable	PROM
Programmable	Fixed	PAL
Programmable	Programmable	PLA

- Programmable implementation technologies
  - PROM
  - Can any combinational circuit with *n* inputs and *m* outputs be implemented with
    - a PROM with *n* inputs and *m* outputs ?
    - a PLA with *n* inputs and *m* outputs ?
    - a PAL with *n* inputs and *m* outputs ?



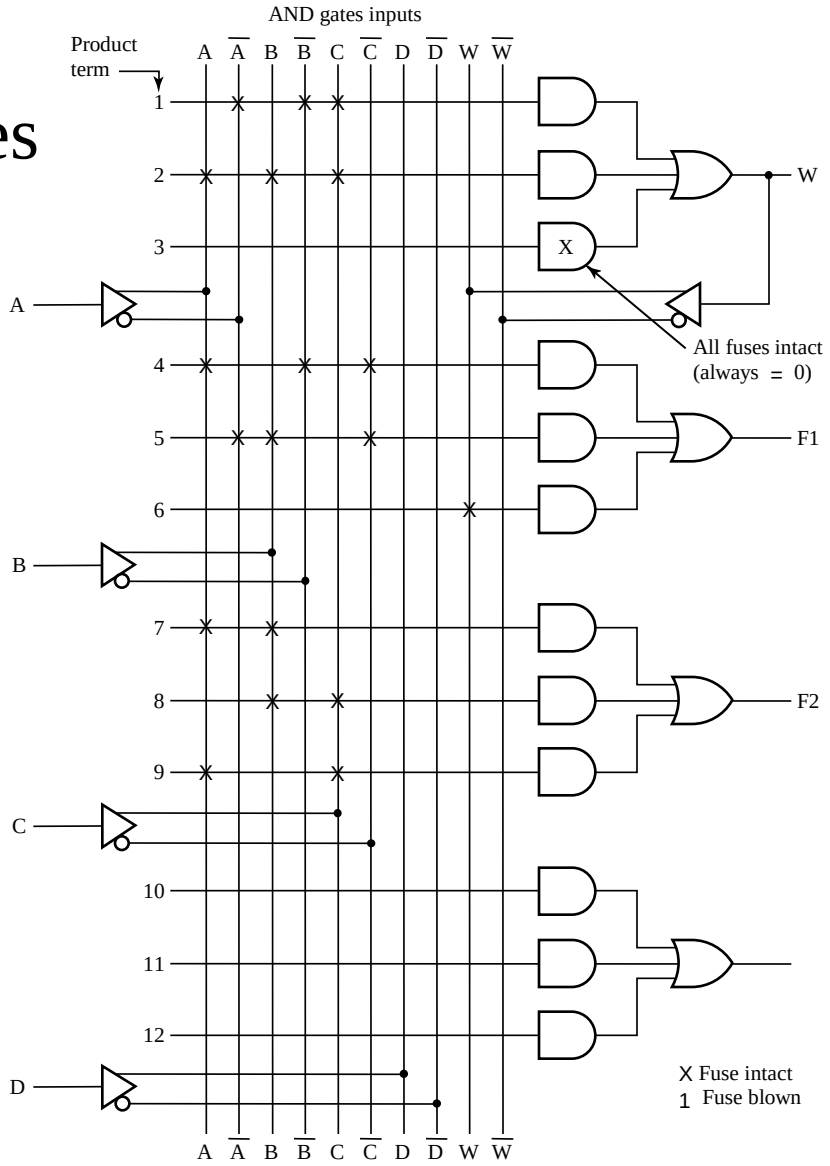
Chapter 5

- Programmable implementation technologies
  - PAL

$$W = \bar{A}\bar{B}C + A\bar{B}\bar{C}$$

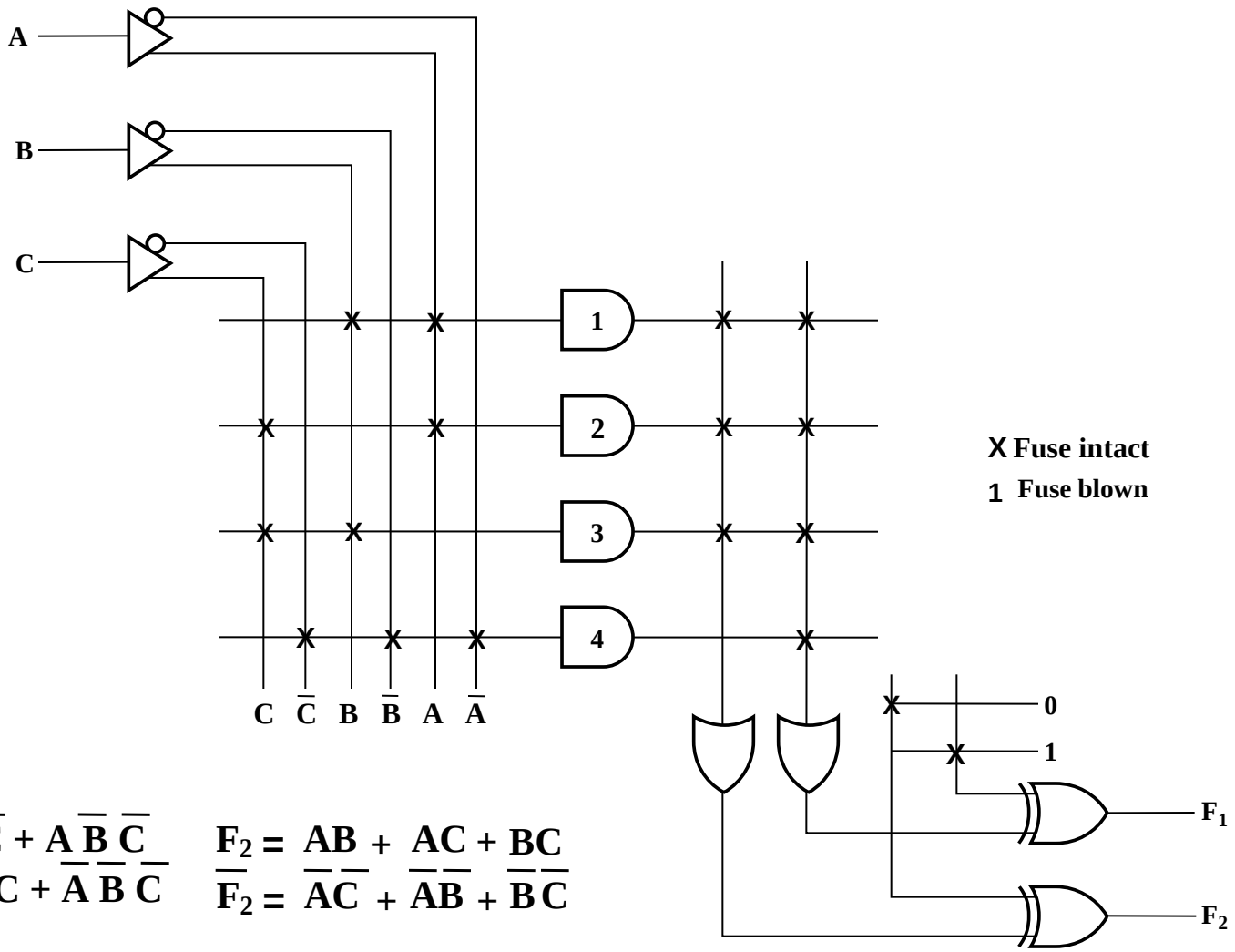
$$F1 = X = A\bar{B}\bar{C} + \bar{A}B\bar{C} + W$$

$$F2 = Y = AB\bar{C} + \bar{B}\bar{C} + A\bar{C}$$



Chapter 5

- Programmable implementation technologies
  - PLA



$$F_1 = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$

$$\bar{F}_1 = AB + AC + BC + \bar{A}\bar{B}\bar{C}$$

$$F_2 = AB + AC + BC$$

$$\bar{F}_2 = \bar{A}\bar{C} + \bar{A}\bar{B} + \bar{B}\bar{C}$$

Chapter 4

- Latches and flip-flops
- Master-slave flip-flop and edge-triggered flip-flop
  - SR master-slave flip-flop
  - edge-triggered D flip-flop
- Direct inputs: preset (direct set), clear (direct reset)
- Flip-flop timing
  - Setup time, hold time, propagation delay time
  - Timing Equations: $t_p = (t_{pd,FF} + t_{pd,COMB} + t_s) + t_{slack}$

    for t_{slack} greater than or equal to zero, $t_p \geq \max (t_{pd,FF} + t_{pd,COMB} + t_s)$ for all paths from flip-flop output to flip-flop input
  - How to calculate the maximum frequency of operation of circuit?

Chapter 4

- Sequential circuit analysis
  - Input equation
  - Next state equation
  - Output equations: Mealy model circuits, Moore model circuits
  - State table: present state, input, next state, output
  - State diagram: don’t-care condition
- Sequential circuit design
  - State diagram: state minimization, state assignment, don’t-care condition
  - State table
  - Excitation equation
  - Input equations
  - Output equations
  - Optimization
  - Technology Mapping
  - Verification

Chapter 4

- Difference between Latches and Flip-Flops

Latch	Flip-Flop
Latches are level sensitive devices	Flip-flops are edge sensitive devices
Latches are sensitive to glitches	Flip-flops are immune to glitches
Latches take less gates and power	Flip-flops take more gates and power
Latches are faster	Flip-flops are slower

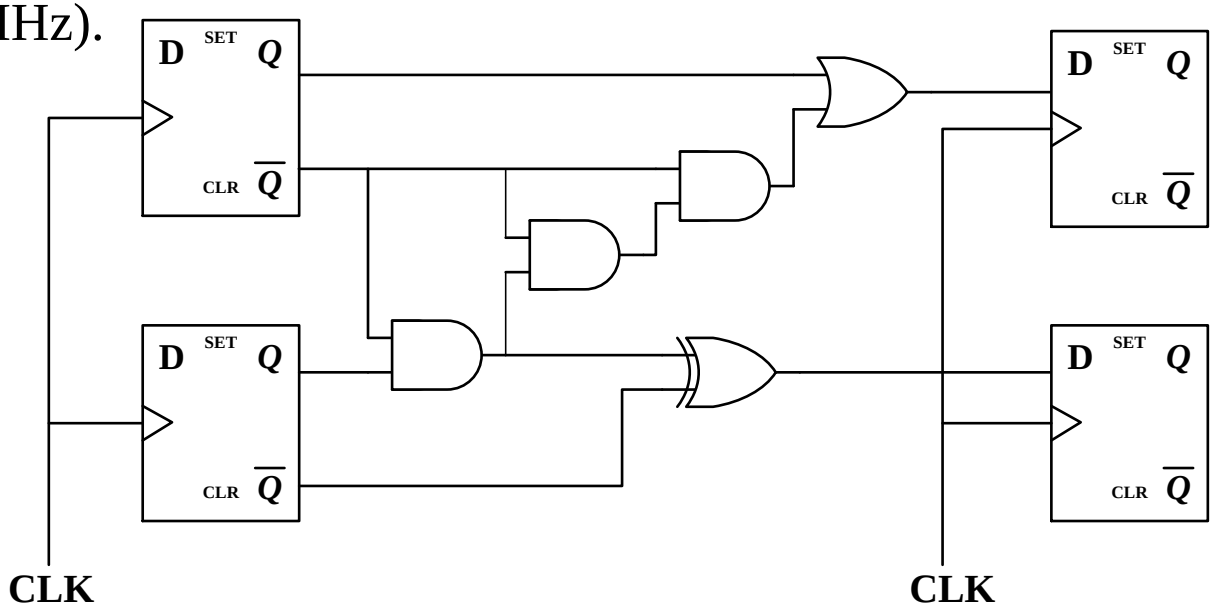
- Latches are transparent.
- Master-slave flip-flops use alternating clocks to break the path from input to output.
- S-R/J-K master-slave flip-flops have "1s catching" behavior.
- Edge-triggered flip-flops respond to the input at a well-defined moment (at the clock-transition).

Chapter 4

- Four basic descriptors for flip-flops
  - Characteristic (truth) tables
  - Characteristic equations
  - Excitation tables
  - Excitation equations
- Four different types of flip-flops
  - SR (S-Set, R-Reset) flip-flop
  - JK (J-Set, K-Reset) flip-flop
  - D (Data or Delay) flip-flop
  - T (Toggle) flip-flop
- Transformation among flip-flops
  - Mealy model circuits and Moore model circuits
  - Flip-Flop Conversion

- Problem: Sequential circuit analysis

The timing parameter in the sequential circuit are as follows: AND Gate: t_{pd} = 7.0ns; OR Gate: t_{pd} = 8.0ns; XOR Gate: t_{pd} = 11.0ns; Flip-flop: t_{pd} = 7.0ns, t_s = 4.0ns, t_h = 2.0ns. Determine the maximum frequency of operation of the circuit in megahertz (MHz).



1) find the longest path delay from clock edge to clock:

$t_{\text{delay-clock edge to clock edg}}$= FF t_{pd}+3*AND t_{pd}+OR t_{pd}+FF t_s=7+3*7+8+4=40 ns

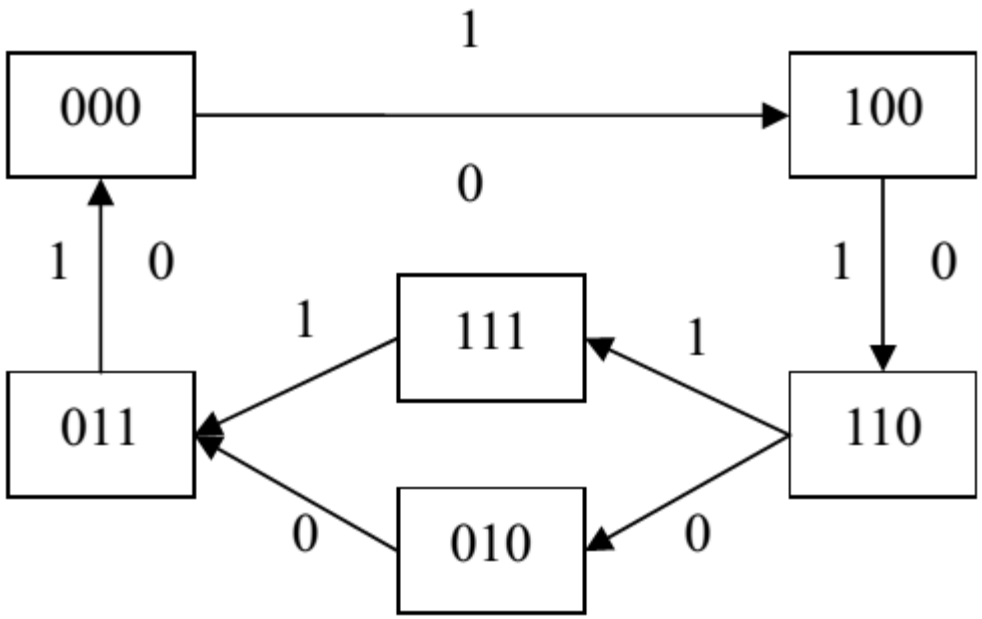
2) determine the maximum frequency: f_{max} = 1/t = 25 MHz

Chapter 4

- Problem: Sequence generation

Design a controllable sequence counter. When control input C = 1, counter sequence is 000→100→110→111→011→000; When C = 0, sequence is 000→100→110→010→011→000. Please derive the state diagram, state table, next state functions and output functions, and draw the circuit diagram.

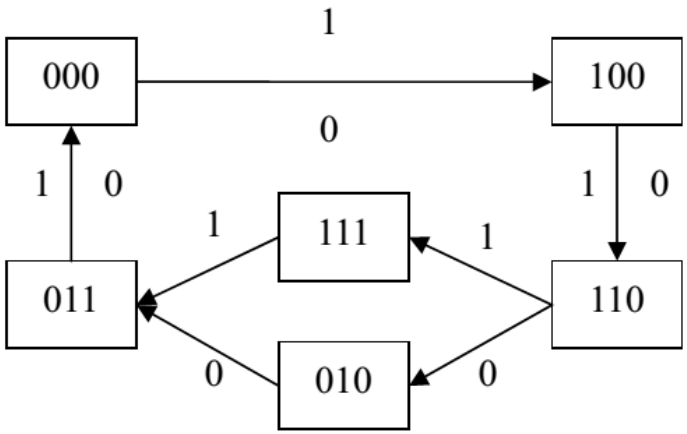
Answer:



State diagram

- Problem: Sequence generation

Answer:



State diagram

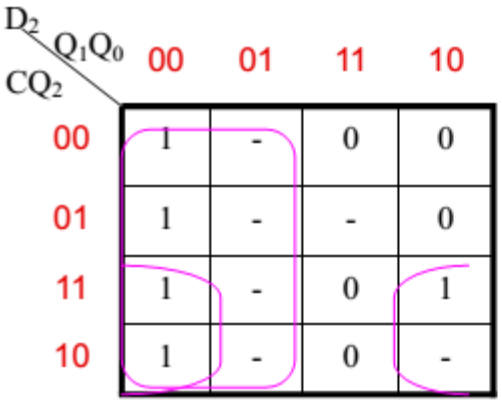
Current State Q2Q1Q0	Next State C = 0 Q2'Q1'Q0'	Next State C = 1 Q2'Q1'Q0'
000	100	100
001	xxx	xxx
010	011	xxx
011	000	000
100	110	110
101	xxx	xxx
110	010	111
111	xxx	011

State table

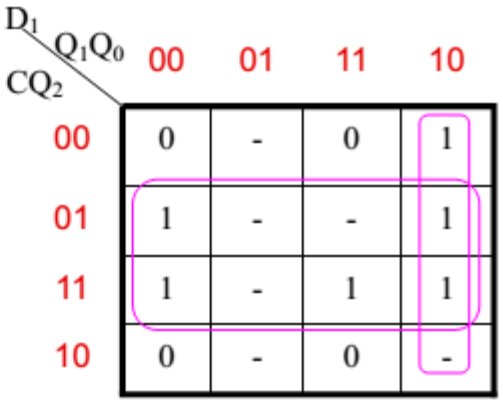
Chapter 4

- Problem: Sequence generation

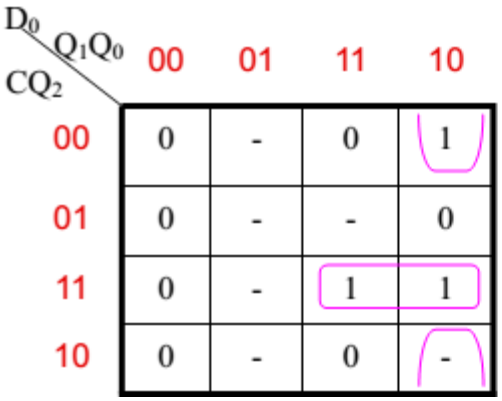
Answer:



D_2 Q_1Q_0 / CQ_2	00	01	11	10
00	1	-	0	0
01	1	-	-	0
11	1	-	0	1
10	1	-	0	-



D_1 Q_1Q_0 / CQ_2	00	01	11	10
00	0	-	0	1
01	1	-	-	1
11	1	-	1	1
10	0	-	0	-



D_0 Q_1Q_0 / CQ_2	00	01	11	10
00	0	-	0	1
01	0	-	-	0
11	0	-	1	1
10	0	-	0	-

Input equations

$$D_2 = \overline{Q_1} + C\overline{Q_0}$$

$$D_1 = Q_2 + Q_1\overline{Q_0}$$

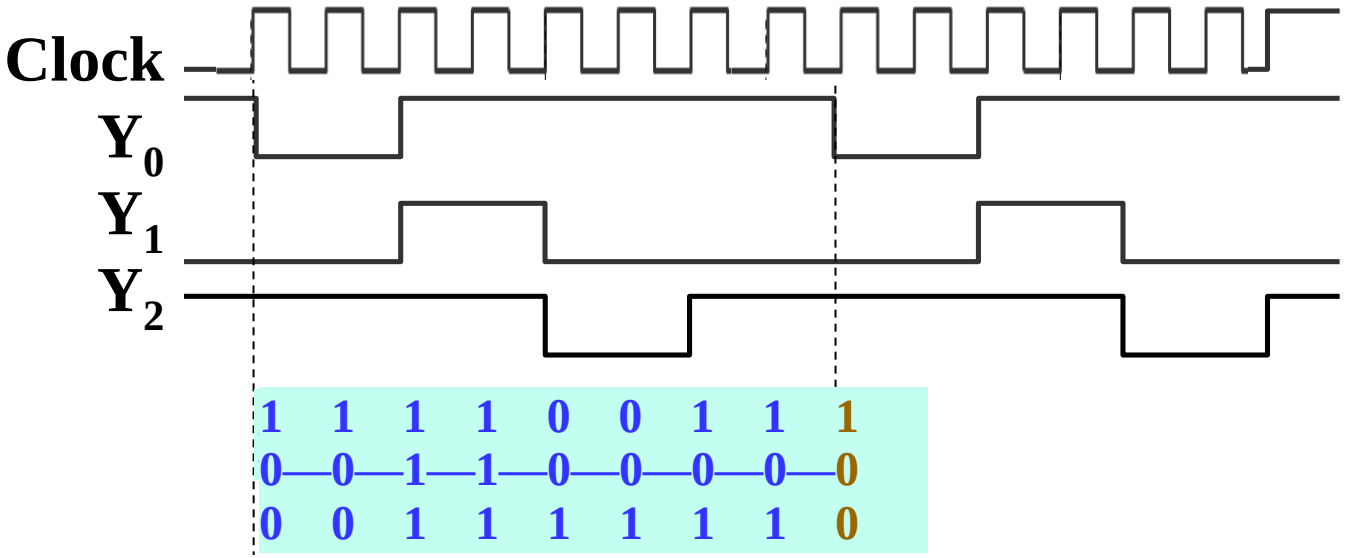
$$D_0 = CQ_2Q_1 + \overline{Q_2}Q_1\overline{Q_0}$$

Chapter 4

- Problem: Sequence generation

Design a waveform generator using D flip-flops and NOR gates. The waveforms of output Y0~Y2 are shown below. Requirement:

- Draw the Moore state diagram for the circuit.
- Find the state table and make a state assignment.
- Derive the next state functions and output functions.
- Design the circuit using D flip-flops and NOR gates, draw the circuit diagram.



Chapter 6

- Register Transfer Operations (RTL)
  - Microoperations
    - Let R1 = 10101010 and R2 = 11110000, then after the operation, R0 becomes:
  - Conditional Transfer
- Register Transfer Structures
  - Multiplexer-Based Transfers
  - Bus-Based Transfers
  - Three-State Bus
  - Other Transfer Structures
- Shift Registers
  - Parallel Load Shift Registers
  - Bidirectional Shift Register

R0	Operation
10101010	R0 ← R1
11111010	R0 ← R1 ∨ R2
10100000	R0 ← R1 ∧ R2
01011010	R0 ← R1 ⊕ R2
01010100	R0 ← sl R1

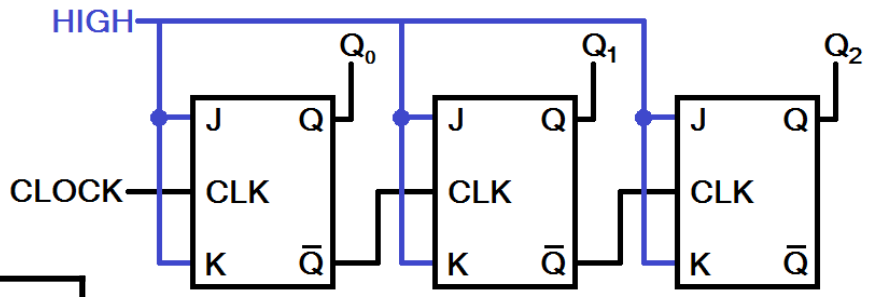
Mode Control		Register Operation
S_1	S_0	
0	0	No change
0	1	Shift down
1	0	Shift up
1	1	Parallel load

Chapter 6

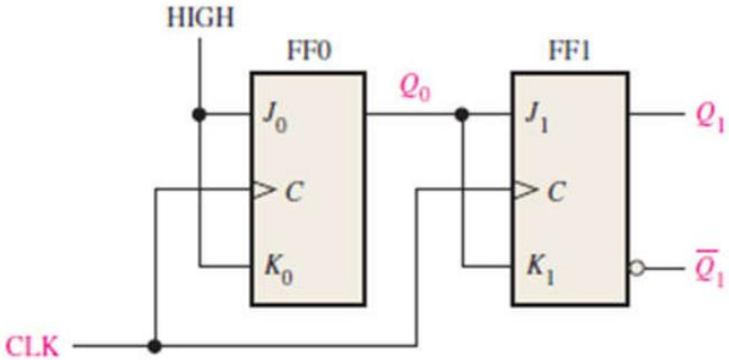
- Counters
  - Asynchronous/Ripple Counter
  - Synchronous Counter
    - Counter with Parallel Load

Load	Count	Action
0	0	Hold Stored Value
0	1	Count Up Stored Value
1	X	Load D

  - Divide-by-n (Modulo n) Counter
    - A synchronous 4-bit binary counter with a synchronous load and an asynchronous clear is used to make a Modulo 7 counter
    - How to design a Modulo-17 counter?



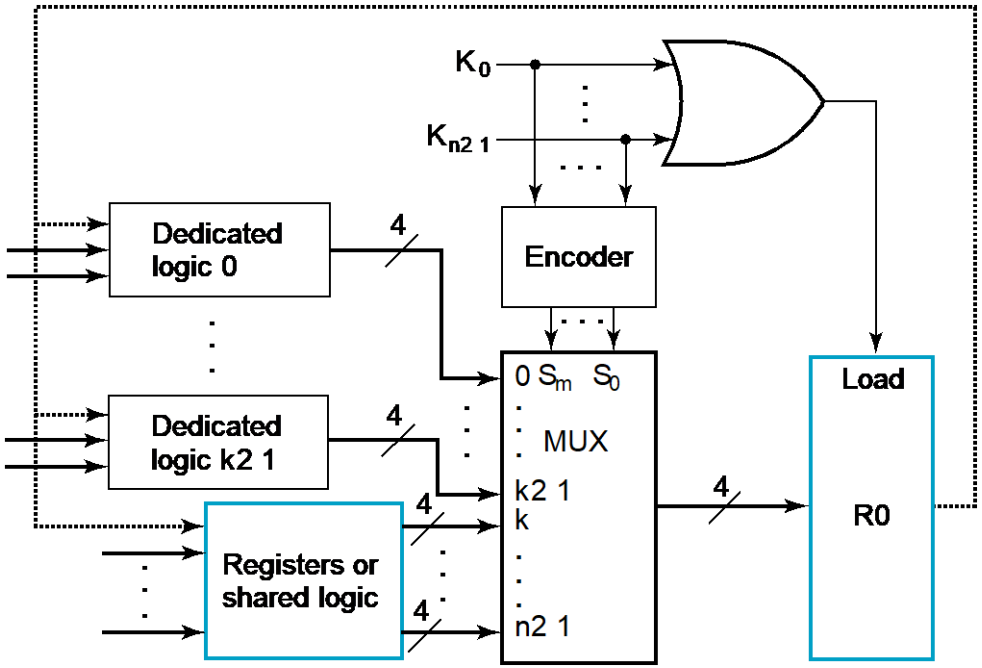
Asynchronous/Ripple Counter

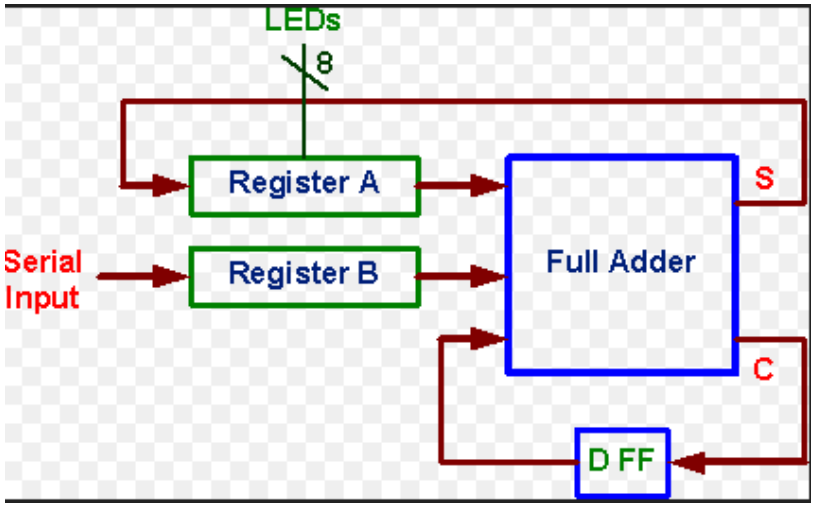


Synchronous Counter

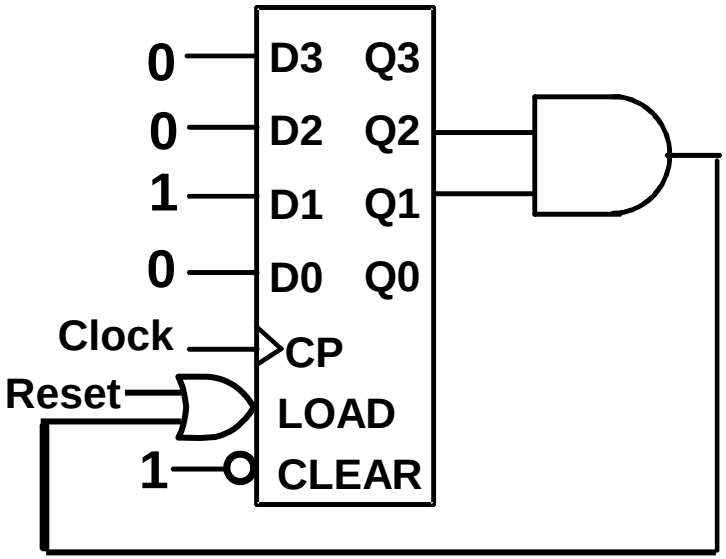
Chapter 6

- Register Cell Design
  - Sequential Circuit Design Approach
  - Multiplexer Approach
- Serial microoperations



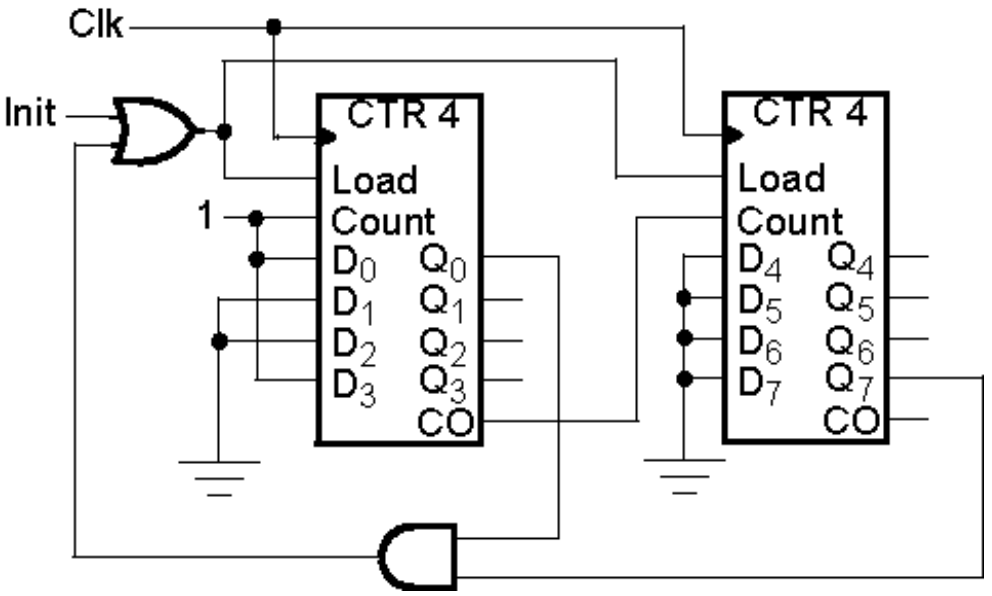


Problem: Circuit Analysis



a 4-bit binary counter

0010-0110



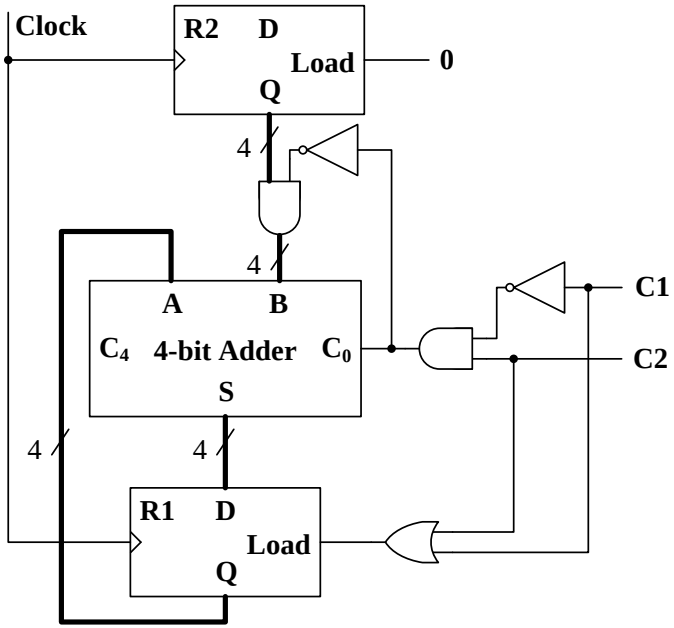
an 8-bit binary counter

00001001-10000001

- Problem: Circuit Analysis

  Analyze the following register transfer circuit, finish the function table, and write down the corresponding register transfer operation statements in RTL forms.

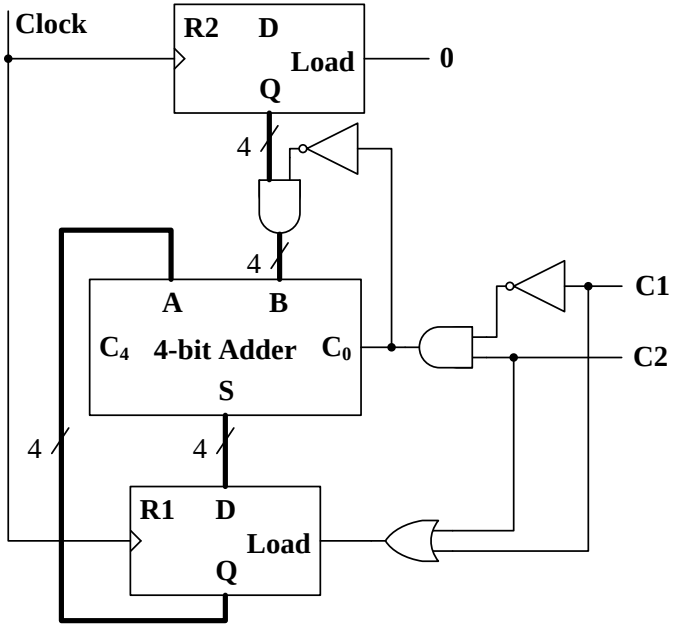
C1	C2	Input Load of R1	C_0	Next state of R1	Function
0	0	0	0	R1	No change
0	1				
1	0				
1	1				



- Problem: Circuit Analysis

  Analyze the following register transfer circuit, finish the function table, and write down the corresponding register transfer operation statements in RTL forms.

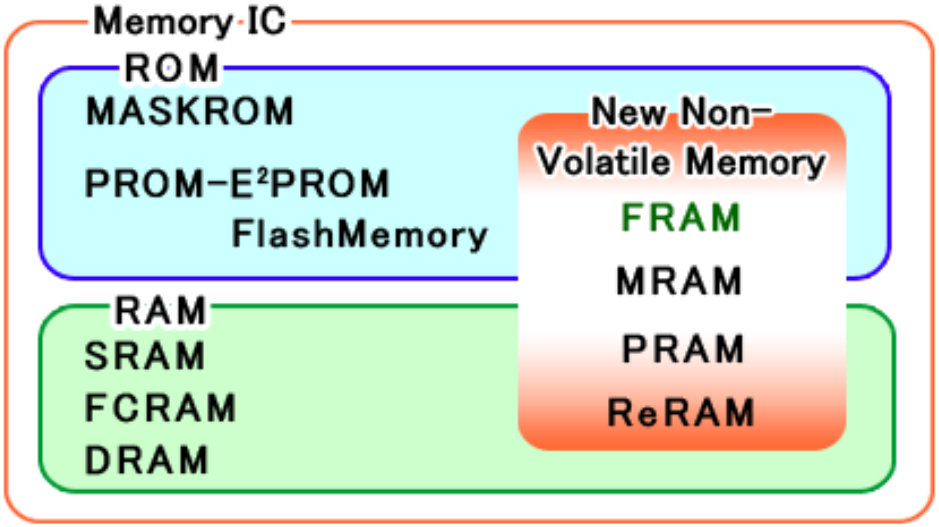
C1	C2	Input Load of R1	C_0	Next state of R1	Function
0	0	0	0	R1	No change
0	1	1	1	R1 + 1	Increment
1	0	1	0	R1 + R2	Addition
1	1	1	0	R1 + R2	Addition



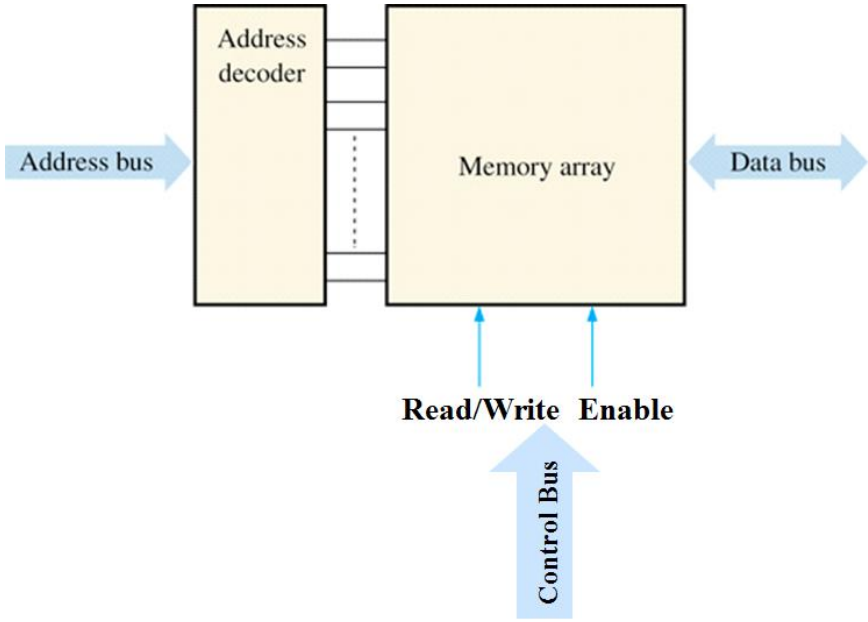
C1: R1←R1 + R2

$\overline{C1}$ C2: R1←R1 + 1

Chapter 7



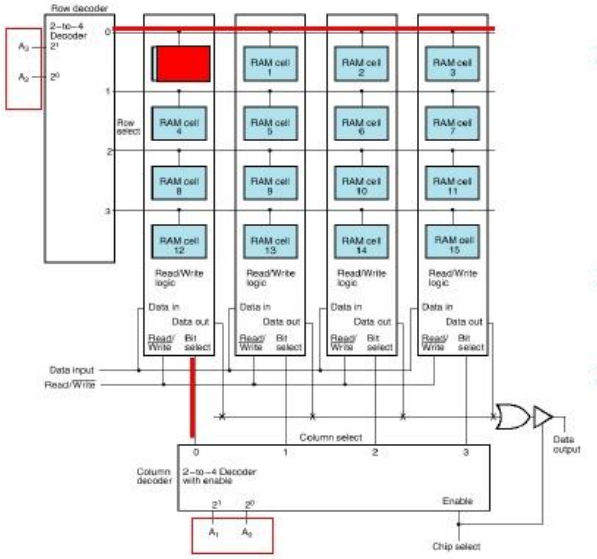
Computer memory

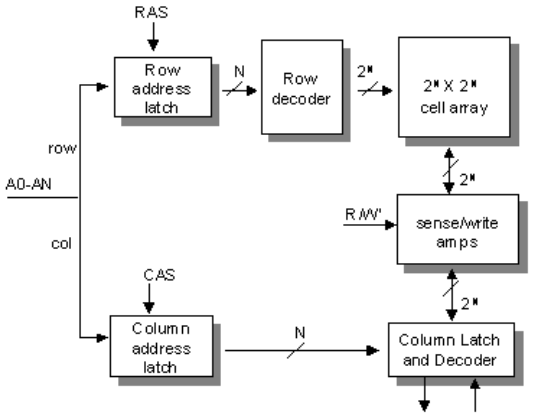


Block Diagram of RAM

Chapter 7

- Memory Operation Timing
  - Read timing
  - Write timing
- Coincident Decoding
  - Row select
  - Column select
- Memory Expansion
  - Word-Capacity Expansion: Word extension (address bits are increased)
  - Word-Length Expansion: Bit extension
- DRAM
  - Address multiplexing
  - Refresh policy
  - Burst read
  - How to calculate memory bandwidth?





Chapter 7

- Problem:

The memory units that follow are specified by the number of words times the number of bits per word. How many address lines and data lines are needed in each case?

(a) 8K * 16 (b) 2G * 8

(c) 16M * 32 (d) 256K * 64

Answer:

(a) 8 K * 16 = 2^{13} * 16 A = 13, D = 16

(b) 2 G * 8 = 2^{31} * 8 A = 31, D = 8

(c) 16 M * 32 = 2^{24} * 32 A = 24, D = 32

(d) 256 K * 64 = 2^{18} * 64 A = 18, D = 64

Chapter 7

- Problem:

A 16K * 4 memory uses coincident decoding by splitting the internal decoder into X-selection and Y-selection.

(a) What is the size of each decoder, and how many AND gates are required for decoding the address?

(b) Determine the X and Y selection lines that are enabled when the input address is the binary equivalent of 6,000.

Answer:

(a) 16 K = $2^{14} = 2^7 * 2^7 = 128 * 128$

Each decoder is a 7-to-128-line decoder

Decoders require 256 AND gates, each with 7 inputs

(b) $(6{,}000)_{10} = (0101110\ 1110000)_2$

X = 46, Y = 112

Chapter 7

- Problem:

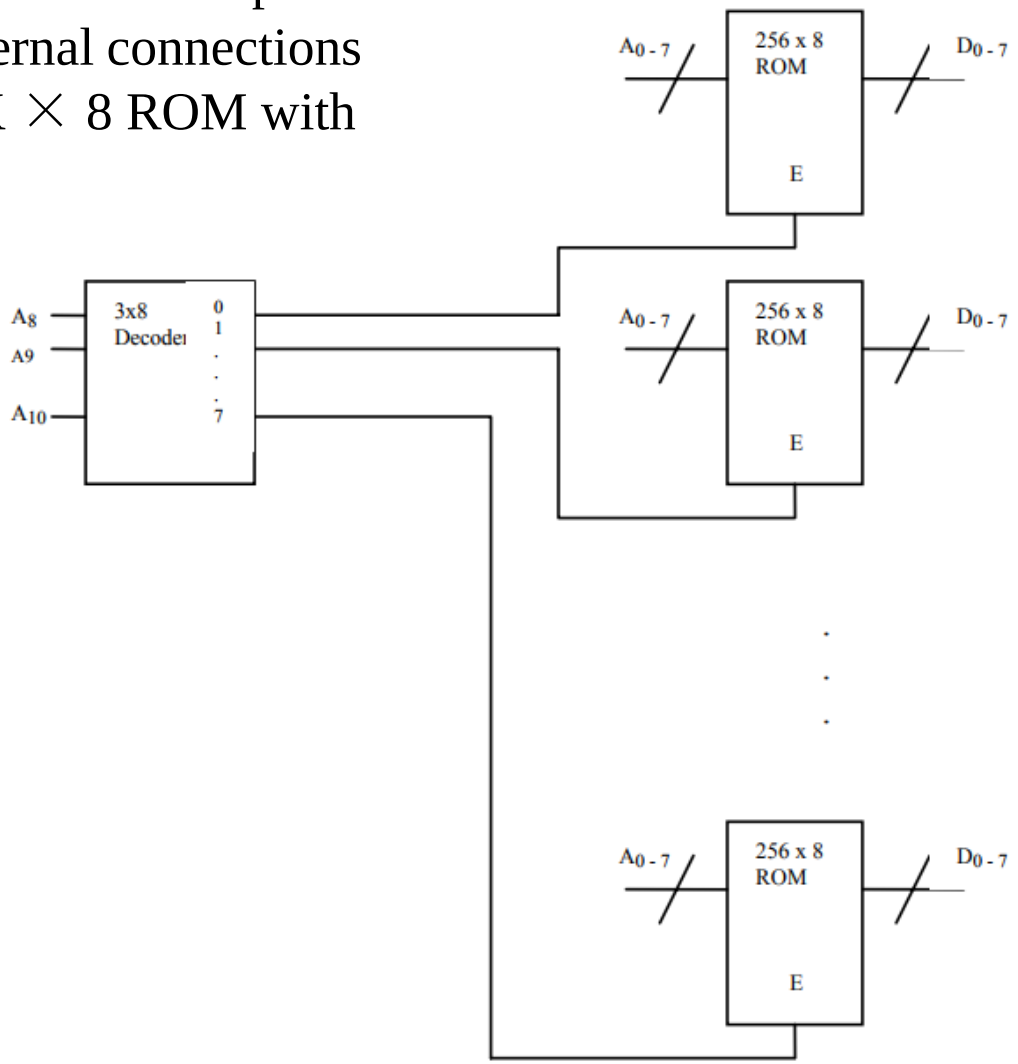
  A DRAM chip uses two-dimensional address multiplexing. It has 13 common address pins, with the row address having one bit more than the column address. What is the capacity of the memory?

  Answer:

  13 + 12 = 25 address lines.

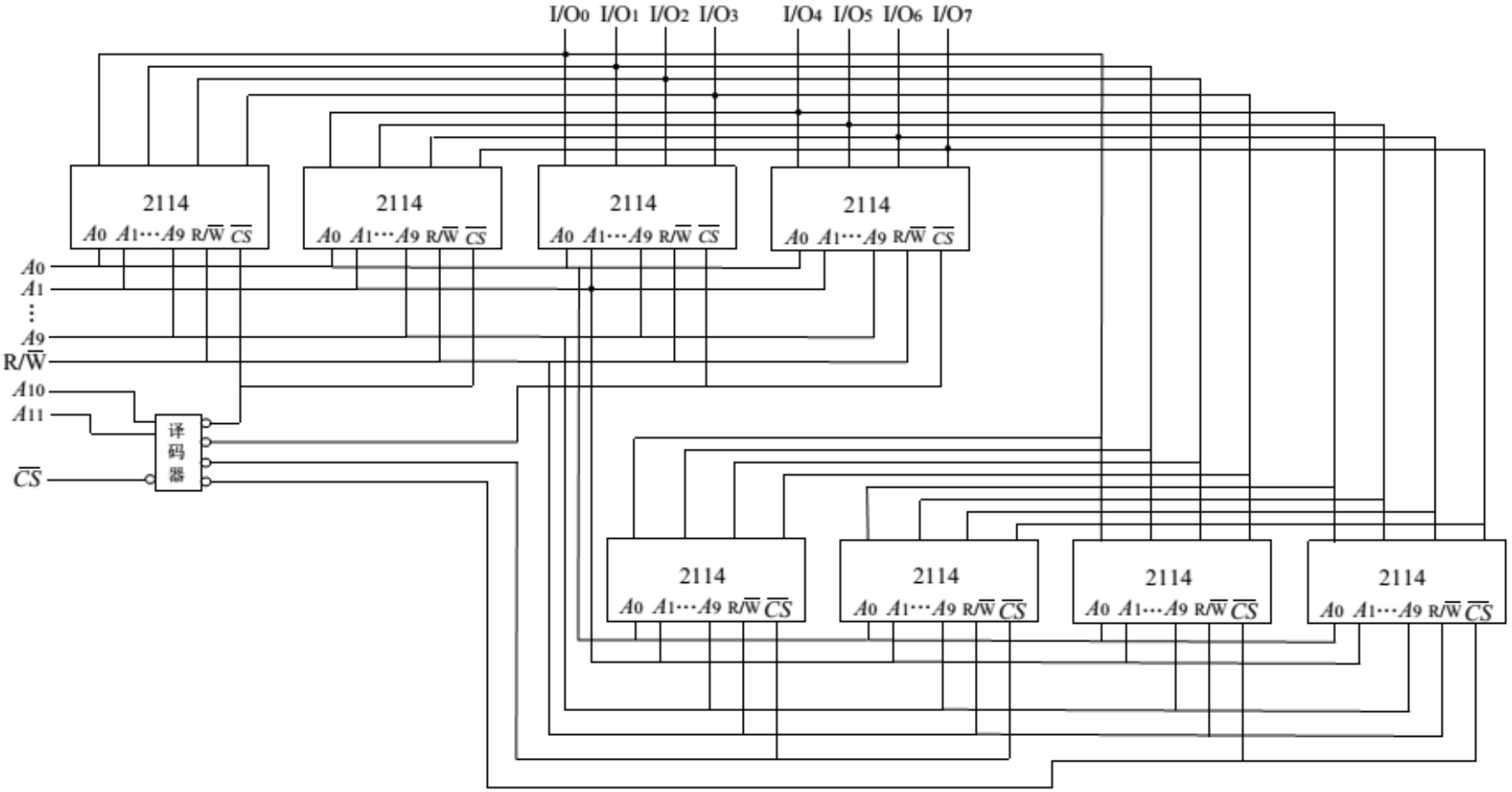
  Memory capacity = 2^{25} words.

- Problem: Given a 256 × 8 ROM chip with Enable input, show the external connections necessary to construct a 2K × 8 ROM with eight chips and a decoder.

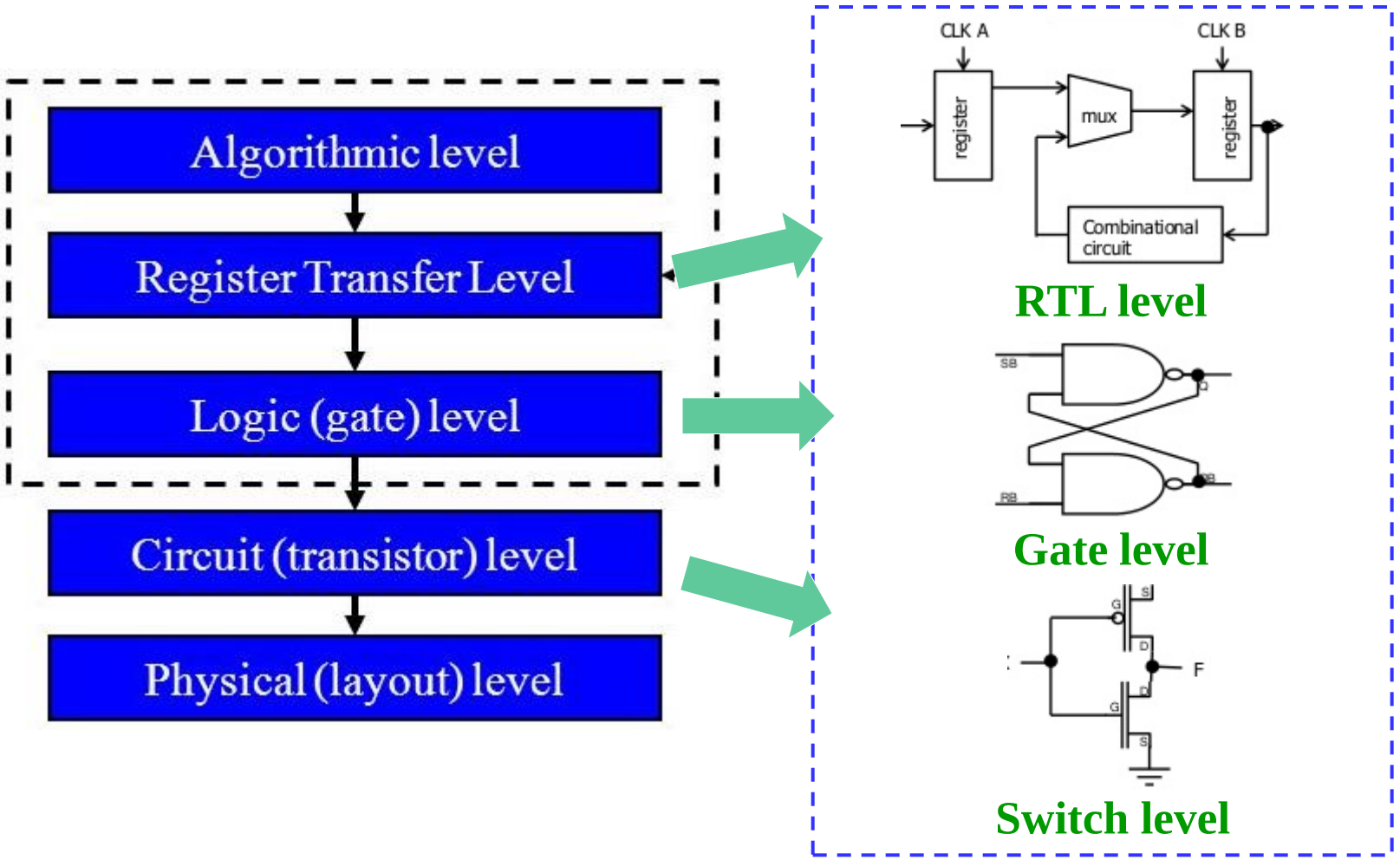


Chapter 7

- Problem: Given a 1K × 4 ROM chip with Enable input, show the external connections necessary to construct a 4K × 8 ROM with eight chips and a decoder.

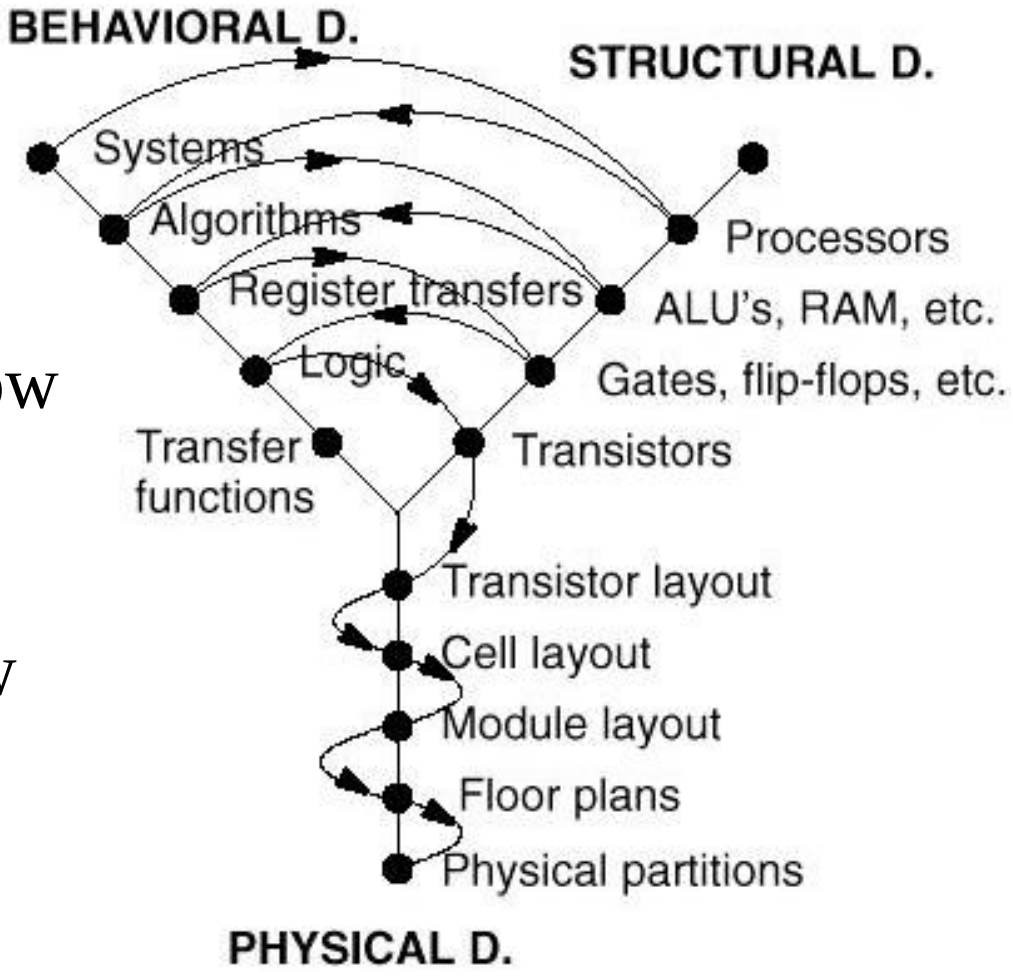


Levels of Logic Design



Levels of Logic Design (continued)

- Behavioral: specifies what a specific system does
- Structural: specifies how entities are connected together
- Physical: specifies how to build a structure



From Structural Design Perspective

- Gates: AND, OR, NOT, NAND, NOR, 3-state buffer (Hi-Z), XOR, XNOR
- Gate-Level Technology Mapping
- Rudimentary functions: decoder, encoder, multiplexer, demultiplexer
- Arithmetic functions: half/full adder, ripple carry/carry lookahead adder, adder/subtractor, multiplication, shifter, ALU
- Programmable devices: PROM, PAL, PLA, FPGA

From Structural Design Perspective (continued)

- Latches: SR, JK, D, T
- Flip-flops: Master-Slave/Edge-triggered, SR/JK/D/T
- Registers: counter, shifter
- Register Transfer Structures: MUX-based, MUX bus, 3-state bus
- Memories: ROM, SRAM, DRAM, SDRAM

From Behavioral Design Perspective

- Methods of designing combinational circuit
  - Designing theory: time-independent logic
  - Describing methods: truth table (canonical form), Karnaugh maps, timing diagram, boolean function, logic circuit
  - Optimization: two-level circuit optimization (Boolean algebra), iterative array, contraction
  - Timing and performance: gate input cost, fan-in/fan-out, delay model

From Behavioral Design Perspective (continued)

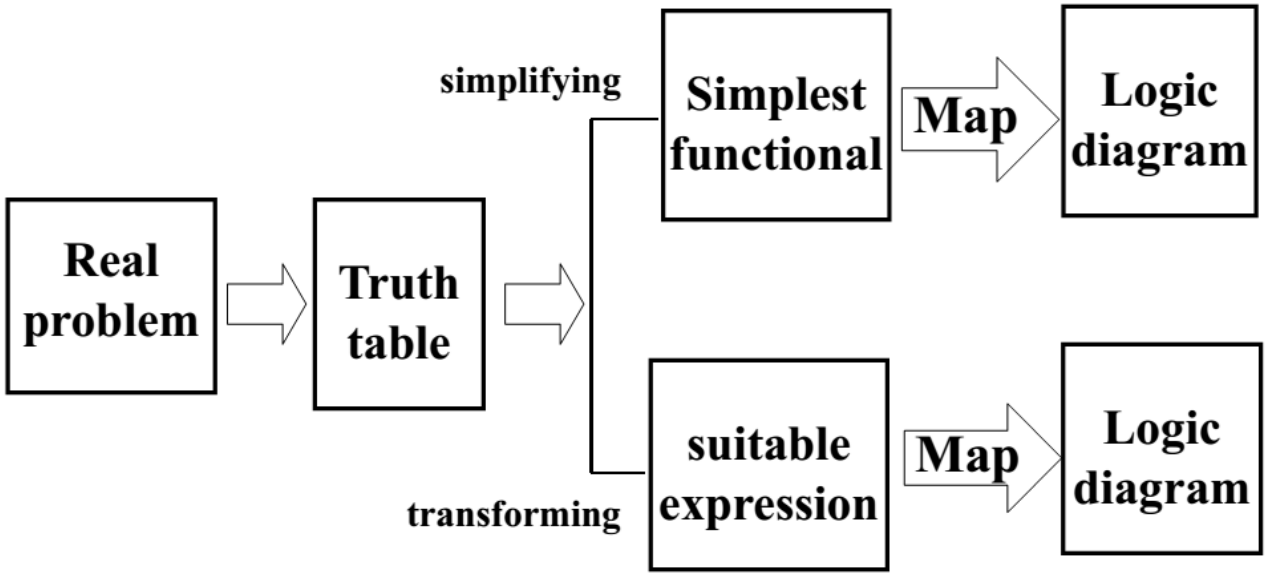
- Methods of designing sequential circuit
  - Designing theory: finite state machine (Mealy model and Moore model)
  - Describing methods: state table, state diagram, next state equation(characteristic equation), input equation, excitation equation, output equation
  - Optimization: state minimization, state assignment
  - Encoding: Counting order, Gray and One-hot code
  - Timing and performance: glitch, 1's catching, t_s, t_h, t_w, t_{px}

From Behavioral Design Perspective (continued)

- Methods of designing digital system
  - Designing theory: state-machine diagram, register design model, datapath+control unit
  - Describing method: Register Transfer Language(RTL)
  - Timing and performance: system timing equation $t_p = t_{pd,FF} + t_{pd,COMB} + t_s + t_{slack}$, maximum frequency
- Methods of designing memory
  - Describing methods: address, data, operation
  - Optimization: coincident decoding, address multiplexing, burst read
  - Timing and performance: read timing, write timing, memory bandwidth

Method of Logic Design

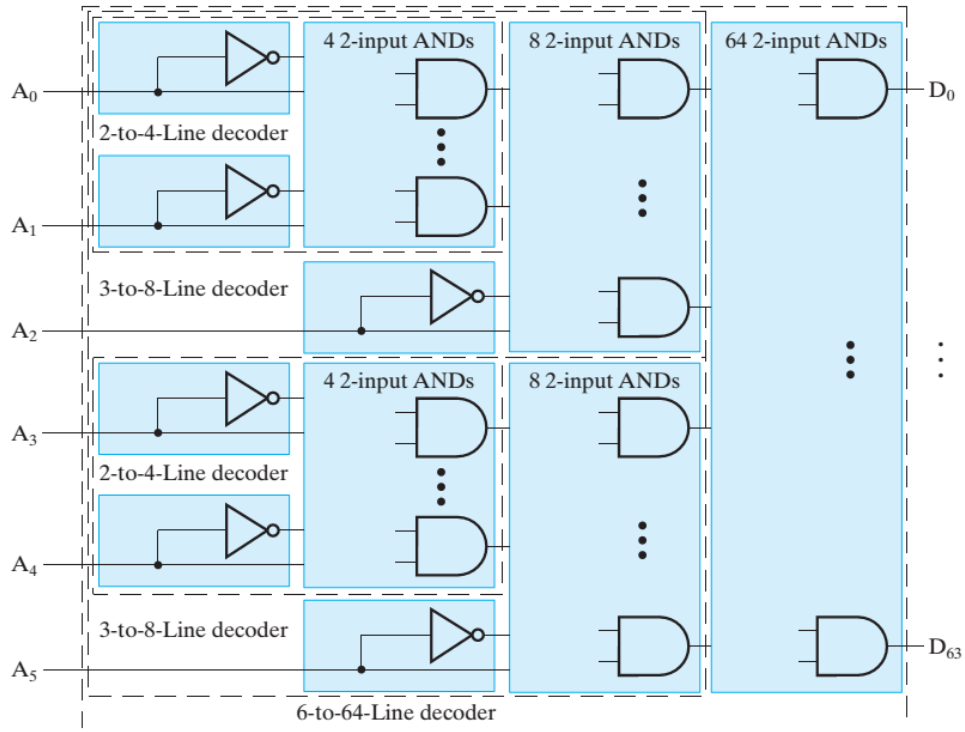
- Methods of designing digital circuits
  - Design by truth table/state table
  - Design by bisection
  - Design in a hierarchical structure
  - Design by iteration
  - Design by contraction/expansion



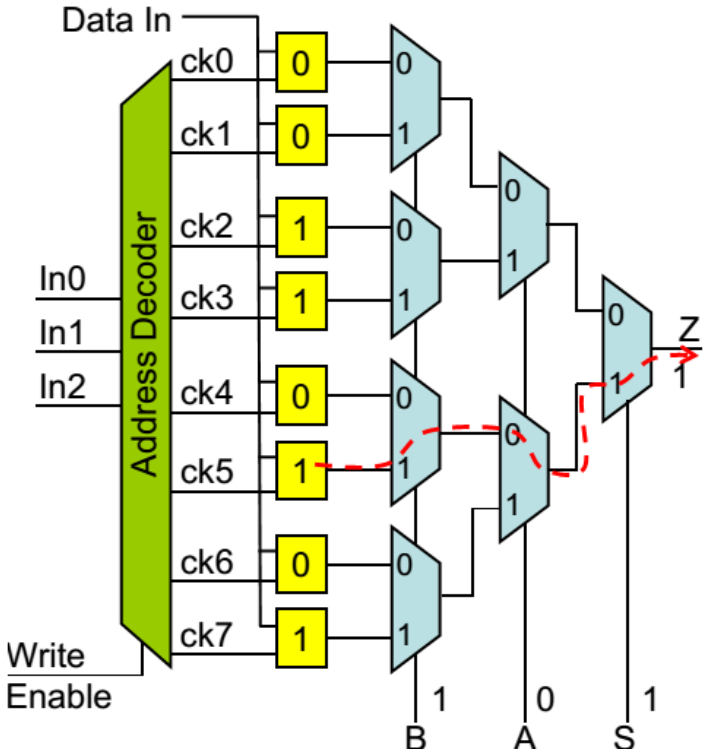
Design by truth table

Design of Combinational Circuits

- Design by bisection



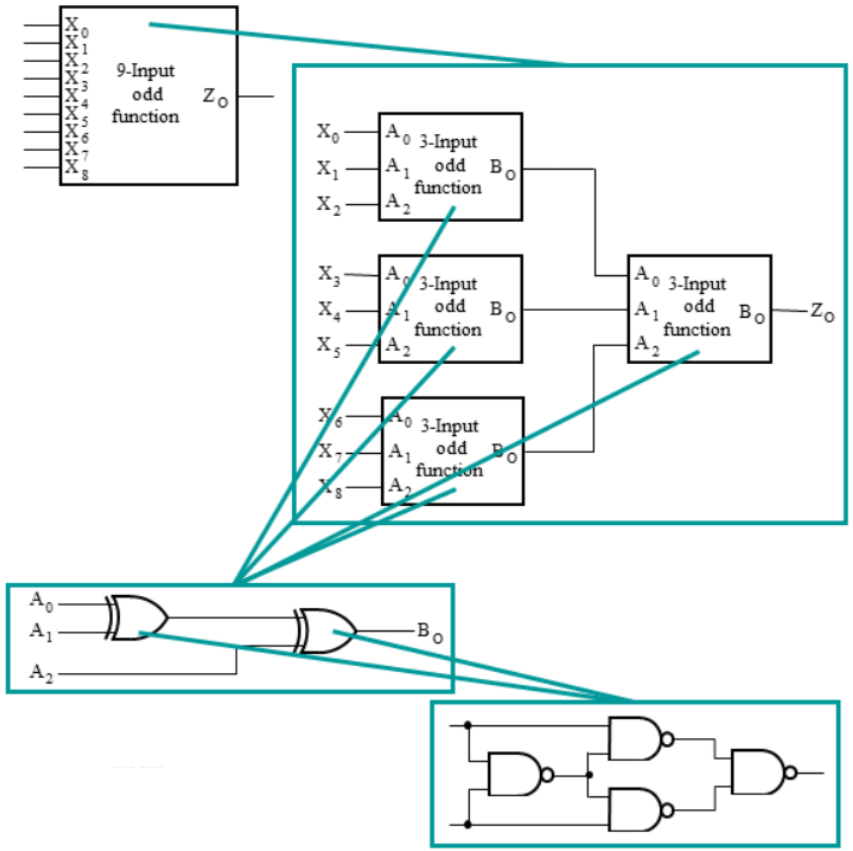
Decoder expansion



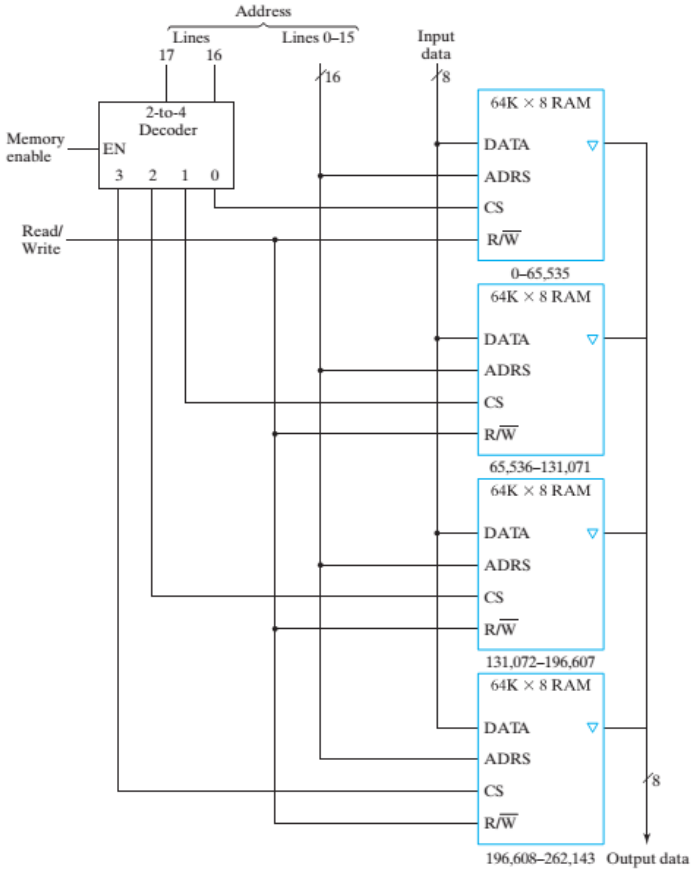
Lookup tables

Design of Combinational Circuits

- Design in a hierarchical structure



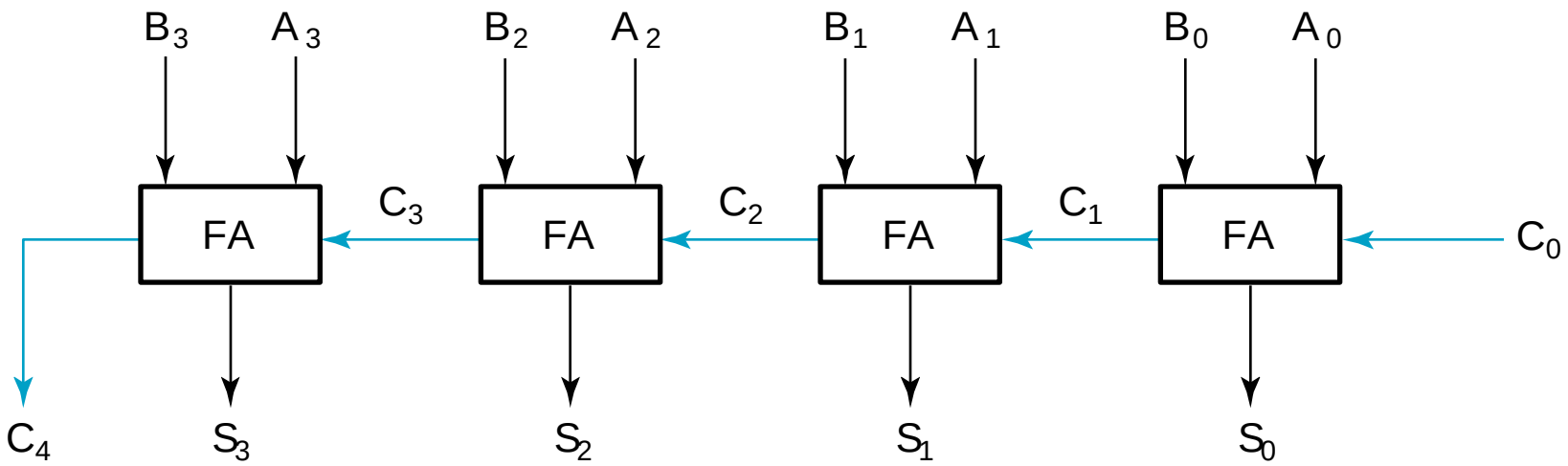
9-input parity tree



256K * 8 RAM with four 64K * 8 RAM

Design of Combinational Circuits

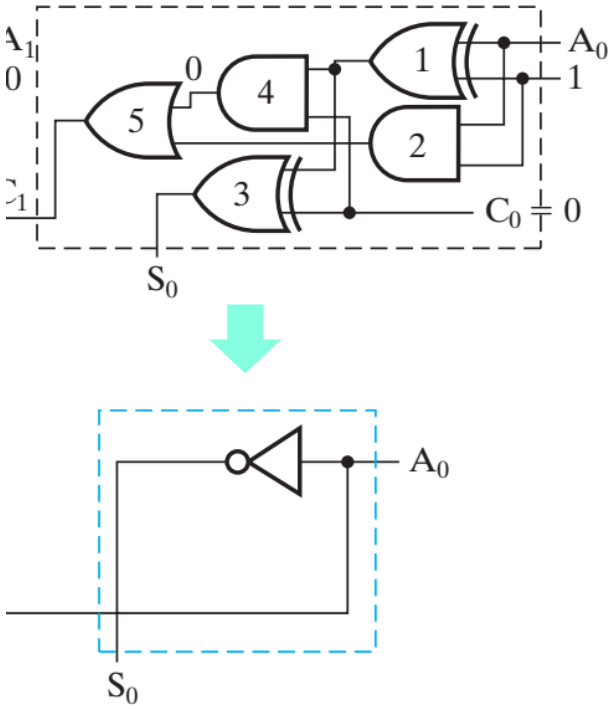
- Design by iteration



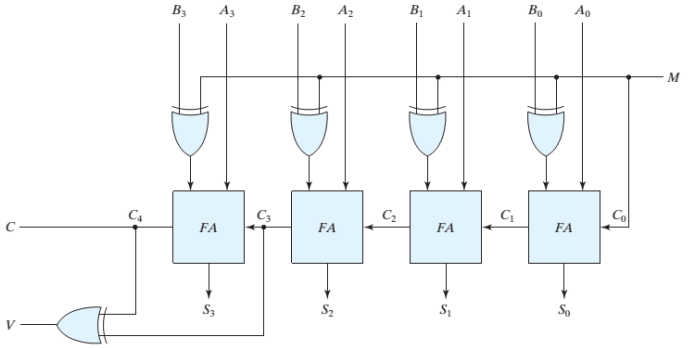
A Four-bit Ripple Carry Adder

Design of Combinational Circuits

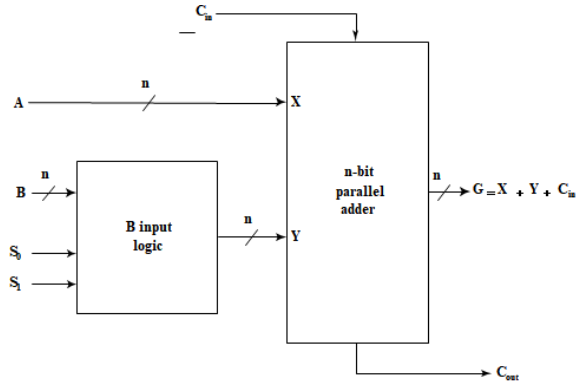
- Design by contraction/expansion



Contraction of Adder to Incrementer



Expansion of inputs for constructing substractor



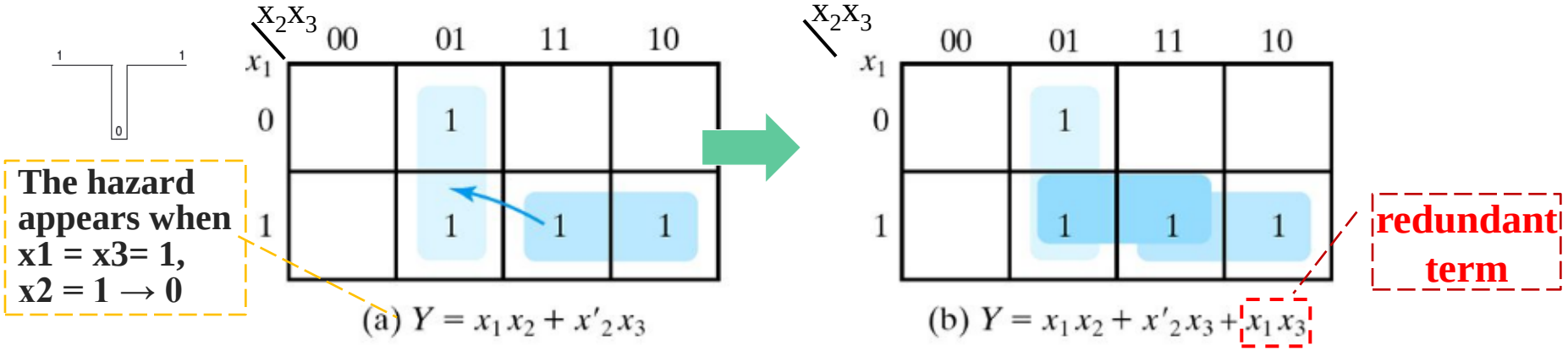
Expansion of inputs for constructing ALU

Design Tradeoffs in Logic Circuits

- Performance-Cost tradeoff
  - Two-level circuit vs. Multiple-level circuit
  - Carry lookahead adder vs. Ripple carry
  - Parallel adder vs. Serial adder
  - Synchronous counter vs. Asynchronous counter
  - SRAM vs. DRAM

Design Tradeoffs in Logic Circuits (continued)

- Performance-Reliability tradeoff
  - Latch vs. Flip-flop
  - Mealy model vs. Moore model
  - Not encoded (One-hot Code) vs. Encoded (Gray Code)
- Cost-Reliability tradeoff
  - No self-correcting vs. Self-correcting
  - Optimal implementation vs. redundant implementation



End

- Final Examination Time:

Jan. 11, 2024 10:30~12:30

- Good luck!

Chapter 2

- Problems:

1. The dual of an algebraic expression is obtained by 1) interchanging OR and AND operations and 2) replacing 1's by 0's and 0's by 1's.
2. Use DeMorgan's Theorem to complement a function: 1) interchange AND and OR operators and 2) complement each constant value and literal
3. Four variables odd function has ____C____ "1" squares in its corresponding K-Map.

   A. 4 B. 7 C. 8 D. 14
4. The gate input cost G of function $F=AB(C+D)+C(\bar{B}D+\bar{A}D)$ is __A__.

   A. 15 B. 14 C. 13 D. 12
5. Which of the following logical gates can be used as a controllable inverter? ____B____.

   A. AND gate B. XOR gate C. Buffer gate D. OR gate
6. The Essential Prime Implicants in the K-Map given below are ___B___.

   A. Y'Z', XZ' B. X'Y', XY C. XY, XZ' D. Y'Z', X'Y'
7. Given below are the waveforms of input A, B and output F of a logic device. Then the device is a ___C___ gate.

   A. NAND B. NOR C. XOR D. OR