Logic and Computer Design Fundamentals

Term Review

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

Previous Year's 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

Course Review

HIGHLIGHTS & PROBLEMS

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 (next state 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).

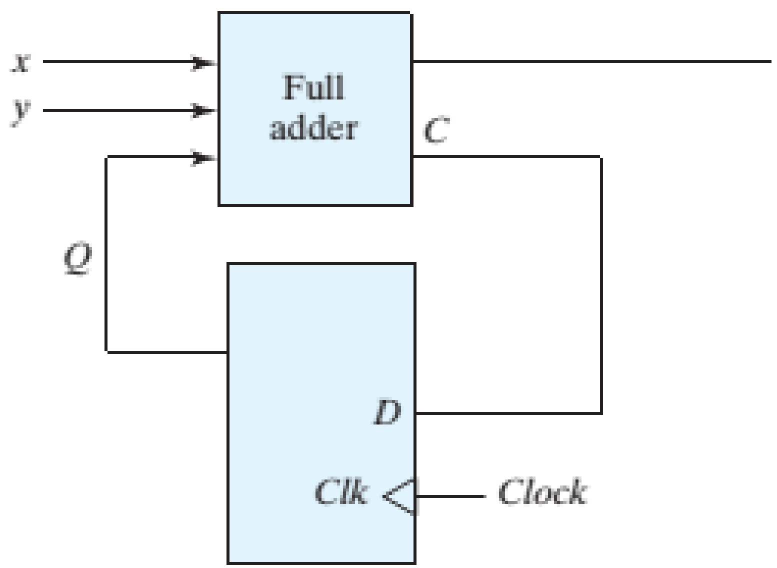
Chapter 4

- Three basic descriptors for flip-flops
  - Characteristic (truth) tables
  - Characteristic equations
  - Excitation tables
- 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

Chapter 4

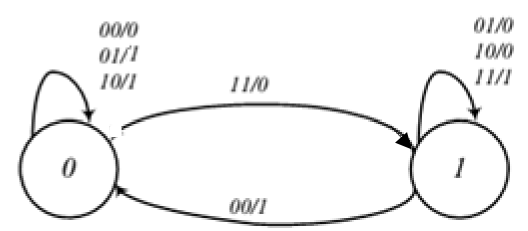
- Problem: Sequential circuit analysis

A sequential circuit has one flip-flop Q, two inputs x and y, and one output S. It consists of a full-adder circuit connected to a D flip-flop, as shown in Fig. Derive the state table and state diagram of the sequential circuit.



Present state Q	Inputs x	Inputs y	Next state Q	Output S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

State table

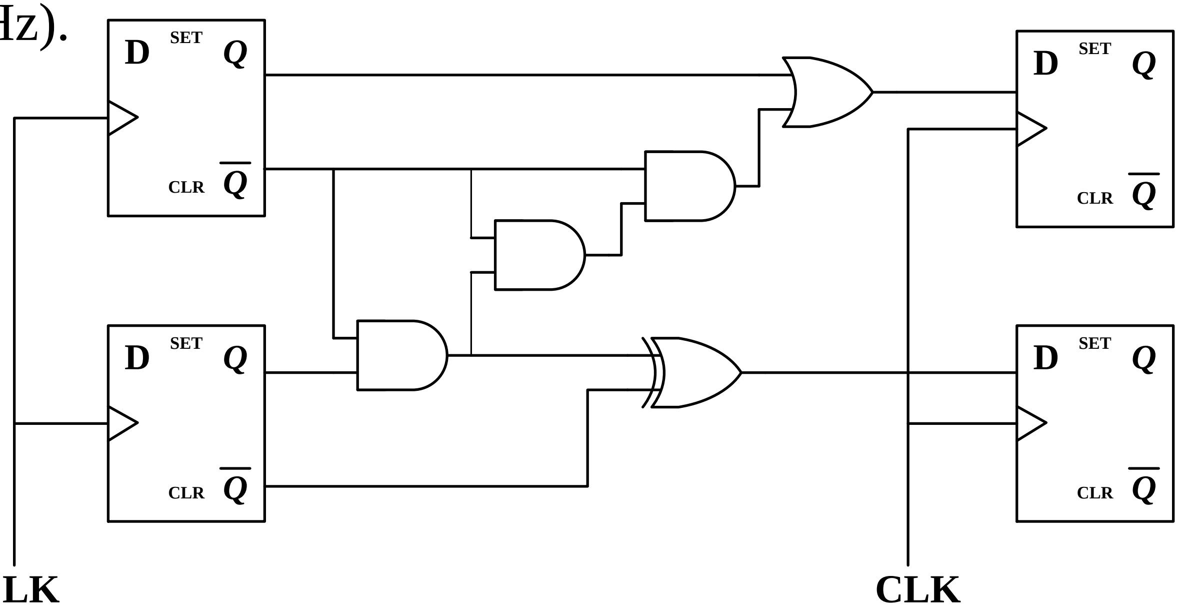


State diagram

Chapter 4

- 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:

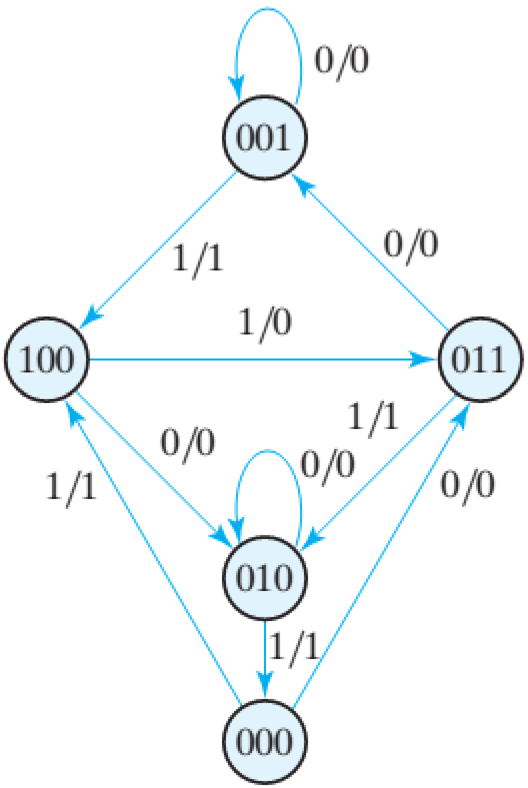
$t_{\text{delay-clock edge to clock edg}}$= FF t_{pd}+3*AND t_{pd}+OR t_{pd}+FF t_s=7+3*7+8+4=40 ns

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

Chapter 4

- Problem: Sequential circuit design

Design the sequential circuit specified by the state diagram in the figure using JK flip-flops.



Answer:

State table

Present state ABC	Input x	Next state ABC	Output y
000	0	011	0
000	1	100	1
001	0	001	0
001	1	100	1
010	0	010	0
010	1	000	1
011	0	001	0
011	1	010	1
100	0	010	0
100	1	011	1

unused states: 101, 110, 111

- Problem: Sequential circuit design

Design the sequential circuit specified by the state diagram in the figure using JK flip-flops.

Answer:

Present state ABC	Input x	Next state ABC	Output y	J_A	K_A	J_B	K_B	J_C	K_C
000	0	011	0	0	x	1	x	1	x
000	1	100	1	1	x	0	x	0	x
001	0	001	0	0	x	0	x	x	0
001	1	100	1	1	x	0	x	x	1
010	0	010	0	0	x	x	0	0	x
010	1	000	1	0	x	x	1	0	x
011	0	001	0	0	x	x	1	x	0
011	1	010	1	0	x	x	0	x	1
100	0	010	0	x	1	1	x	0	x
100	1	011	1	x	1	1	x	1	x

State table — JK flip-flop input (Flip-flop inputs)

$J_A = B'x$ $\quad K_A = 1$

$J_B = A + C'x'$ $\quad K_B = C'x + Cx'$

$J_C = Ax + A'B'x'$ $\quad K_C = x$

$y = x$

The machine is self-correcting because $K_A = 1$.

unused states:

101, 110, 111

Chapter 4

- Problem: Sequence Detector

  Design a Mealy model circuit to recognize a sequence 01010011. The circuit outputs 1 to indicate a sequence 01010011 was recognized. Otherwise, the circuit outputs 0. Please derive the state diagram, state table, next state functions and output functions, and draw the circuit diagram.

- Problem: Multiple-sequence Detector

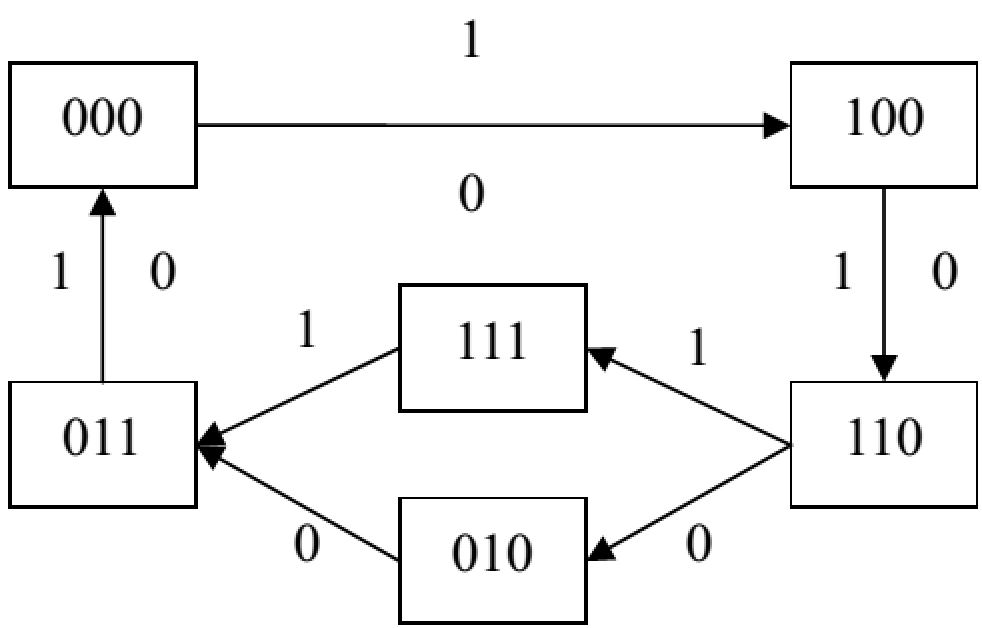
  Design a Moore model circuit to recognize two sequences 0101 and 0011. The circuit outputs 01 to indicate a sequence 0101 was recognized and outputs 11 to indicate 0011 recognized. Otherwise, the circuit outputs 00. Please derive the state diagram, state table, next state functions and output functions, and draw the circuit diagram.

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:

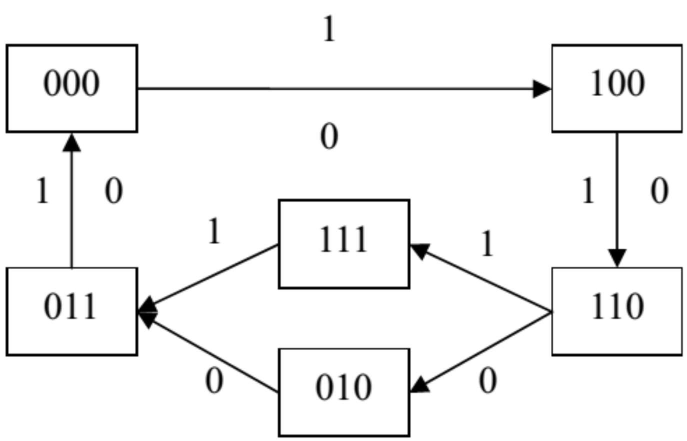


State diagram

Chapter 4

- 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

- Problem: Sequence generation

Answer:

D_2: CQ_2 \ Q_1Q_0	00	01	11	10
00	1	-	0	0
01	1	-	-	0
11	1	-	0	1
10	1	-	0	-

D_1: CQ_2 \ Q_1Q_0	00	01	11	10
00	0	-	0	1
01	1	-	-	1
11	1	-	1	1
10	0	-	0	-

D_0: CQ_2 \ Q_1Q_0	00	01	11	10
00	0	-	0	1
01	0	-	-	0
11	0	-	1	1
10	0	-	0	-

Input equations

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

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

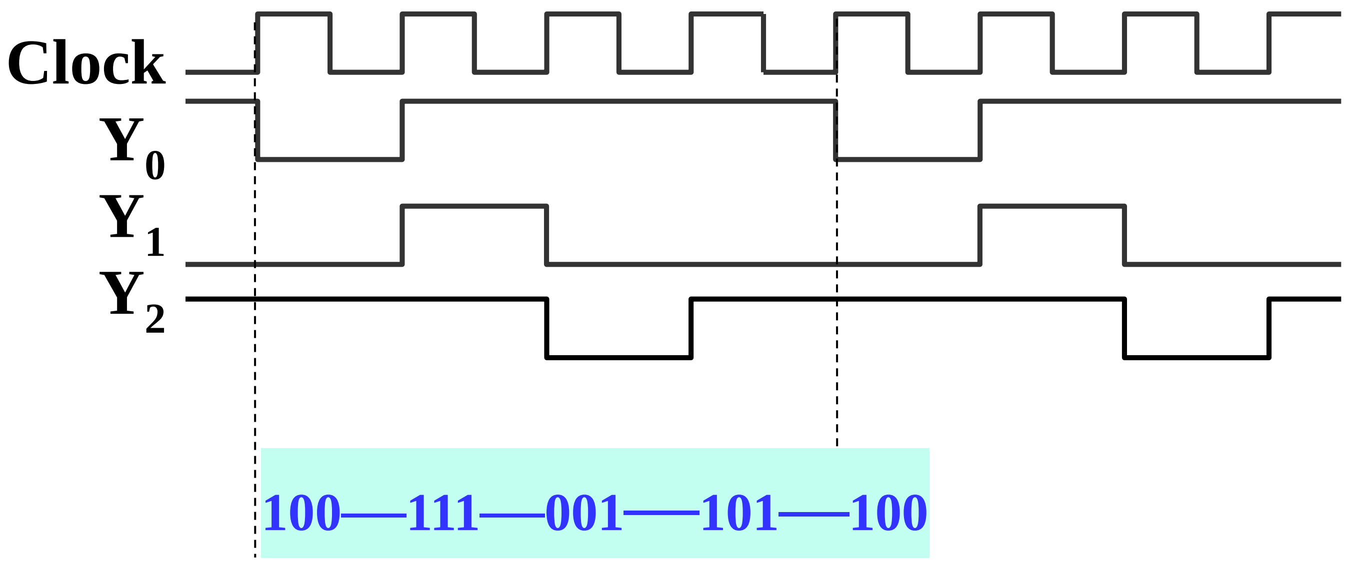
$$D_0 = CQ_2Q_1 + \overline{Q_2}Q_1\overline{Q_0}$$

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 4

- Problem: Flip-Flop Conversion

Construct a JK flip-flop using a D flip-flop, a two-to-one-line multiplexer, and an inverter.

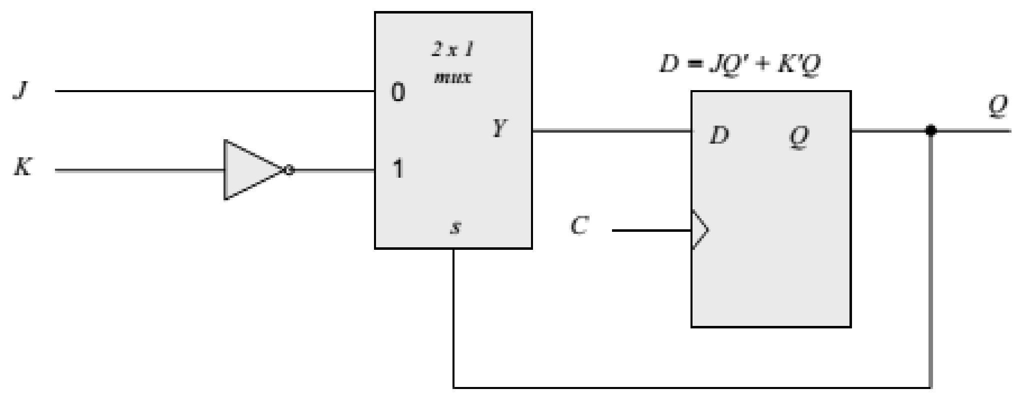
Answer:

JK flip-flop characteristic equation

$$Q(t+1) = J\overline{Q} + \overline{K}Q$$

D flip-flop excitation equation

$$D = Q(t+1) \Rightarrow D = J\overline{Q} + \overline{K}Q$$



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

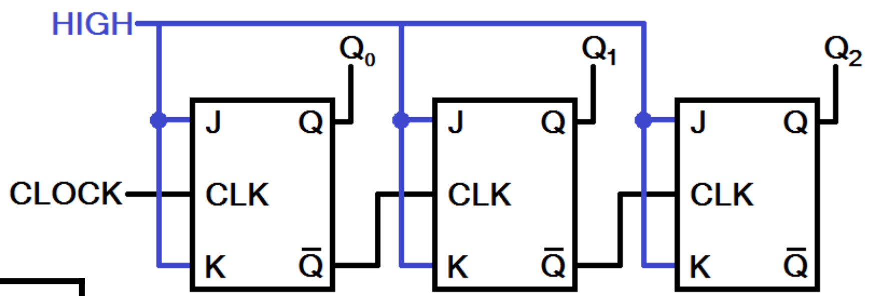
R0	Operation
10101010	$R0 \leftarrow R1$
11111010	$R0 \leftarrow R1 \vee R2$
10100000	$R0 \leftarrow R1 \wedge R2$
01011010	$R0 \leftarrow R1 \oplus R2$
01010100	$R0 \leftarrow$ sl R1

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

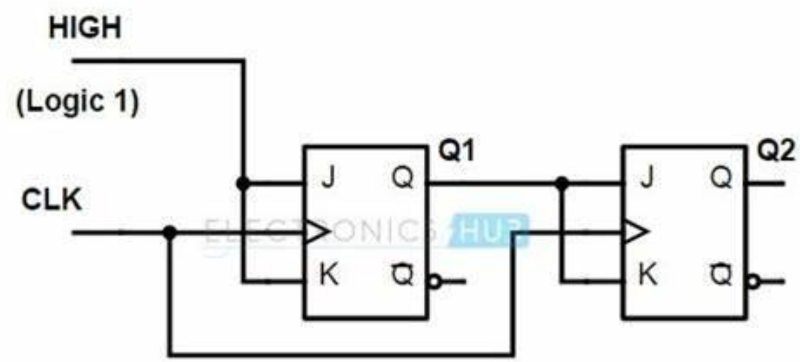
Chapter 6

- 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



Asynchronous/Ripple Counter

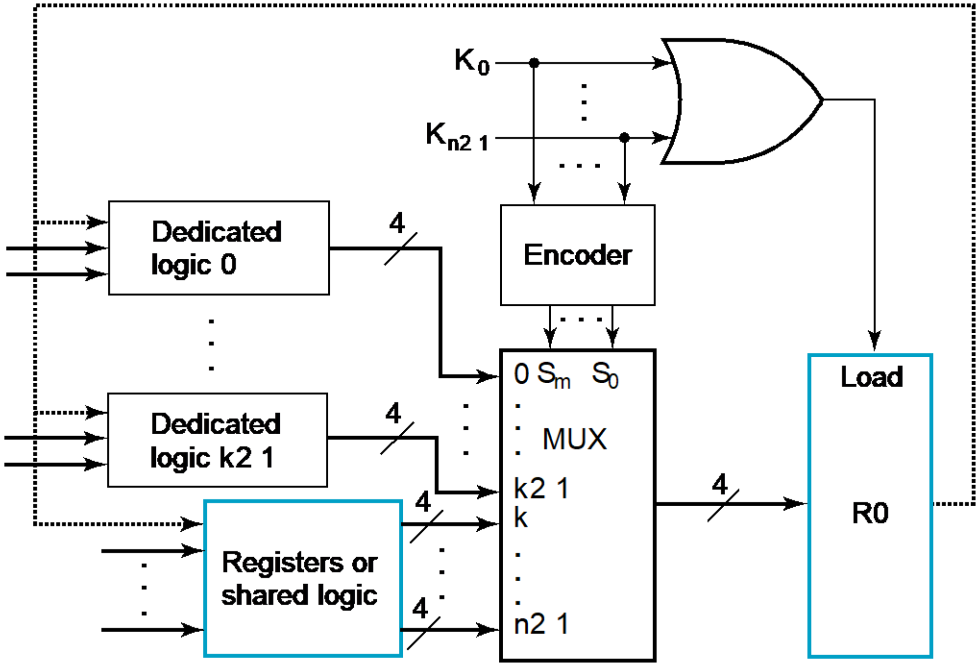


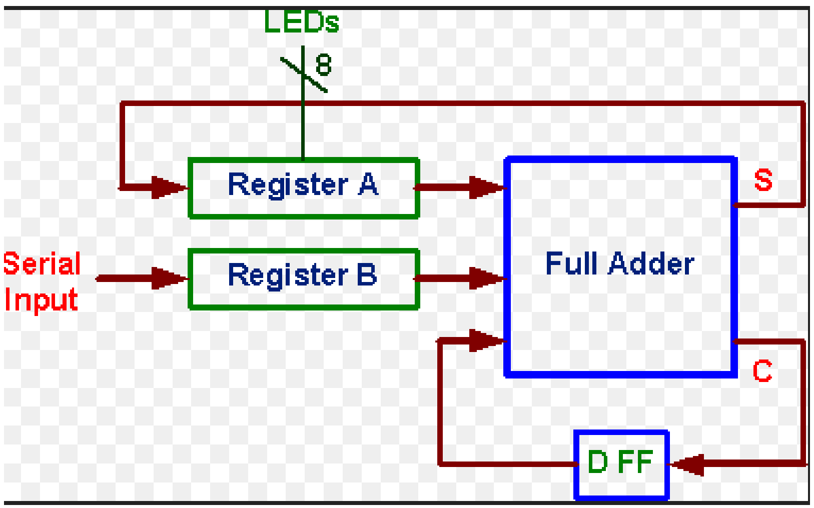
Synchronous Counter

  - 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?

Chapter 6

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

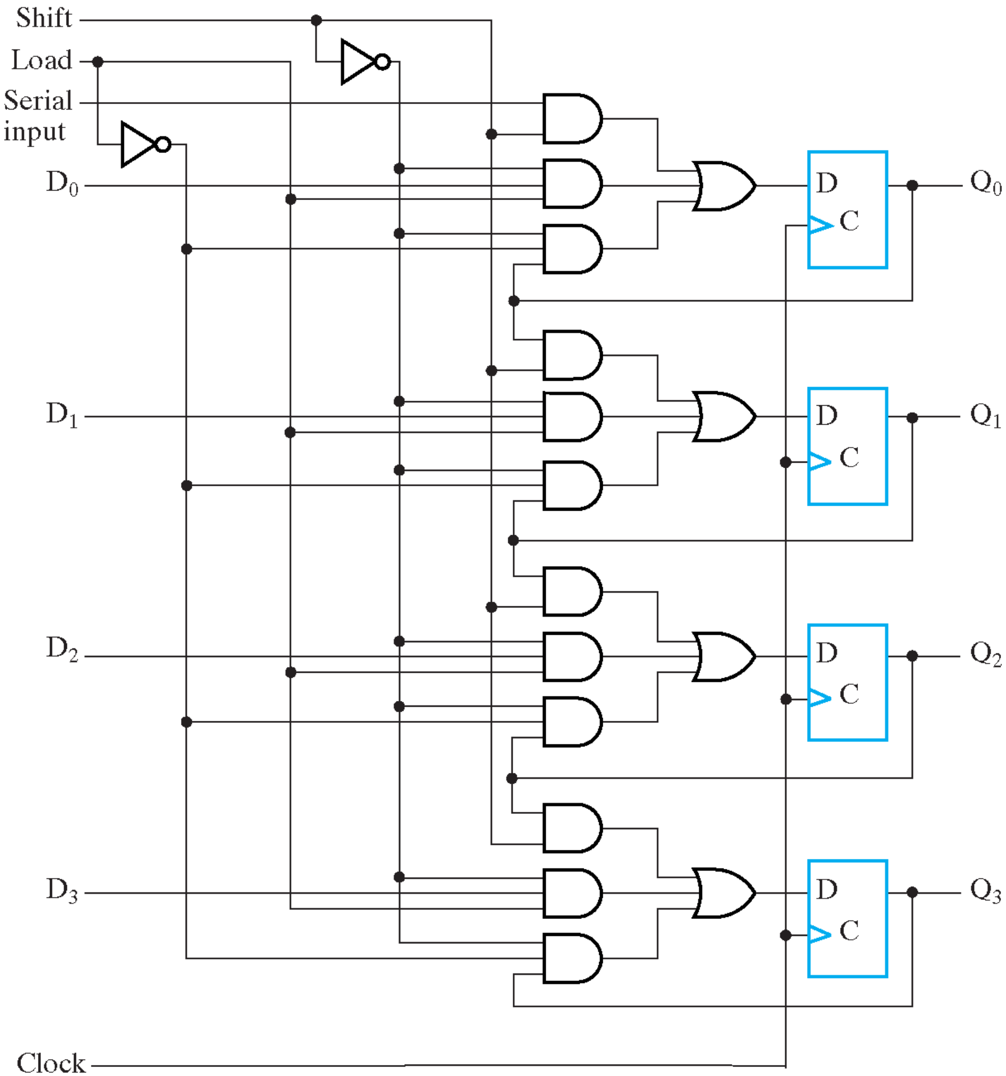




Chapter 6

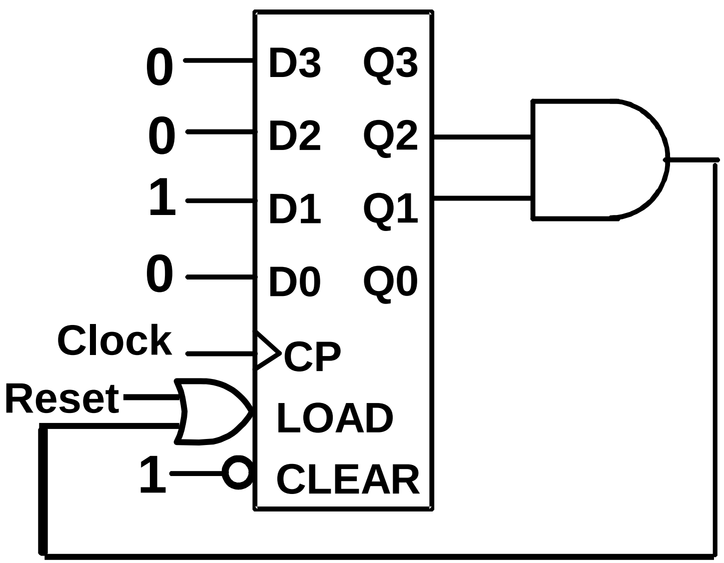
Problem: Circuit Analysis

Shift	Load	Operation
0	0	No Change
0	1	Load
1	x	Shift down

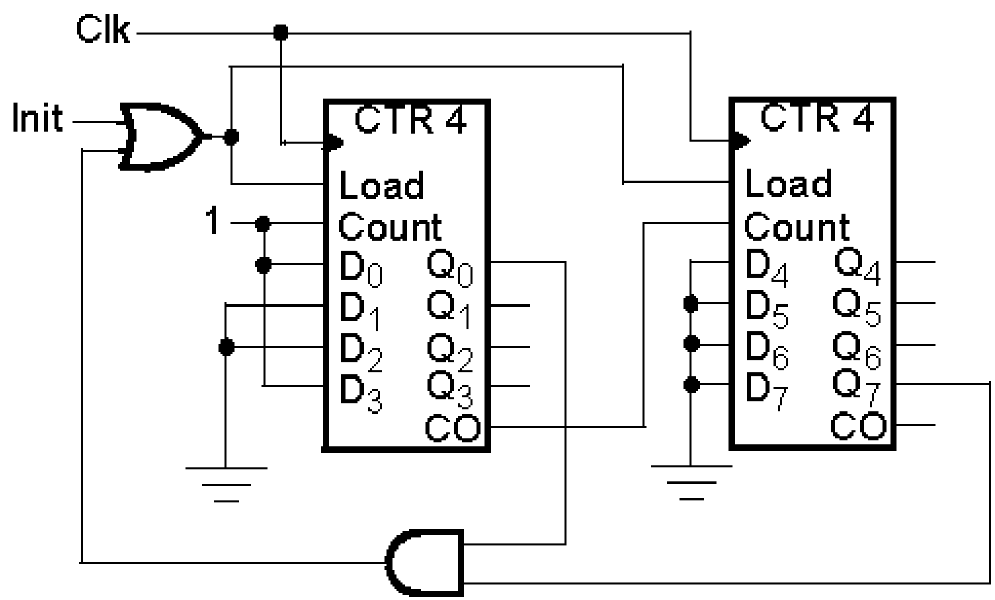


Chapter 6

Problem: Circuit Analysis



a 4-bit binary counter



a 8-bit binary counter

0010-0110

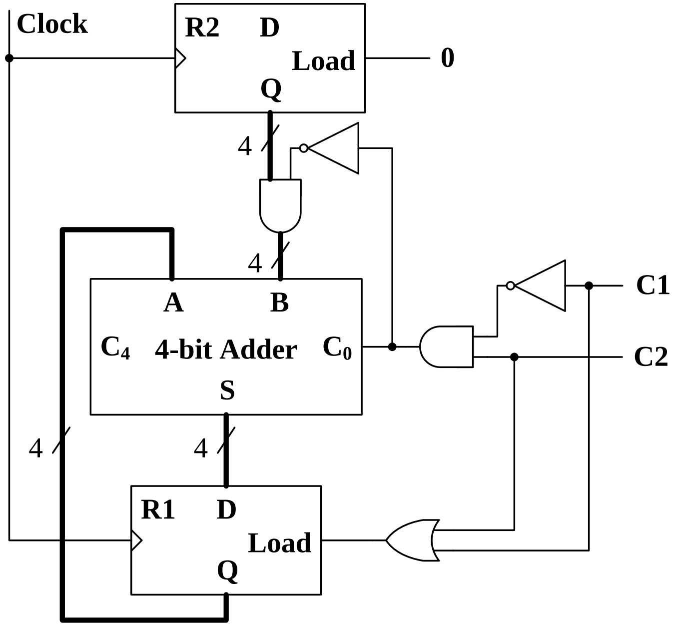
00001001-10000001

Chapter 6

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				

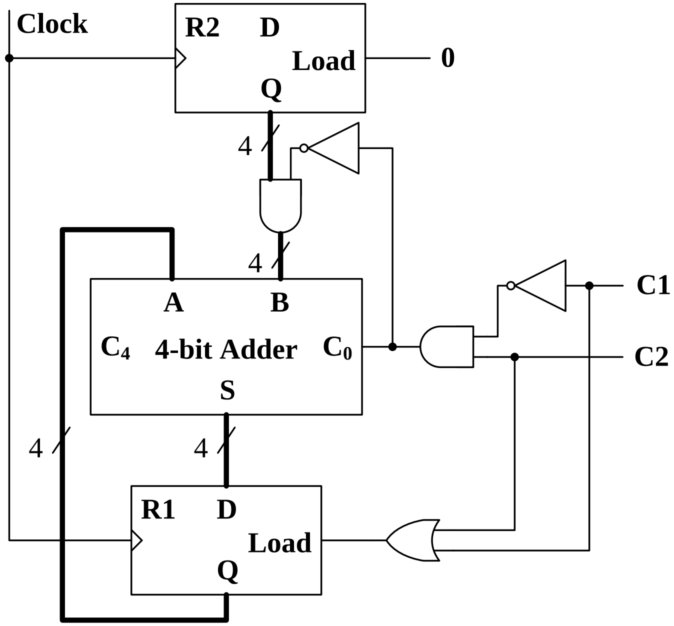


Chapter 6

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

C1'C2: R1←R1 + 1

Chapter 6

- Problem: Circuit Design

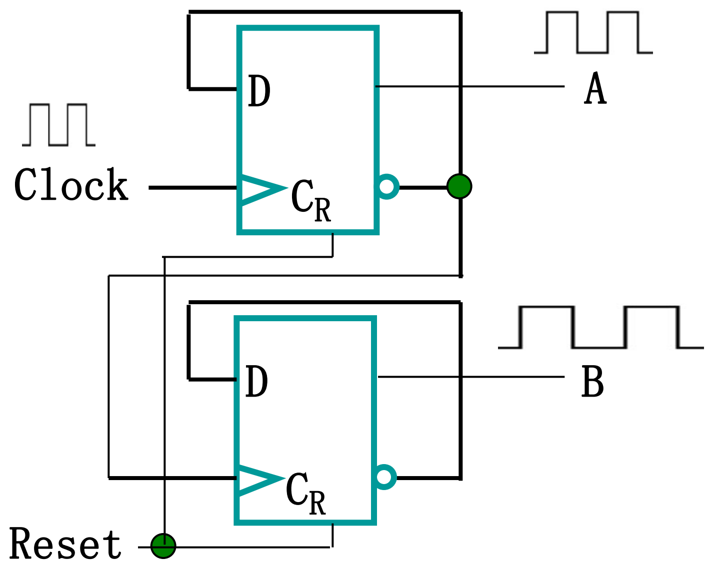
A digital system has a clock generator that produces pulses at a frequency of 80 MHz. Design a circuit that provides a clock with a cycle time of 50 ns.

Answer:

The clock generator at a frequency of 80 MHz has a period of 12.5 ns.

50/12.5 = 4

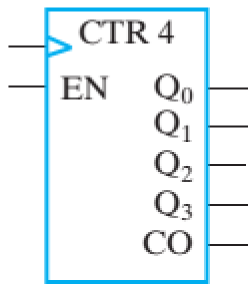
Use a 2-bit ripple counter to count four pulses.



Chapter 6

Problem: Circuit Design

Using two synchronous binary counters and logic gates to construct a minute counter that counts from BCD code “00” through BCD code “59”.



L=9: L⟵0,

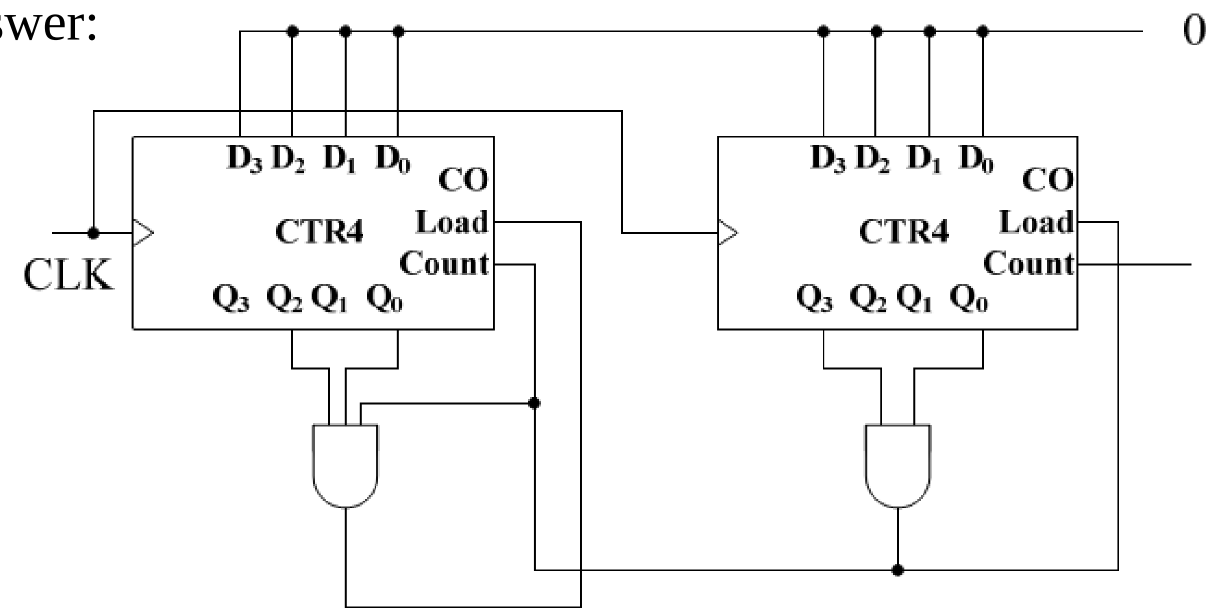
H⟵H+1

H=5, L=9:

L⟵0,

H⟵0

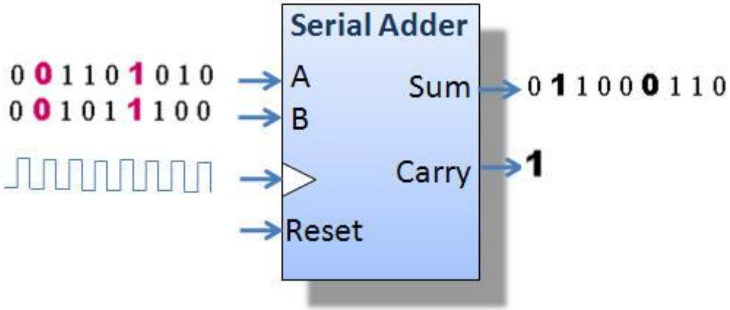
Answer:



Chapter 6

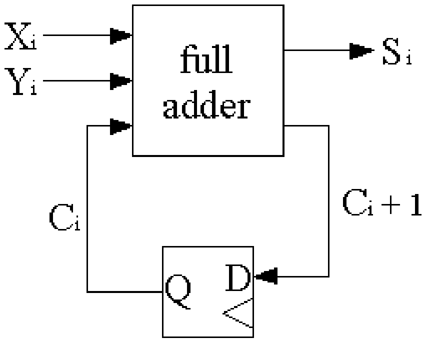
- Problem: Circuit Design

How to design a serial adder by means of sequential circuit procedure?



Answer:

Present State	Inputs		Next State	Output	Flip-Flop
Q	x	y	Q	S	D
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	1	1



$$\mathbf{D = XY+(X \oplus Y)Q} \qquad \mathbf{S = X \oplus Y \oplus Q}$$

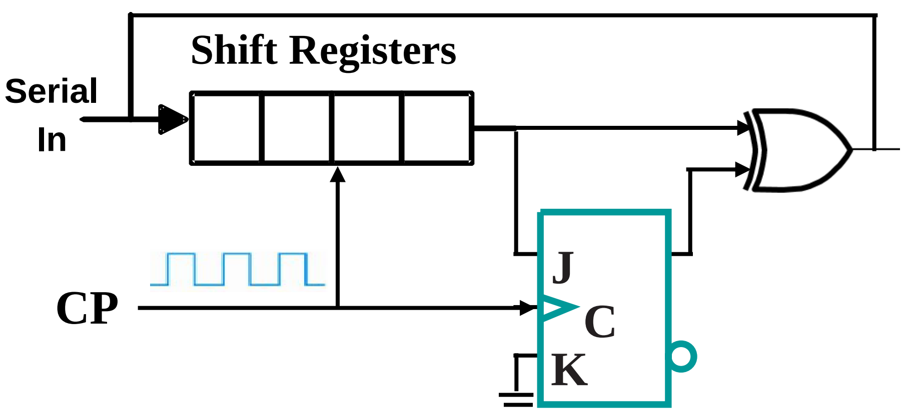
Chapter 6

- Problem: Circuit Design

Design a serial 2's complementer with a shift register and a JK flip-flop. The binary number is shifted out from one side and it's 2's complement shifted into the other side of the shift register.

Answer:

input	Present state	Next state	output	Flip-flop	
X	Q(t)	Q(t+1)	Y	J	K
0	0	0	0	0	X
0	1	1	1	X	0
1	0	1	1	1	X
1	1	1	0	X	0



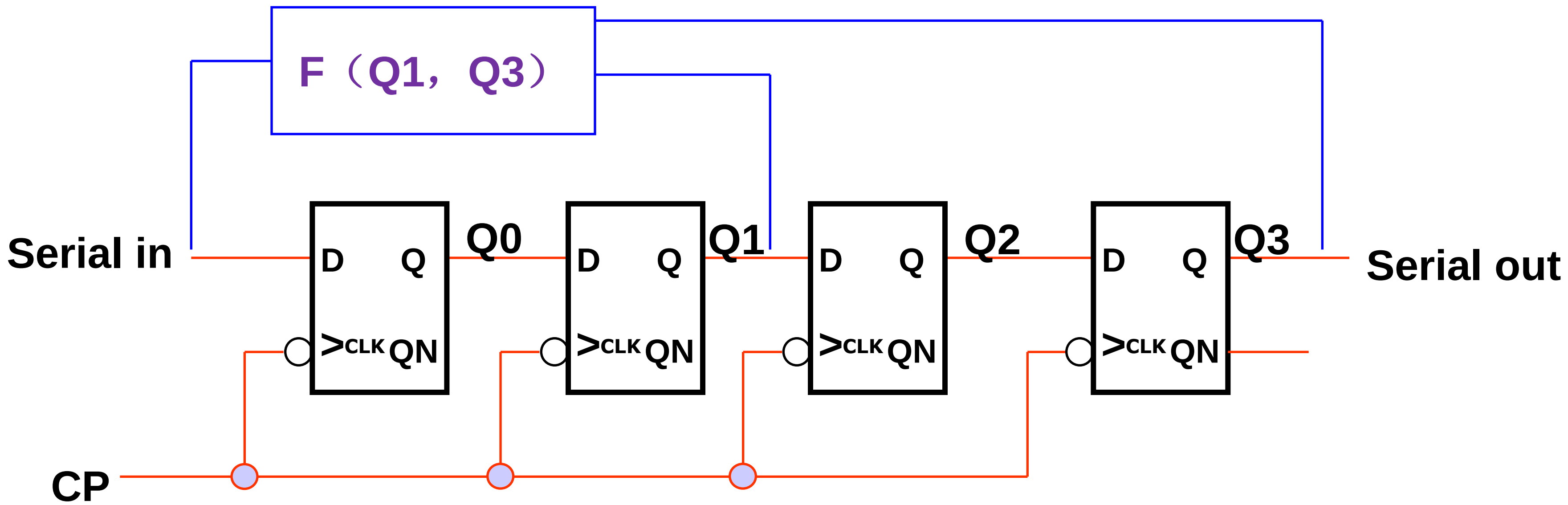
J = X

K = 0

Y = $X \oplus Q$

Chapter 6

- Problem: Linear-feedback shift register
  - A linear feedback shift register (LFSR) is the heart of any digital system that relies on pseudorandom bit sequences (PRBS).
  - An LFSR is like a black box into which you feed a number, and the generated output is some linear function of the input (typically created by some combination of shifting and XOR of the bits).

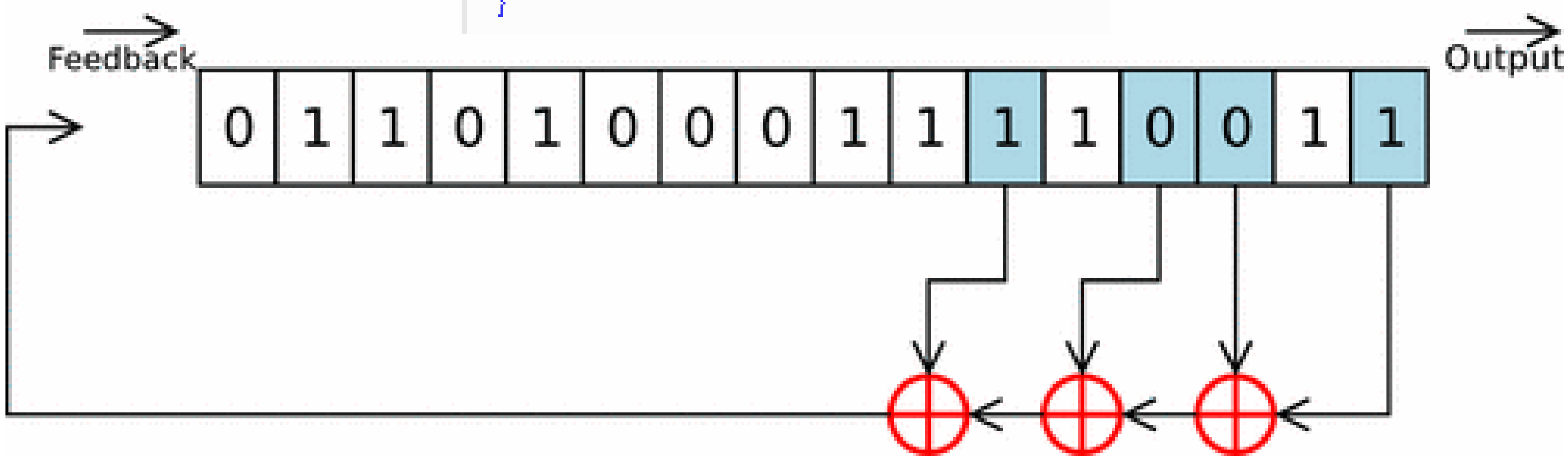


Chapter 6

- Problem: Linear-feedback shift register

A linear-feedback shift register (LFSR) is a shift register whose **input bit is a linear function of its previous state**. The initial value of the LFSR is called the **seed**.

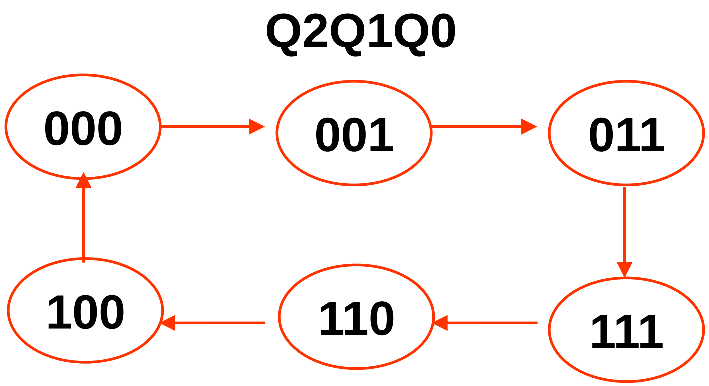
```
int main() {
    int a;
    srand((unsigned)time(NULL));
    a = rand();
    printf("%d\n", a);
    return 0;
}
```



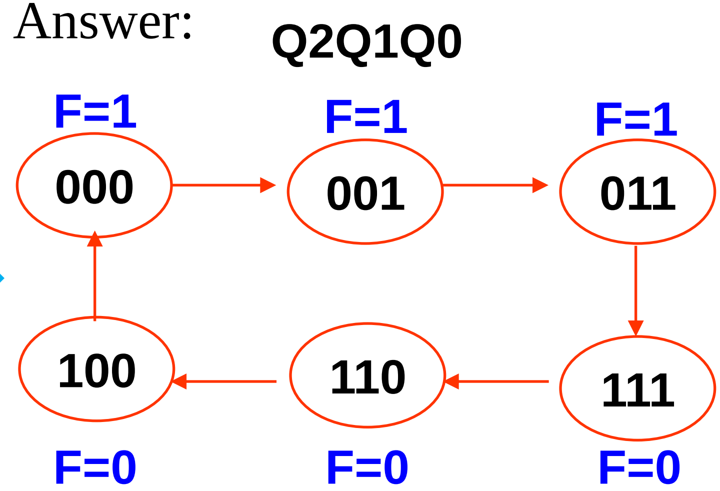
Chapter 6

- Problem: Linear-feedback shift register

Design a shift register whose state diagram appears in Figure.

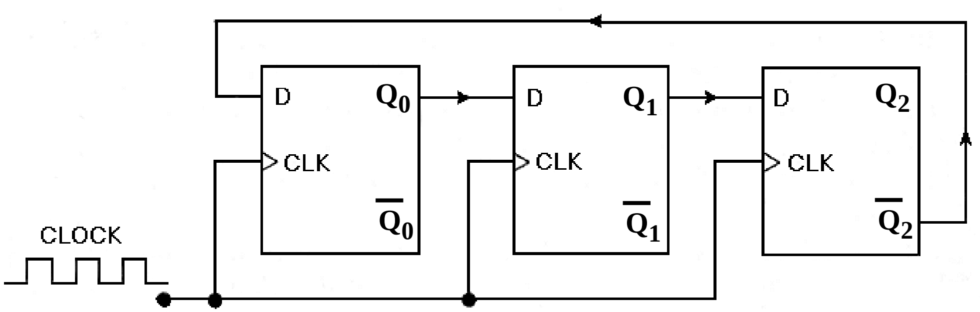


Answer:



F: Q2Q1 \ Q0	0	1
00	1	1
01	X	1
11	0	0
10	0	X

$$F = \overline{Q}_2$$



Johnson Counter

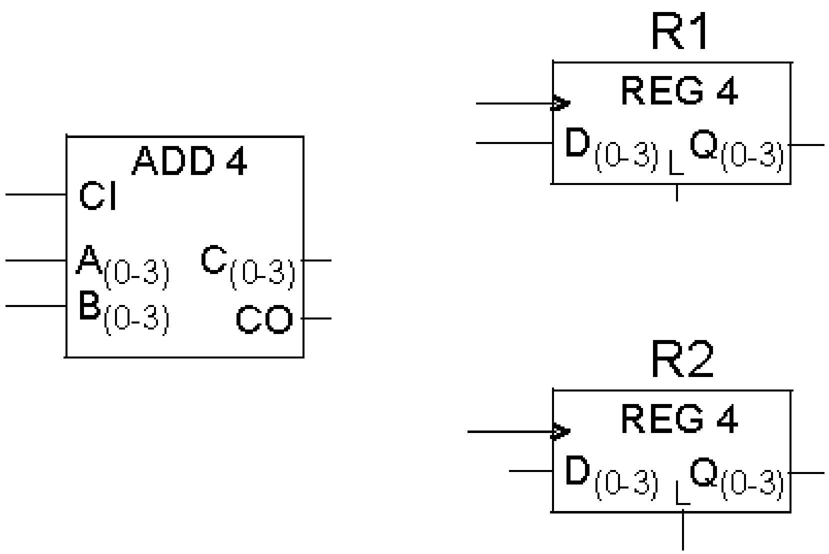
Chapter 6

- Problem: Register Transfer Operations

Two register transfer statements are given (otherwise, R1 is unchanged):

$$C1: R1 \leftarrow R1 + R2$$
$$\overline{C1}C2: R1 \leftarrow R1 - R2$$

Using a 4-bit adder plus external gates as needed. Draw the logic diagram that implements these register transfers.



Chapter 6

- Problem: Register Transfer Operations

  Two register transfer statements are given (otherwise, R1 is unchanged):

  $$C1: R1 \leftarrow R1 + R2$$
  $$\overline{C1}C2: R1 \leftarrow R1 - R2$$

  Using a 4-bit adder plus external gates as needed.

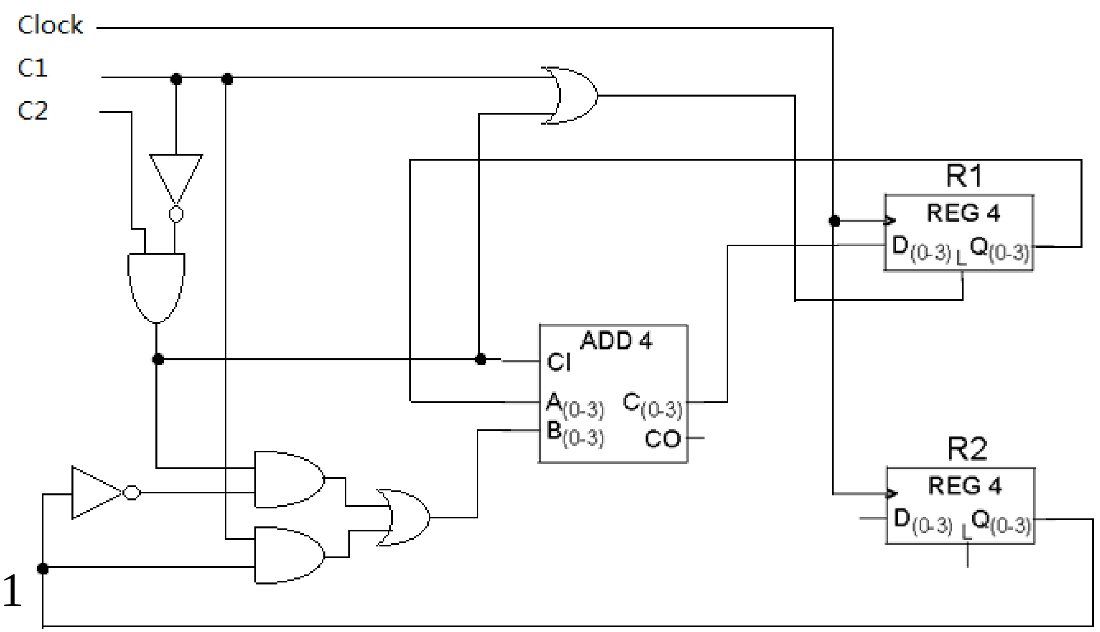
  Answer:

- Multiplexer Approach
  - Load Control

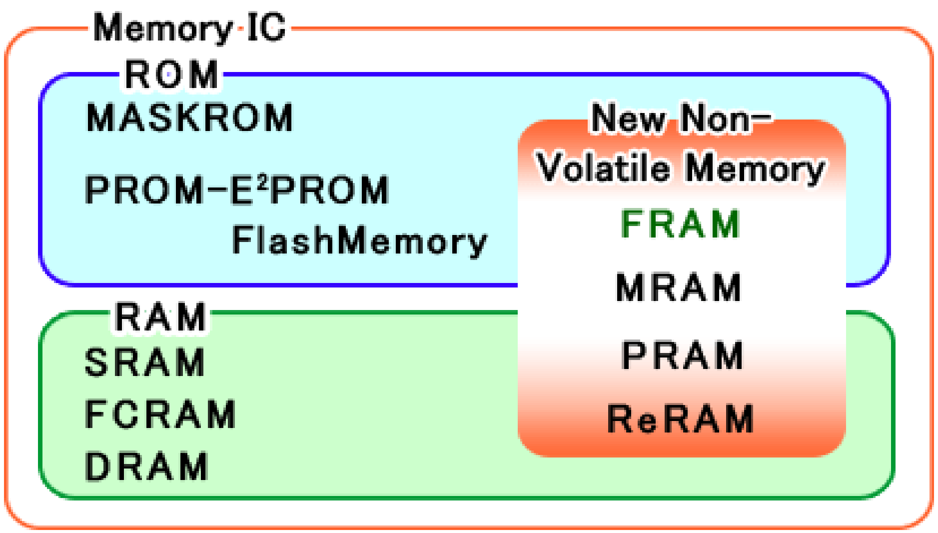
    Load = $C1 + \overline{C1}C2$ for R1
  - Multiplexer

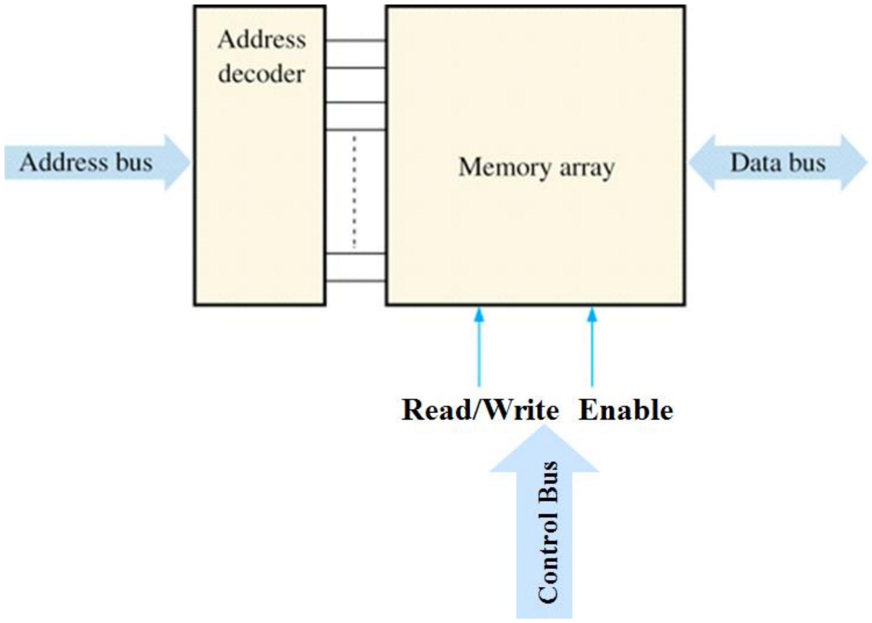
selection input
S0 = C1
S1 = C2

infor-mation input
D0 = X Hold R1
D1 = $R1 \leftarrow R1 - R2$ $\overline{C1}C2$
D2 = $R1 \leftarrow R1 + R2$ C1



Chapter 7



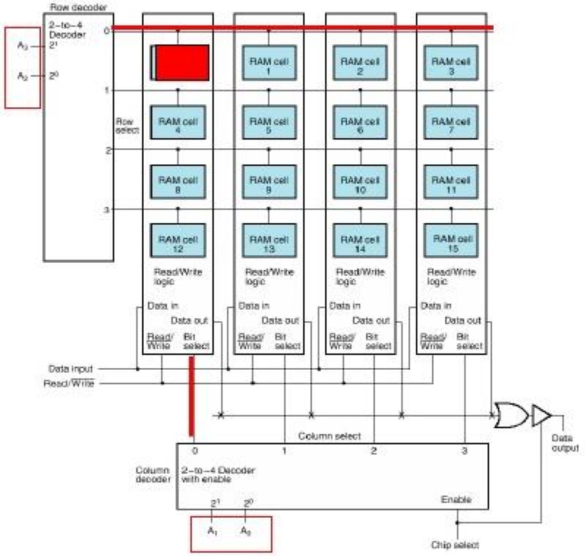


Computer memory

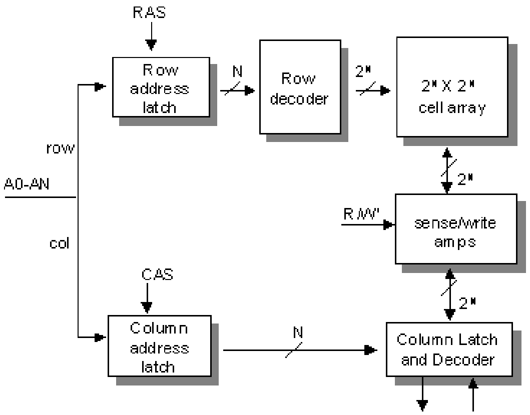
Block Diagram of RAM

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?

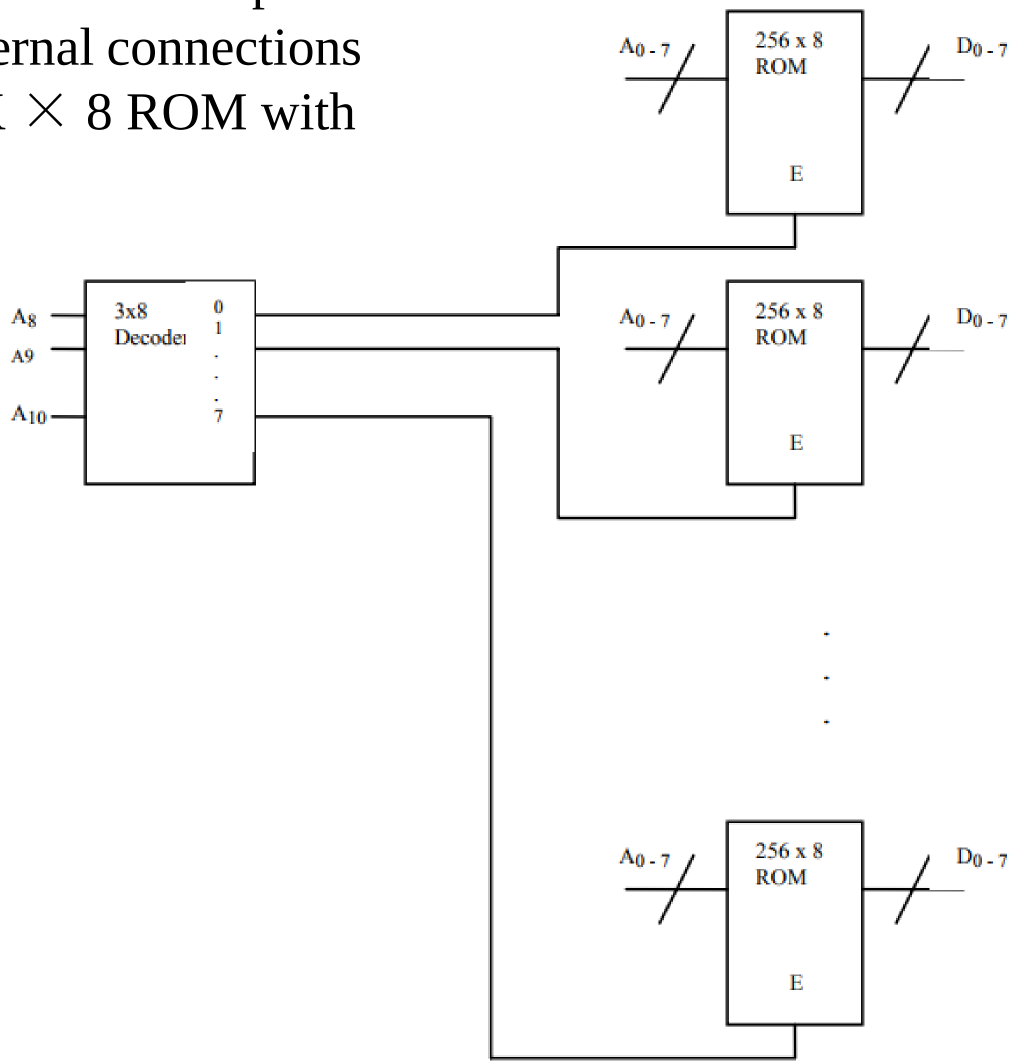
Answer:

13 + 12 = 25 address lines.

Memory capacity = 2^{25} words.

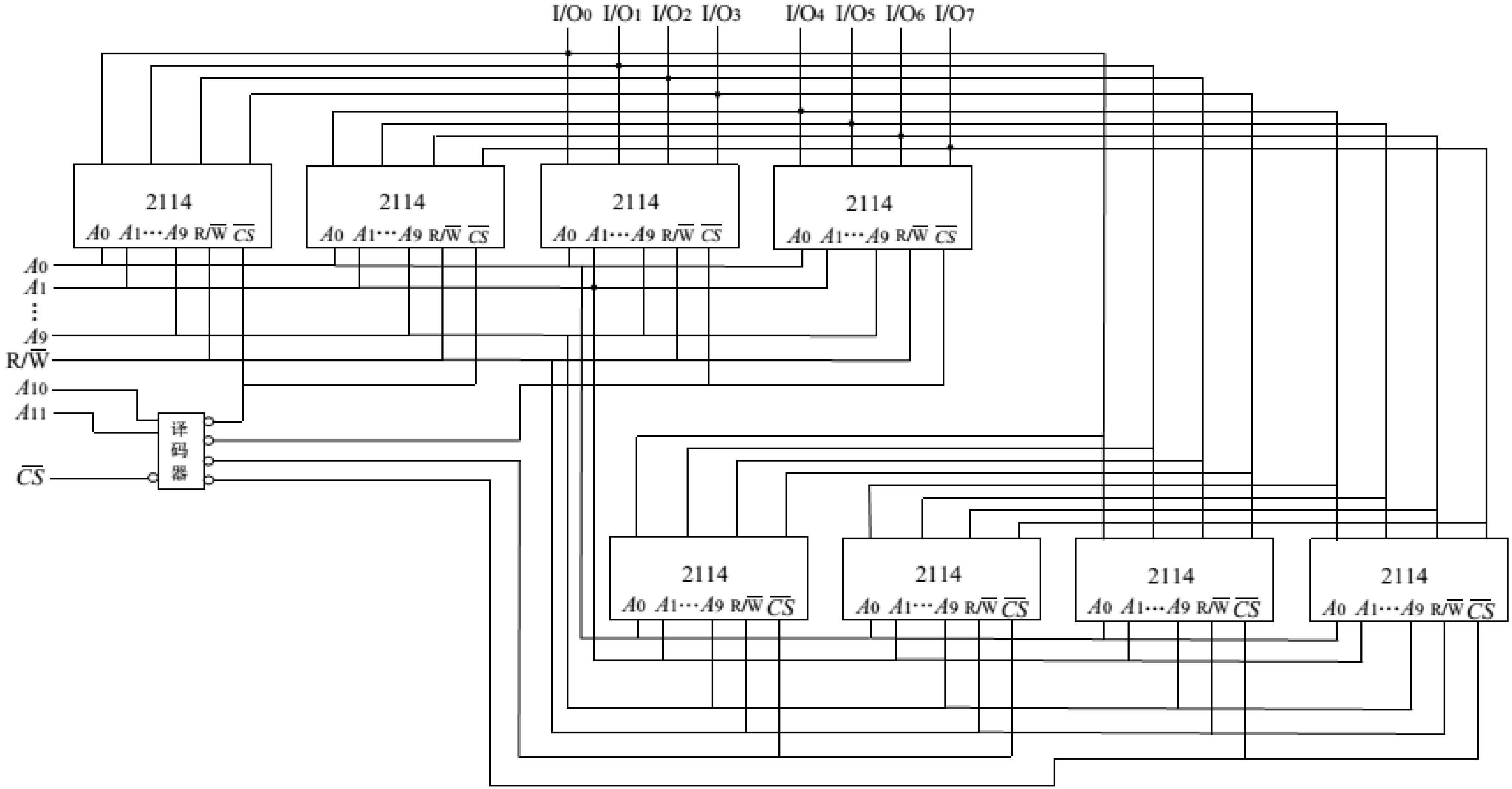
Chapter 7

- 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.

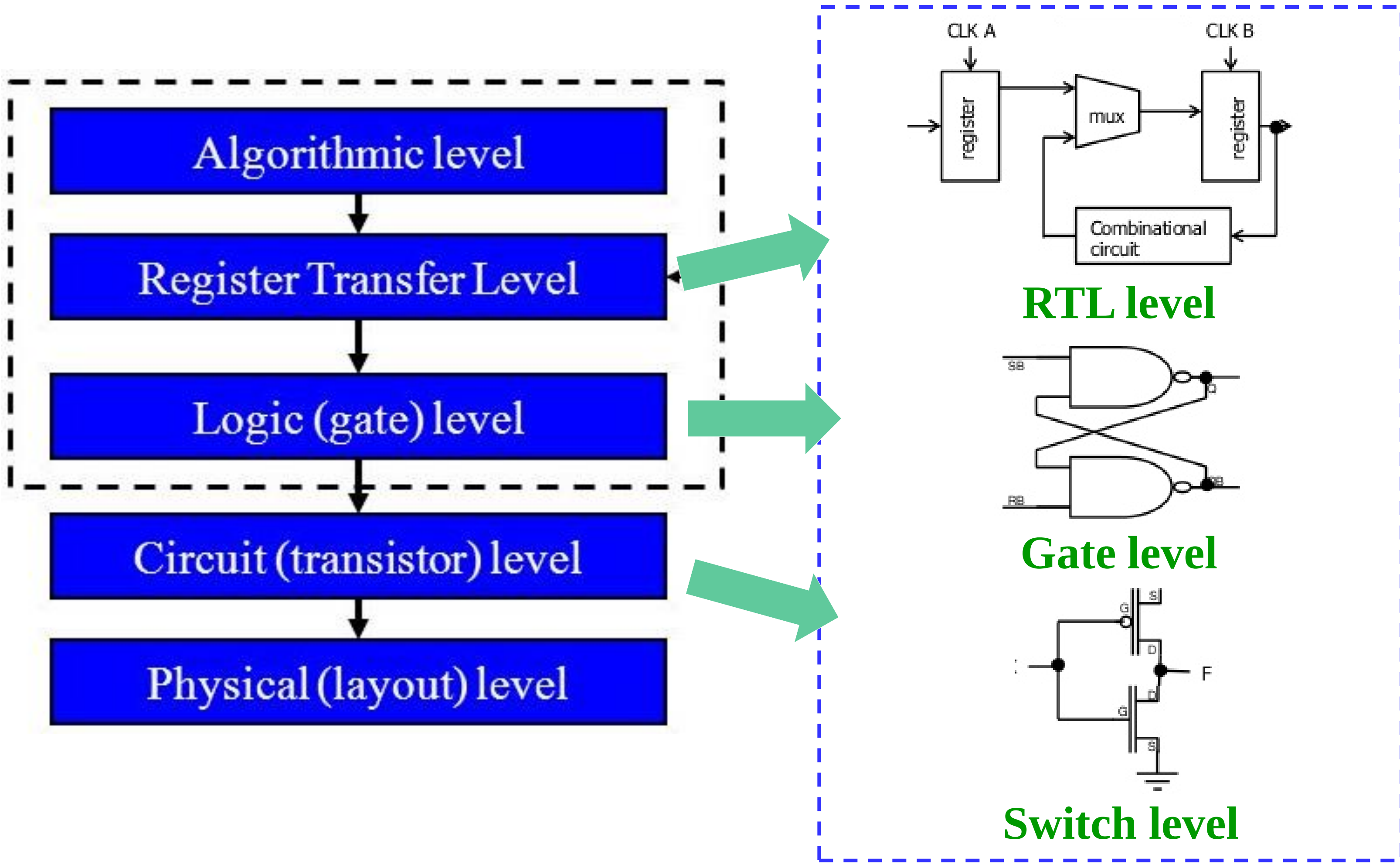


Chapter 7

- **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.

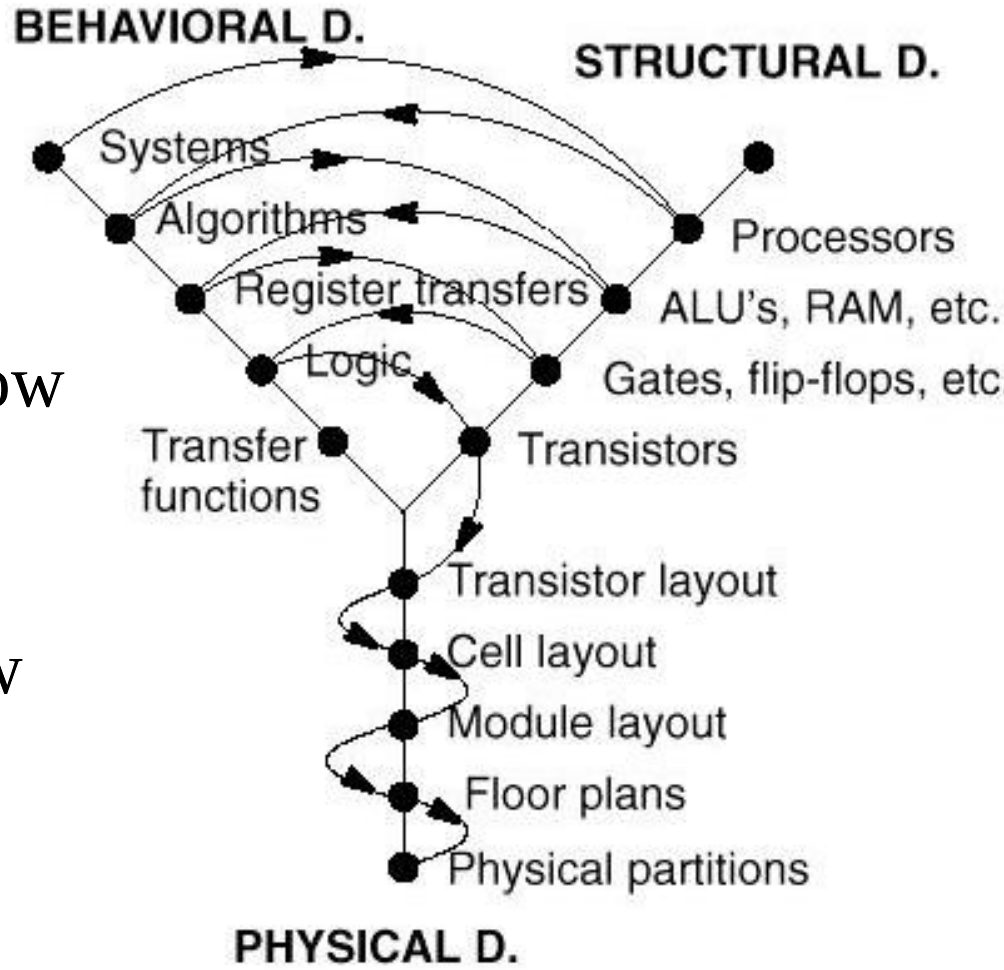


Levels of Logic Design



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
  - Timing and performance: glitch, 1's catching, t_s, t_h, t_w, t_{px}

From Behavioral Design Perspective (continued)

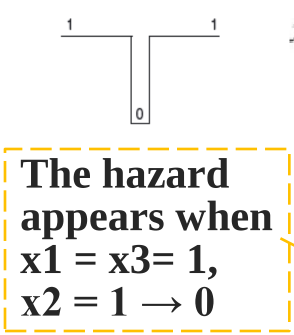
- Methods of designing digital system
  - Designing theory: register design model, datapath+control unit, state-machine diagram
  - Describing method: Register Transfer Language(RTL)
  - Timing and performance: system timing equation
    $t_p = t_{pd,FF} + t_{pd,COMB} + t_s + t_{slack}$
- Methods of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

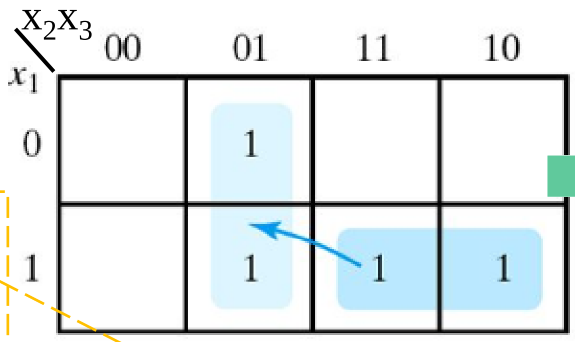
Design Tradeoffs in Logic Circuits

- 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

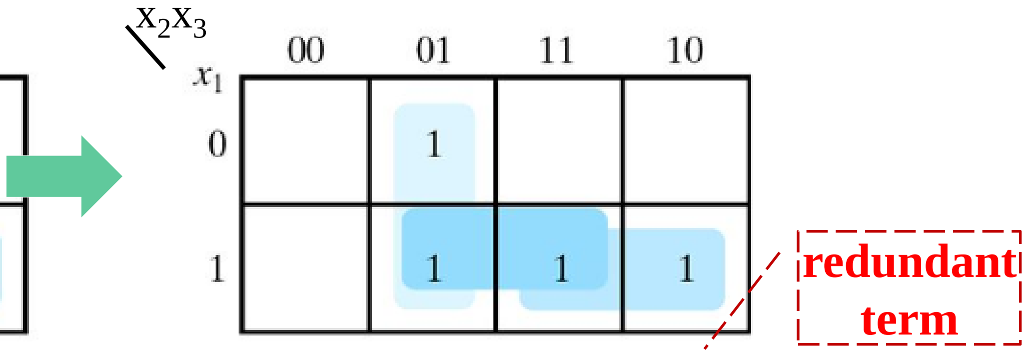
Design Tradeoffs in Logic Circuits (continued)

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





(a) $Y = x_1x_2 + x'_2x_3$



(b) $Y = x_1x_2 + x'_2x_3 + \boxed{x_1x_3}$

End

- Tutorial time:

  Jan. 21 TBD

- Final Examination Time:

  Jan. 22, 2019 10:30~12:30

- Good luck!