电子技术
Introduction to Electronics

By Bao Qilian
鲍其莲

Chapter 8 Counters

Objectives

- Asynchronous counter Operation
- Synchronous counter Operation
- Up/Down Asynchronous counter Operation
- Design of Synchronous Counters
- Analysis and Design of Sequence Circuit

- Cascaded Counters
- Counter Decoding
- Counter Application

Reading assignments

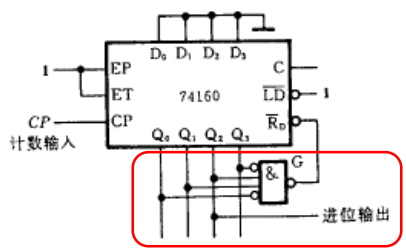
- P454-P493

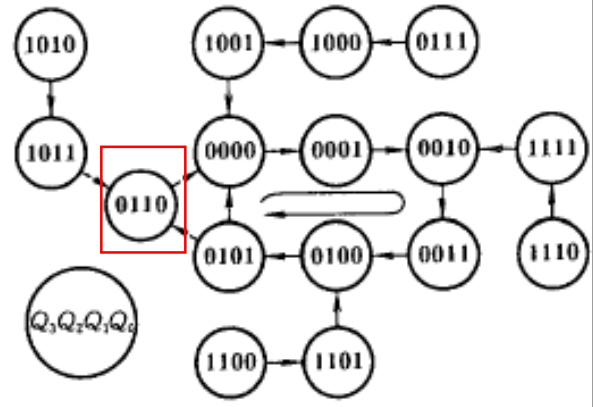
Design of Any Modulus Counters

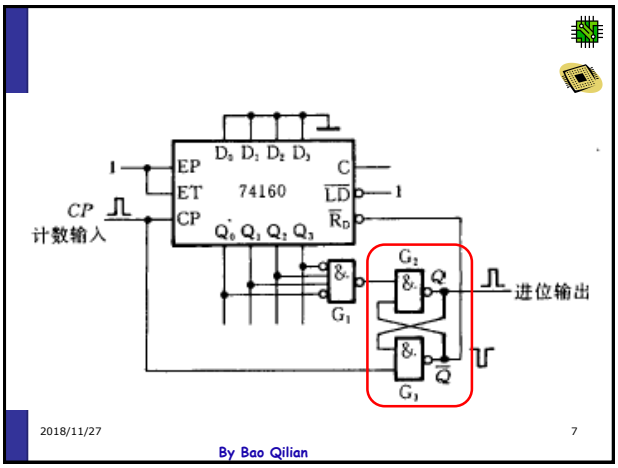
- RESET-based Design
- LOAD-based Design

RESET-based Counter Design

- Example:



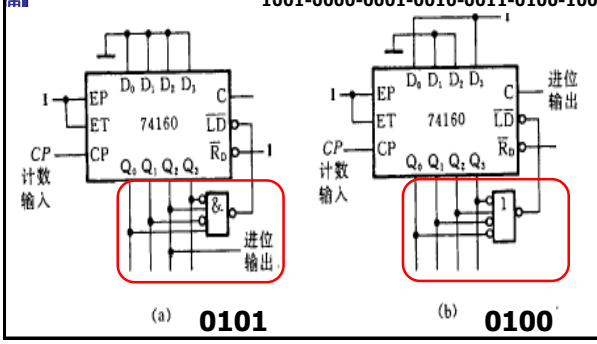


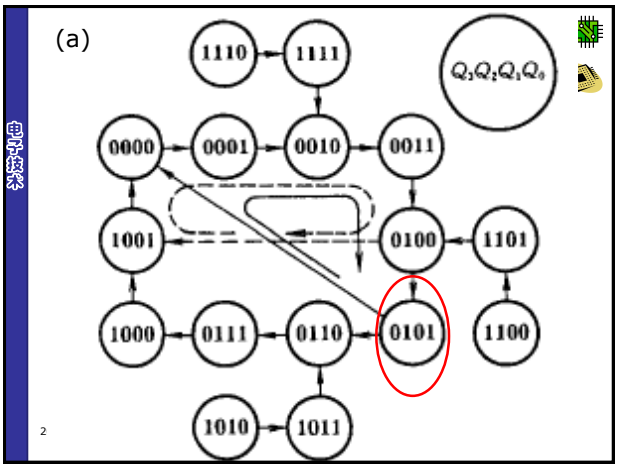


By Bao Qilian

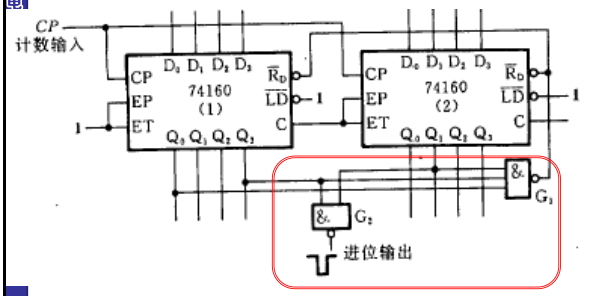
LOAD-based Counter Design

1001-0000-0001-0010-0011-0100-1001



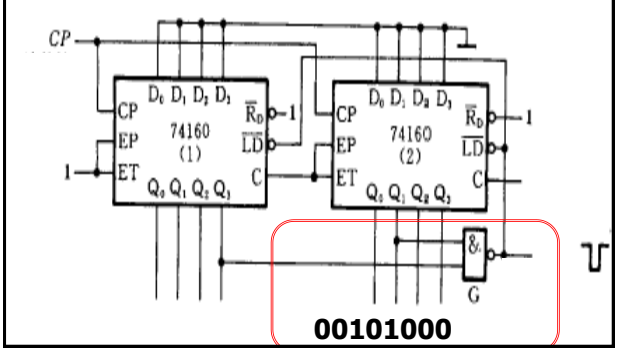


RESET-based cascaded counters



By Bao Qilian 00101001

LOAD-based cascaded counters



00101000

小结：任意进制计数器的构成方法

假定已有N进制计数器,需要得到M进制计数器

1. M<N 的情况

在N进制计数器的顺序计数过程中,

设法使之跳过N-M个状态,就可得到M进制计数器

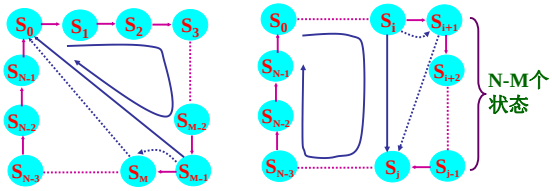
实现跳跃的方式有 置零法 和 置数法

置零法 适用于有置零输入端的计数器

置数法 适用于有预置数功能的计数器

By Bao Qilian

置零法　　置数法



异步清零信号从S_M译出，
同步清零信号从S_{M-1}译出

异步置数信号从S_{i+1}译出，
同步置数信号从S_i译出

2. M > N 的情况

必须使用多片N进制计数器构成M进制计数器

各片之间的连接方式：

串行进位、并行进位、整体置零、整体置数

若M可以分解为两个小于N的因数相乘即
$M=N_1 \times N_2$

可采用串行进位或并行进位方式

串行进位：低位进位输出信号作为高位时钟信号

并行进位：低位进位输出信号作为高位计数使能信号

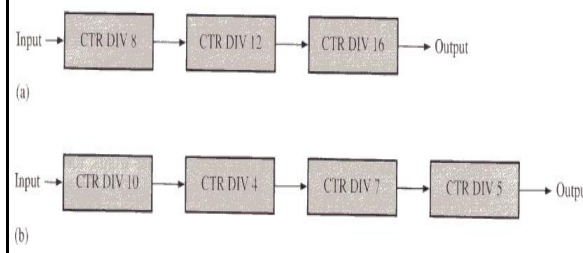
当M为大于N的素数时，不能分解为N_1和N_2

必须采取整体置零或整体置数方式

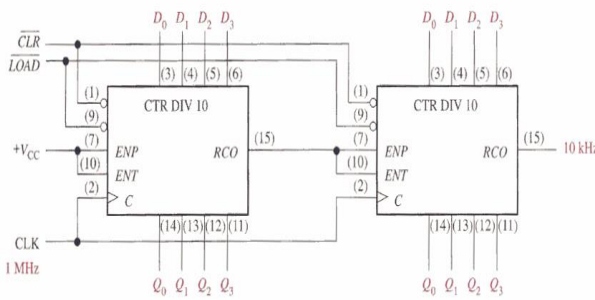
整体置零：首先将两片N进制计数器按最简单的方式
接成一个大于M的计数器（如N*N进制）
然后从M状态译出置零信号
将两片N进制计数器同时置零
基本原理和 M < N 时置零法一样

整体置数：基本原理和 M < N 时置数法类似

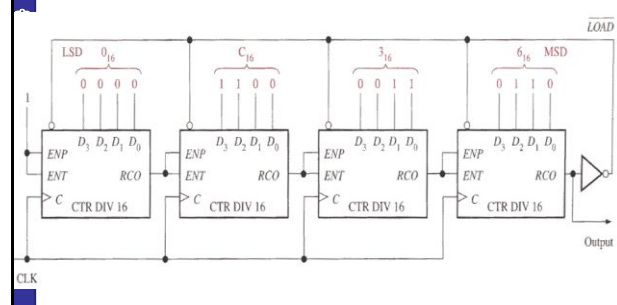
Exercise: Determine the modulus



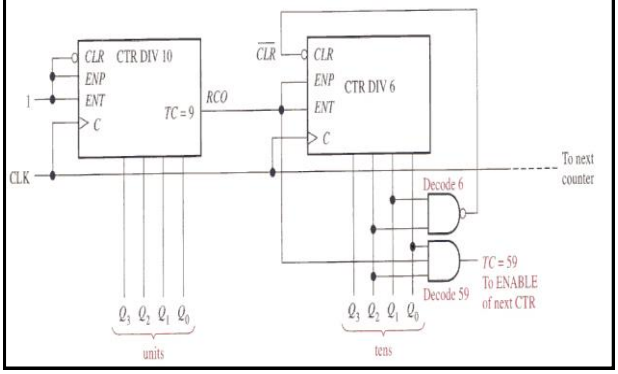
A divide-by-100 counter using two 74HC160



Divde-by-40,000 counter using 74HC161



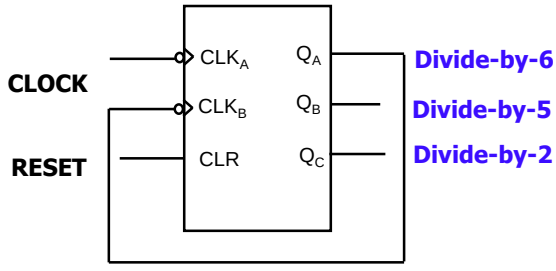
Exercise



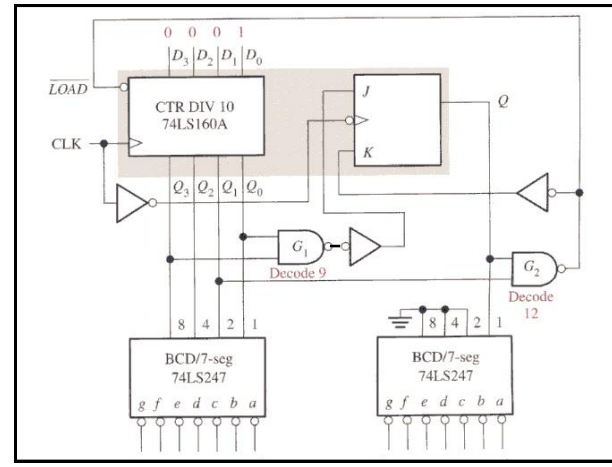
- Divide-by-60 counter

By Bao Qilian

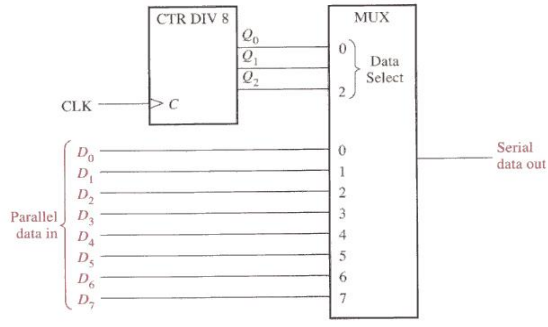
MSI CHIP: 74LS57 (divide-by-60)



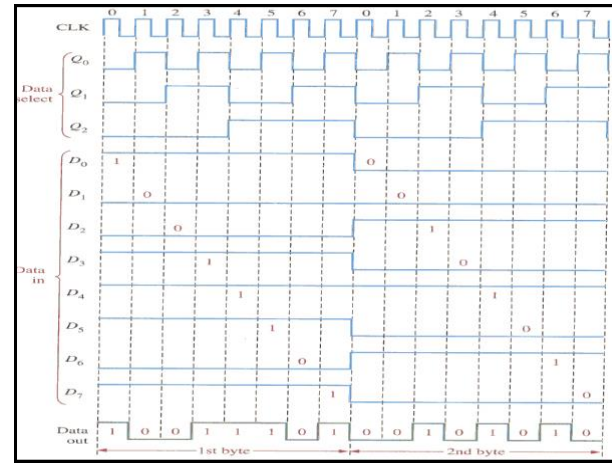
By Bao Qilian



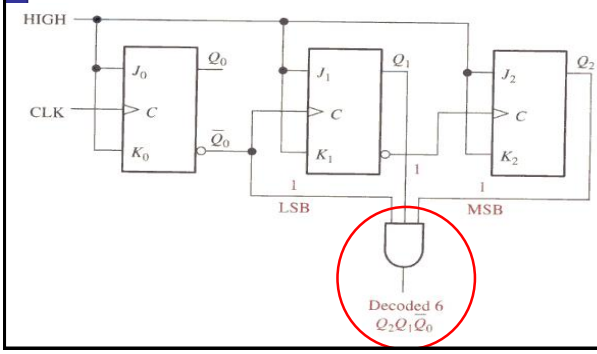
Parallel-to-serial conversion

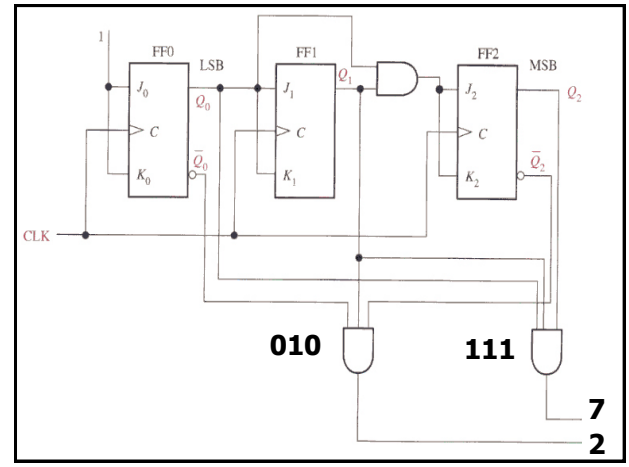


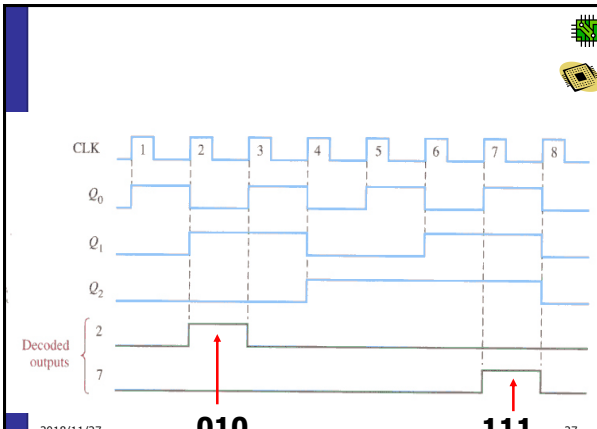
By Bao Qilian



8.5 Counter Decoding



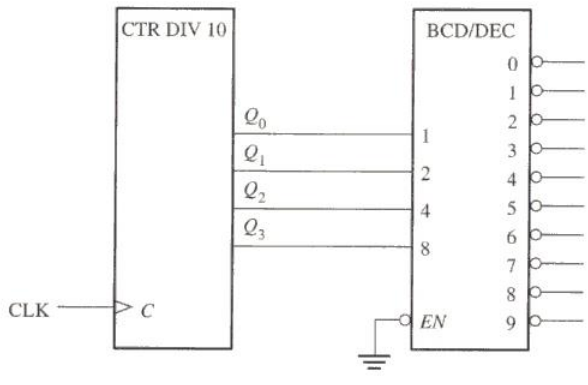


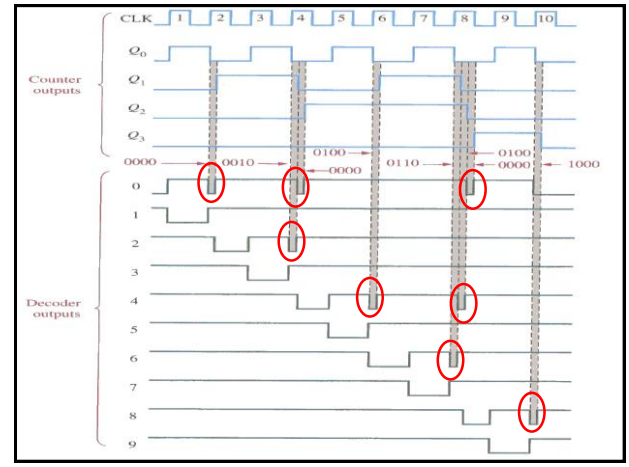


Glitches Problem

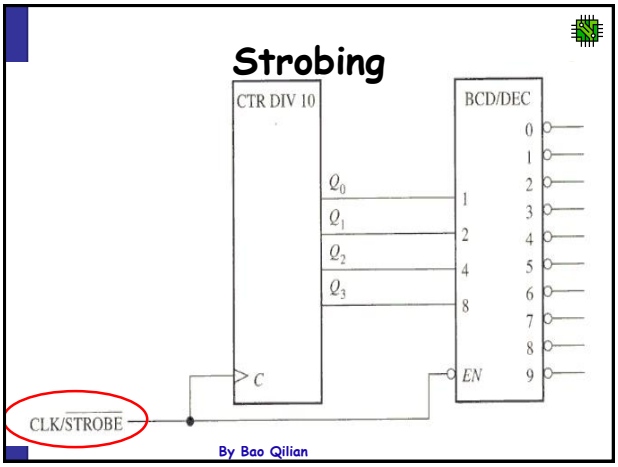
- The difference of propagation time produces undesired spikes of short duration.

Asynchronous Counter

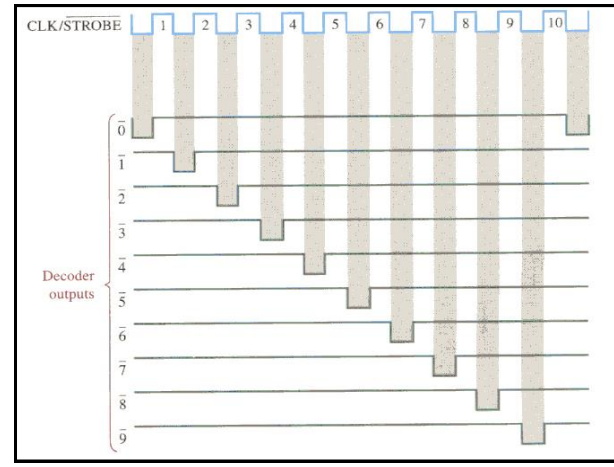




Strobing



By Bao Qilian

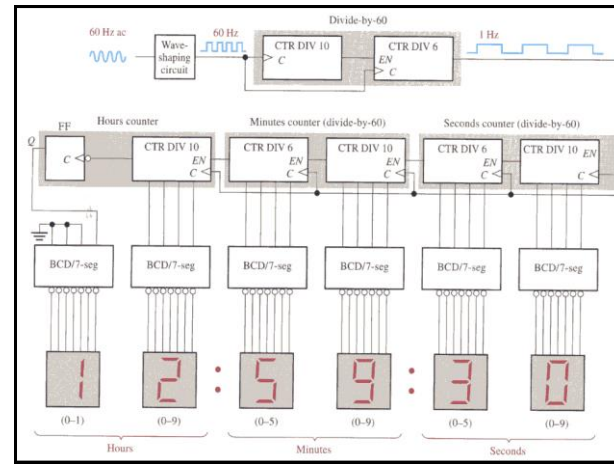


Application Examples

电子技术

2018/11/27 33

By Bao Qilian



MSI CHIP	RESET （清零）	LOAD (置数)	MODULE (进制)
74LS160	异步 Asynchronous	同步 Synchronous	10
74LS161	异步 Asynchronous	同步 Synchronous	16
74LS163	同步 Synchronous	同步 Synchronous	16
74LS190	无 none	异步 Asynchronous	(加/减)10 Up/down 10
74LS191	无 none	异步 Asynchronous	Up/down 16

35

By Bao Qilian

8.6 Design and Analysis of Synchronous Sequential Circuits

电子技术

2018/11/27 36

By Bao Qilian

Synchronous Sequential Circuit Models:

- *Structural*
  - Logic diagram
  - Excitation Equations
  - Output equations
- *Behavioral*
  - Transition and output equations
  - Transition table
  - State table (Next-state table)
  - State diagram (graph)

- SSC Analysis: Derive one of the behavioral models from an instance of a structural model
- SSC Design : Derive a structural model from one of the behavioral models

1 Analysis of Synchronous Sequential Circuits

Step1: Derive the excitation and output equations

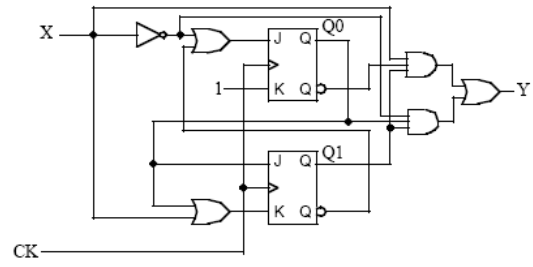
Step2: Derive the transition equations

Step3: Construct the transition table

Step4: Construction the next-state table

Step5: Draw state diagram

Example



Step1: Derive the excitation and output equations

Excitation equation:

$$J_0 = \overline{X} + \overline{Q}_1$$
$$K_0 = 1$$
$$J_1 = Q_0$$
$$K_1 = X + Q_0$$

Output equation:

$$Y = X\overline{Q}_0 Q_1 + \overline{X} Q_0 Q_1$$

Step2: Derive the transition equations

- Use the ***characteristic equation*** for J-K flip-flops: $Q^{n+1} = J\overline{Q} + \overline{K}Q$

☐ The resulting transition equations are:

$$Q_0^{n+1} = J_0\overline{Q}_0 + \overline{K}_0 Q_0 = (\overline{X} + \overline{Q}_1)\overline{Q}_0 + 0 \cdot Q_0$$
$$= \overline{X}\overline{Q}_0 + \overline{Q}_0\overline{Q}_1$$
$$Q_1^{n+1} = J_1\overline{Q}_1 + \overline{K}_1 Q_1 = Q_0\overline{Q}_1 + \overline{X + Q_0} Q_1$$
$$= Q_0\overline{Q}_1 + \overline{X}\overline{Q}_0 Q_1$$

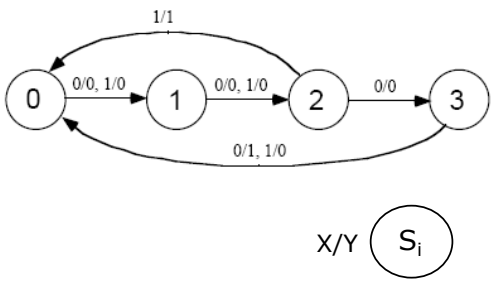
Step3: Construct the transition table

	Q_1	Q_0	X=0 Q_1^{n+1}	X=0 Q_0^{n+1}	X=0 Y	X=1 Q_1^{n+1}	X=1 Q_0^{n+1}	X=1 Y
0	0	0	0	1	0	0	1	0
1	0	1	1	0	0	1	0	0
2	1	0	1	1	0	0	0	1
3	1	1	0	0	1	0	0	0

Step4: Construction the next-state table

	X=0		X=1	
Q	Q	Y	Q	Y
0	1	0	1	0
1	2	0	2	0
2	3	0	0	1
3	0	1	0	0

Step5: Draw state diagram



X/Y S_i

同步时序逻辑电路的分析方法

分析一个时序电路，就是要找出给定电路的逻辑功能（状态表、状态图或时序图）。

一般步骤：

1. 写方程式

时钟方程：各个触发器时钟信号的逻辑表达式；

输出方程：时序电路各个输出信号的逻辑表达式

驱动方程：各个触发器同步输入端信号的逻辑表达式。

2. 求状态方程

把驱动方程代入相应触发器的特性方程即可求出。

3. 进行计算

把电路输入和现态的各种可能取值，代入状态方程和输出方程进行计算，求出相应的次态和输出方程

注意事项：

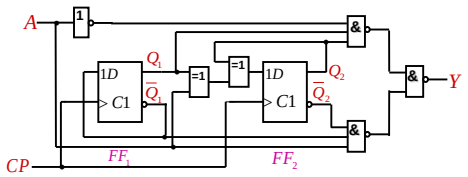
- 状态方程有效的时钟条件，凡不具备时钟条件者，触发器将保持原状态不变；
- 电路的现态；
- 不能漏掉任何可能出现的状态和输入的取值；
- 若给定了起始值，可从给定起始值开始依次计算，否则可自设起始值。

4. 画状态图或列状态表、画时序图

注意事项：

- 状态转换是由现态转换到次态；
- 输出是现态和输入的函数；
- 画时序图时注意时钟条件。

5. 说明电路功能

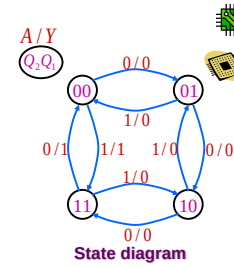


Solution:

Excitation equation: $\begin{cases} D_1 = \overline{Q}_1 \\ D_2 = A \oplus Q_1 \oplus Q_2 \end{cases}$ **State euqation:** $\begin{cases} Q_1^{n+1} = D_1 = \overline{Q}_1 \\ Q_2^{n+1} = D_2 = A \oplus Q_1 \oplus Q_2 \end{cases}$

Output euqation $Y = \overline{\overline{A Q_1 Q_2} \cdot \overline{A \overline{Q}_1 \overline{Q}_2}} = \overline{A} Q_1 Q_2 + A \overline{Q}_1 \overline{Q}_2$

A \ $Q_2^{n+1}Q_1^{n+1}$ / $Q_2^nQ_1^n$	00	01	11	10
0	01/0	10/0	00/1	11/0
1	11/1	00/0	10/0	01/0



State diagram

Function：A=1 count down

A=0 count up

同步时序电路的分析小结

- 求输出函数，触发器激励函数(控制函数)
- 根据输入和触发器激励函数求状态表
- 画状态图，时序图，分析电路功能
- 分析状态时一定先假设一个初始状态
- 输出要求是次态时，要用触发器存储

2 Design Procedure of Synchronous Sequential Circuits

- From state diagram or state table to logic diagram
- Design method I
  - From next state table and transition table to karnaugh map of input, then excitation equation
- Design method II
  - From next state table and karnaugh map of states to state equation, then to excitation equation

Design method I

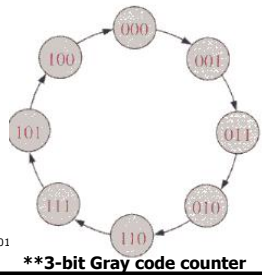
- From next state table and transition table to excitation equation

Design Procedure

- Step1: Specify the Sequence and Draw a State Diagram
- Step 2: Derive a Next-State Table from the State Diagram
- Step 3: Develop a Flip-Flop Transition Table showing the FF inputs required for each transition
- Step 4: Transfer the J and K states from the transition table to Karnaugh Maps.
- Step 5: Generate a Logic Expressions for FF inputs
- Step 6: Implementation the Expressions with Combinational Logic combining with FFs

Step1: Specify the Sequence and Draw a State Diagram

- *State Diagram*: a graph shows a sequence of states or values.



**3-bit Gray code counter

Step 2: Derive a Next-State Table from the State Diagram

- *Next-State Table*: shows the state that the circuit goes to from its present upon application of a clock pulse.

Present State Q_2	Q_1	Q_0	Next State Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Step 3: Develop a Flip-Flop Transition Table showing the FF inputs required for each transition

- *Flip-Flop Transition Table*: shows the all output transition of the FF goes from present states to next states.

Output Transitions Q_N		Q_{N+1}	Flip-Flop Inputs J	K
0	⟶	0	0	X
0	⟶	1	1	X
1	⟶	0	X	1
1	⟶	1	X	0

Step 4: Transfer the J and K states from the transition table to Karnaugh Maps.

- Using Karnaugh Maps to determine the logic required for the J and K inputs of each FF.
- Note: the value in the cell is the value of J or K according to the state of the cell

By Bao Qilian

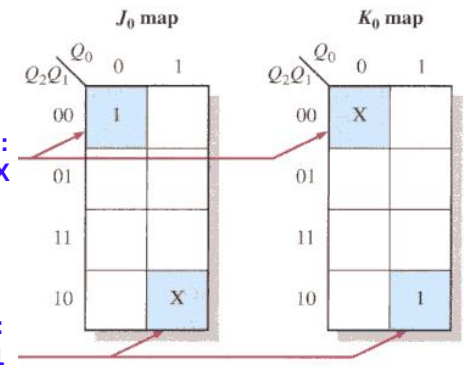
Present State Q_2	Q_1	Q_0	Next State Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Q_0

Output Transitions Q_N		Q_{N+1}	Flip-Flop Inputs J	K
0	⟶	0	0	X
0	⟶	1	1	X
1	⟶	0	X	1
1	⟶	1	X	0

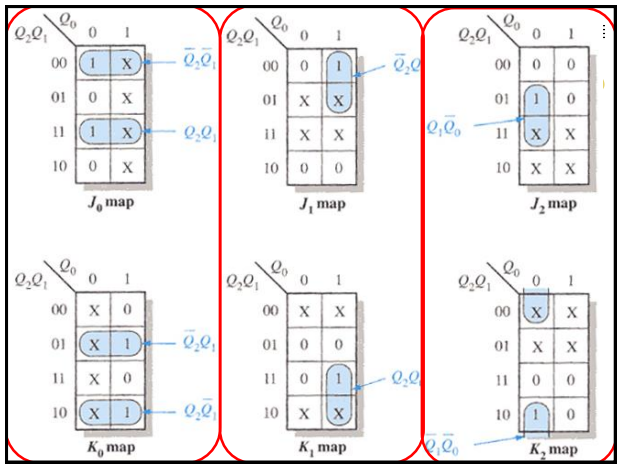
By Bao Qilian

Karnaugh Maps of Inputs of FF_0:



000→001: J_0=1,K_0=X

101→100: J_0=X,K_0=1

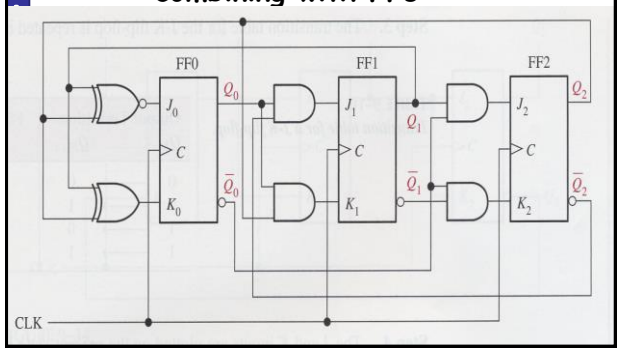


Step 5: Generate a Logic Expressions for FF inputs

From the minimization results of the Karnaugh maps, obtain the expressions:

$$J_0 = Q_2Q_1 + \overline{Q}_2\overline{Q}_1 = \overline{Q_2 \oplus Q_1}$$
$$K_0 = Q_2\overline{Q}_1 + \overline{Q}_2Q_1 = Q_2 \oplus Q_1$$
$$J_1 = \overline{Q}_2Q_0$$
$$K_1 = Q_2Q_0$$
$$J_2 = Q_1\overline{Q}_0$$
$$K_2 = \overline{Q}_1\overline{Q}_0$$

Step 6: Implementation the Expressions with Combinational Logic combining with FFs

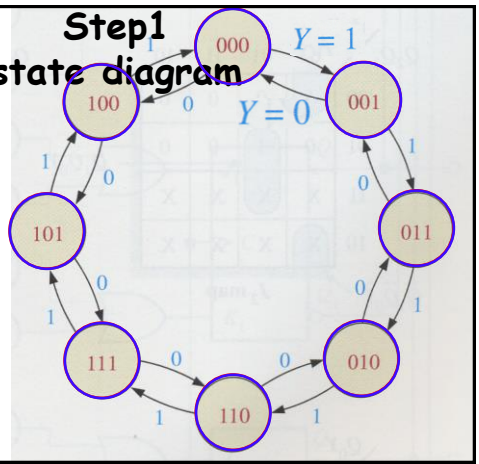


Example

Develop a synchronous 3-bit up/down counter with a Gray Code sequence. When up/down control input is 1, counter should count up. When the control input is 0. counter should count down.

By Bao Qilian

Step1 state diagram



Step2: Next-state table

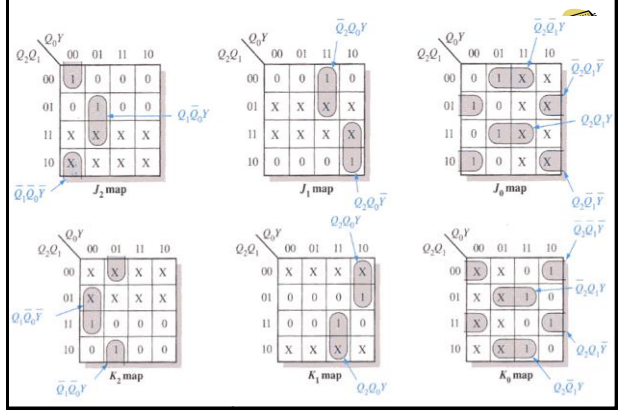
Present State			Next State Y = 0 (DOWN)			Next State Y = 1 (UP)		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	1
0	1	1	0	0	1	0	1	0
0	1	0	0	1	1	1	1	0
1	1	0	0	1	0	1	1	1
1	1	1	1	1	0	1	0	1
1	0	1	1	1	1	1	0	0
1	0	0	1	0	1	0	0	0

Y = UP/$\overline{\text{DOWN}}$ control input.

Stpe3: transition table for J-K FFs

Output Transitions Q_N		Q_{N+1}	Flip-Flop Inputs J	K
0	⟶	0	0	X
0	⟶	1	1	X
1	⟶	0	X	1
1	⟶	1	X	0

Step 4: Karnaugh Maps



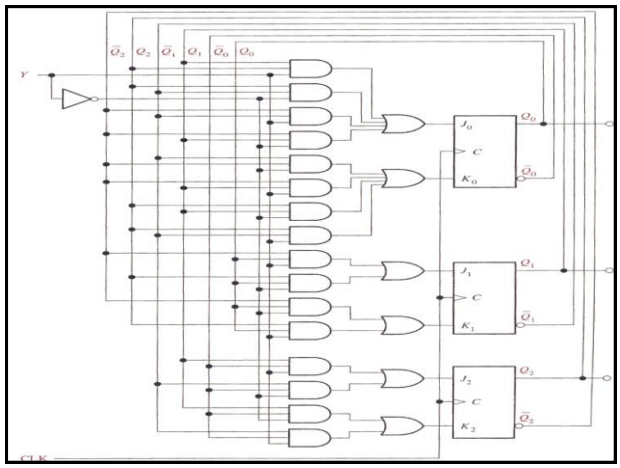
Step5: Logic expressions:

$$\begin{cases} J_0 = Q_2Q_1Y + Q_2\overline{Q}_1\overline{Y} + \overline{Q}_2Q_1\overline{Y} + \overline{Q}_2\overline{Q}_1Y \\ K_0 = \overline{Q}_2\overline{Q}_1\overline{Y} + \overline{Q}_2Q_1Y + Q_2Q_1\overline{Y} + Q_2\overline{Q}_1Y \end{cases}$$

$$\begin{cases} J_1 = \overline{Q}_2Q_0Y + Q_2Q_0\overline{Y} \\ K_1 = \overline{Q}_2Q_0\overline{Y} + Q_2Q_0Y \end{cases}$$

$$\begin{cases} J_2 = Q_1\overline{Q}_0Y + \overline{Q}_1\overline{Q}_0\overline{Y} \\ K_2 = Q_1\overline{Q}_0\overline{Y} + \overline{Q}_1\overline{Q}_0Y \end{cases}$$

Step 6: Implementation

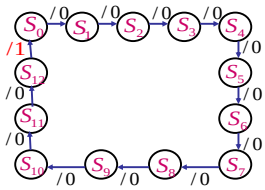


Design method II

- From state equation to excitation equation

Example: design a muduli-13 counter with carry out

Solution: (1)obtain state diagram



State diagram

State encoding:

0000 ~ 1100 : S_0~S_{12}

	code					
	Q_3	Q_2	Q_1	Q_0	C	
S_0	0	0	0	0	0	0
S_1	0	0	0	1	0	1
S_2	0	0	1	0	0	2
S_3	0	0	1	1	0	3
S_4	0	1	0	0	0	4
S_5	0	1	0	1	0	5
S_6	0	1	1	0	0	6
S_7	0	1	1	1	0	7
S_8	1	0	0	0	0	8
S_9	1	0	0	1	0	9
S_{10}	1	0	1	0	0	10
S_{11}	1	0	1	1	0	11
S_{12}	1	1	0	0	1	12
S_0	0	0	0	0	0	0

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	0001/0	0010/0	0100/0	0011/0
01	0101/0	0110/0	1000/0	0111/0
11	0000/1	××××/×	××××/×	××××/×
10	1001/0	1010/0	1100/0	1011/0

$(Q_3^{n+1}Q_2^{n+1}Q_1^{n+1}Q_0^{n+1}/C)$ 的卡诺图

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	×	×	×
10	1	1	1	1

Q_3^{n+1}

$$Q_3^{n+1} = Q_3\overline{Q}_2 + Q_2Q_1Q_0$$

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	0	×	×	×
10	0	0	1	0

Q_2^{n+1}

$$Q_2^{n+1} = \overline{Q}_3Q_2\overline{Q}_1 + \overline{Q}_3Q_2\overline{Q}_0 + \overline{Q}_2Q_1Q_0$$

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	0001/0	0010/0	0100/0	0011/0
01	0101/0	0110/0	1000/0	0111/0
11	0000/1	××××/×	××××/×	××××/×
10	1001/0	1010/0	1100/0	1011/0

$(Q_3^{n+1}Q_2^{n+1}Q_1^{n+1}Q_0^{n+1}/C)$ 的卡诺图

$$Q_1^{n+1} = \overline{Q}_1Q_0 + Q_1\overline{Q}_0$$

$$Q_0^{n+1} = \overline{Q}_3\overline{Q}_0 + \overline{Q}_2\overline{Q}_0$$

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	×	×	×
10	0	1	0	1

Q_1^{n+1}

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	0	×	×	×
10	1	0	0	1

Q_0^{n+1}

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	0001/0	0010/0	0100/0	0011/0
01	0101/0	0110/0	1000/0	0111/0
11	0000/1	××××/×	××××/×	××××/×
10	1001/0	1010/0	1100/0	1011/0

$(Q_3^{n+1}Q_2^{n+1}Q_1^{n+1}Q_0^{n+1}/C)$ 的卡诺图

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	×	×	×
10	0	0	0	0

C

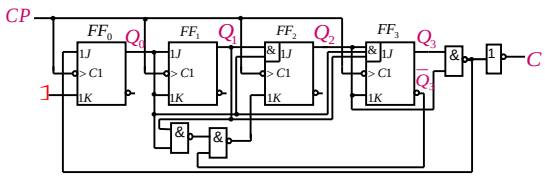
$$C = Q_3Q_2$$

$$\begin{cases} Q_3^{n+1} = Q_3\overline{Q}_2 + Q_2Q_1Q_0(Q_3+\overline{Q}_3) = (Q_2Q_1Q_0)\overline{Q}_3 + \overline{Q}_2Q_3 \\ Q_2^{n+1} = (Q_0Q_1)\overline{Q}_2 + (\overline{Q}_3 \bullet \overline{Q_1Q_0})Q_2 \\ Q_1^{n+1} = Q_0\overline{Q}_1 + \overline{Q}_0Q_1 \\ Q_0^{n+1} = (\overline{Q}_3+\overline{Q}_2)\overline{Q}_0 + \bar{1} \bullet Q_0 = (\overline{Q_3Q_2})\overline{Q}_0 + \bar{1}Q_0 \end{cases}$$

$$\begin{cases} J_3 = Q_2Q_1Q_0, & K_3 = Q_2 \\ J_2 = Q_1Q_0, & K_2 = \overline{Q_3Q_1Q_0} \\ J_1 = Q_0, & K_1 = Q_0 \\ J_0 = \overline{Q_3Q_2}, & K_0 = 1 \end{cases}$$

J=1,K=X,=> Q=1
∴ $J_3=Q_2Q_1Q_0$,K_3=X, Q_3=1
∴ K_3可以去掉$Q_2Q_1Q_0$这一项，简化为Q_2

$$\begin{cases} J_3 = Q_2Q_1Q_0, & K_3 = Q_2 \\ J_2 = Q_1Q_0, & K_2 = \overline{\overline{Q_3}\,\overline{Q_1Q_0}} \\ J_1 = Q_0, & K_1 = Q_0 \\ J_0 = \overline{Q_3Q_2}, & K_0 = 1 \end{cases}$$



Sequence detectors

Construct a circuit that will detect a specified serial input sequence. The detection of the required bit pattern can occur in a longer data string and the correct pattern can overlap with another pattern. When the input pattern has been detected, cause an output to be asserted HIGH.

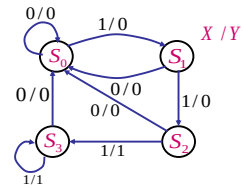
Example: code detector: find 111 in data sequence

Solution: input data = X

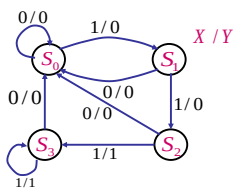
result of detection =Y

S^{n+1}/Y \ S^n, X	S_0	S_1	S_2	S_3
0	$S_0/0$	$S_0/0$	$S_0/0$	$S_0/0$
1	$S_1/0$	$S_2/0$	$S_3/1$	$S_3/1$

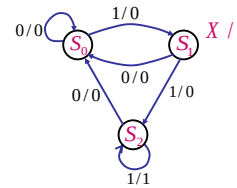
S^{n+1}/Y \ S^n, X	S_0	S_1	S_2	S_3
0	$S_0/0$	$S_0/0$	$S_0/0$	$S_0/0$
1	$S_1/0$	$S_2/0$	$S_3/1$	$S_3/1$



状态转换图



S_2、S_3 have the same output and next state when input is the same，so S_2=S_3



X \ $Q_1^nQ_0^n$	00	01	11	10
0	00/0	00/0	××/×	00/0
1	01/0	10/0	××/×	10/1

$(Q_1^{n+1}Q_0^{n+1}/Y)$

$Q_1^nQ_0^n$ / X	00	01	11	10
0	00/0	00/0	××/×	00/0
1	01/0	10/0	××/×	10/1

$(Q_1^{n+1}Q_0^{n+1}/Y)$

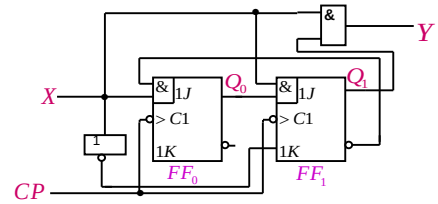
$$\begin{cases} Q_1^{n+1} = XQ_1 + XQ_0 = XQ_1 + XQ_0(Q_1 + \overline{Q}_1) \\ = (XQ_0)\overline{Q}_1 + XQ_1 \\ Q_0^{n+1} = X\overline{Q}_1\overline{Q}_0 = (X\overline{Q}_1)\overline{Q}_0 + 1\cdot Q_0 \end{cases}$$

$$\begin{cases} J_1 = XQ_0, & K_1 = \overline{X} \\ J_0 = X\overline{Q}_1, & K_0 = 1 \end{cases} \qquad \begin{cases} D_1 = XQ_1 + XQ_0 = X\overline{\overline{Q}_1\overline{Q}_0} \\ D_0 = X\overline{Q}_1\overline{Q}_0 \end{cases}$$

$Y = XQ_1$

$$\begin{cases} J_1 = XQ_0, & K_1 = \overline{X} \\ J_0 = X\overline{Q}_1, & K_0 = 1 \end{cases}$$

$Y = XQ_1$



同步时序电路的设计

- 设计：文字描述 → 状态图 → 逻辑图
- 同步时序电路的设计步骤：
  - 形成原始状态图和状态表
  - 状态化简
  - 求控制函数和输出函数
  - 画逻辑图

Summary

- Counter
  - Asynchronous counter
  - Synchronous counter
  - Up/Down synchronous counter
- Design of Counters
  - Design of Cascaded Counters
  - Design of Any Modulus Counters
  - Counter Decoding
- Design and Analysis of Sequence Circuit