Chapter 6
Registers and Counters

January 2006
Chapter 6 Registers and Counters

6-1 Registers
6-2 Shift Registers
6-3 Ripple Counters
6-4 Synchronous Counters
6-5 Other Counters
6-6 HDL for Registers and Counters
6-1 Registers

- Clocked sequential circuit
  - No flip-flops / no feedbacks → reduce to combinational circuit
  - No combinational circuit → remain a sequential circuit
    - registers and counters
- Register: a group of flip-flops capable of storing one bit of information
  - n-bit register consists of a group of n flip-flops capable of storing n bits
- Counter: a register going through a predetermined sequence of states
4-bit Register

- Simplest register: consisting of only flip-flops without any gates
- Example 6-1: 4-bit register
  - positive edge trigger
  - When the clear input goes to 0, all flip-flops are reset
  - The R inputs must be maintained at logic 1 during normal clocked operation
Register with Parallel Load

Parallel load
• loading: the transfer of new information into a register
• parallel loading: all the bits of the register are loaded simultaneously with a common clock pulse
  – Load control: determine when to load new information

Approaches to register with parallel load
1. controlling the clock input signal with an enabling gate:
   uneven propagation delays between the master clock and the inputs of flip-flops
2. controlling the D inputs: ensure that all clock pulses arrive at the same time anywhere in the system
4-bit register with a load control input

load=1
- data are transferred into the register with the next positive edge of the lock

load=0
- outputs are connected to their respective inputs

- The feedback connection is necessary because the D flip-flop does not have a “no change” condition
- The clock pulses are applied to the C inputs at all times

Fig. 6-2 4-Bit Register with Parallel Load
6-2 Shift Registers

• shift register: a register capable of shifting its binary information in one or both directions

• Example Fig. 6-3: each clock pulse shifts the contents of the register one bit position to the right
  – serial input: determines what goes into the leftmost flip-flop
  – serial output: taken from the output of the rightmost flip-flop

• **Shift control**: make the shift occur only with certain pulses
  – inhibiting the clock
  – control through the D inputs (shown later)

![4-Bit Shift Register](image)
**Serial Transfer**

Serial transfer: information is transferred one bit at a time by shifting the bits out of source register into destination register.

- The serial output (SO) of register A is connected to the serial input (SI) of register B and the SI of register A itself.
- The shift control input determines when and how many times the registers are shifted.

- **serial vs. parallel**

---

![Block diagram](image1)

(a) Block diagram

![Timing diagram](image2)

(b) Timing diagram

**Table 6-1 Serial-Transfer Example**

<table>
<thead>
<tr>
<th>Timing Pulse</th>
<th>Register A</th>
<th>Register B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial value</td>
<td>1 0 1 1</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>After $T_1$</td>
<td>1 1 0 1</td>
<td>1 0 0 1</td>
</tr>
<tr>
<td>After $T_2$</td>
<td>1 1 1 0</td>
<td>1 1 0 0</td>
</tr>
<tr>
<td>After $T_3$</td>
<td>0 1 1 1</td>
<td>0 1 1 0</td>
</tr>
<tr>
<td>After $T_4$</td>
<td>1 0 1 1</td>
<td>1 0 1 1</td>
</tr>
</tbody>
</table>

Fig. 6-4 Serial Transfer from Register A to register B
Serial Addition

Register A holds the augend and register B holds the addend
- Initially, register A and carry flip-flop are cleared to 0
All the numbers are transferred serially into B and added to A

- **parallel adder:** use registers with parallel load
  - # of full adders = # of bits
  - faster
  - combinational circuit

- **serial adder:** use shift registers
  - requiring less equipment
  - only one full adder
  - sequential circuit
Second Form of Serial Adder

Design a serial adder using a JK FF
• Assume 2 shift registers as input
• Obtain state table with FF input/outputs
• Obtain input and output equations
• Draw the circuit

---

**Table 6-2 State Table for Serial Adder**

<table>
<thead>
<tr>
<th>Present State</th>
<th>Inputs</th>
<th>Next State</th>
<th>Output</th>
<th>Flip-Flop Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>X</td>
<td>Y</td>
<td>Q</td>
<td>S</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q(t)</th>
<th>Q(t + 1)</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

---

**Fig. 6-6** Second form of Serial Adder

\[ J_Q = xy \]

\[ K_Q = x'y' = (x + y)' \]

\[ S = x \oplus y \oplus Q \]

no full-adder
Universal Shift Register

• Unidirectional shift register: capable of shifting in one direction only
• Bidirectional shift register: capable of shifting in both directions
• Universal shift register: has both shifts and parallel load capabilities
• The most general shift register has the following capabilities:

1. A clear control to clear the register to 0.
2. A clock input to synchronize the operations.
3. A shift-right control to enable the shift right operation and the serial input and output lines associated with the shift right.
4. A shift-left control to enable the shift left operation and the serial input and output lines associated with the shift left.
5. A parallel-load control to enable a parallel transfer and the $n$ input lines associated with the parallel transfer.
6. $n$ parallel output lines.
7. A control state that leaves the information in the register unchanged in the presence of the clock.
4-bit Universal Shift Register

Has all the capabilities listed above

Selection inputs control the mode of operation

Shift registers are often used to interface digital systems situated remotely from each other
6-3 Ripple Counters

• **counter**: a register that goes through a prescribed sequence of states upon the application of input pulses
  – may occur at a fixed interval of time or at random
  – may follow the binary number sequence or any other sequence of states
  – n-bit **binary counter**: n flip-flops counting in binary from 0~2^{n-1}

• Two categories
  – *Ripple counters*: FF output transition serves as a source for triggering other via the clock pin
    • Binary ripple counter
    • BCD ripple counter
  – *Synchronous counters*: inputs of all FF receive the common clock
    • discussed in Sections 6-4 and 6-5
Figure 6-8 4-Bit Binary Ripple Counter

- One single count input
- Output of each FF connected to C input of next higher-order FF
- Three approaches
  - from T
  - from JK: J and K inputs tied together
  - from D: complement output connected to the D input

Every time $A_i$ goes from 1 to 0, it complements $A_{i+1}$ (Negative trigger)

Binary count-down counter
- use positive-trigger T flip-flops instead

<table>
<thead>
<tr>
<th>$A_3$</th>
<th>$A_2$</th>
<th>$A_1$</th>
<th>$A_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

A3 A2 A1 A0
**BCD Ripple Counter**

- Decade decimal counter: 0 ~ 9
- Need at least 4 flip-flops, similar to a binary counter, but state after 1001 is 0000

<table>
<thead>
<tr>
<th>$Q_5$</th>
<th>$Q_4$</th>
<th>$Q_2$</th>
<th>$Q_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$Q_1$: count input  
$Q_2$: $Q_1$ negative-edge and $Q_8 = 0$  
$Q_4$: $Q_2$ negative-edge  
$Q_8$: $Q_1$ negative-edge and $Q_2 = Q_4 = 1$

Fig. 6-9 State Diagram of a Decimal BCD-Counter

Fig. 6-10 BCD Ripple Counter
Three-Decade Decimal BCD Counter

• n-decade counter: count from 0 to $10^n-1$
• Input to $n^{th}$ decades come from $Q_8$ of the previous $(n-1)^{th}$ decade
• When $Q_8$ in one decade goes from 1 to 0, it triggers the count for the next higher-order decade while its own decade goes from 9 to 0

Fig. 6-11 Block Diagram of a Three-Decade Decimal BCD Counter
6-4 Synchronous Counters

- synchronous counter: clock pulses are applied to inputs of all FF
- 3-bit binary counter with T flip-flops

\[ T_{A2} = A_1 A_2 \quad T_{A1} = A_0 \quad TA_0 = 1 \]
4-Bit Synchronous Binary Counter

- least significant position: complemented with every pulse
- any other positions: complemented if all lower significant bits are equal to 1

- It can be extended to any number of stages, with each stage having an addition FF and an AND gate that gives the output of 1 if all previous FF outputs are 1
- It can be triggered with either the positive or the negative clock edge
- It can be either of the JK-type, the T-type, or the D-type with XOR gates

<table>
<thead>
<tr>
<th>A_3</th>
<th>A_2</th>
<th>A_1</th>
<th>A_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig. 6-12 4-Bit Synchronous Binary Counter
Synchronous Count Down Binary Counter

- similar to 4-Bit synchronous count up binary counter
- least significant position: complemented with every pulse
- any other positions: complemented if all lower significant bits are equal to 0
Up-Down Binary Counter

<table>
<thead>
<tr>
<th>up</th>
<th>down</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x</td>
<td>count up</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>count down</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>no change</td>
</tr>
</tbody>
</table>

an up-down binary counter using T flip-flops
BCD Counter

count from 0000 to 1001 and back to 0000

minterms 10 to 15 are taken as don’t-care terms

Table 6-5 State Table for BCD Counter

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>Output</th>
<th>Flip-Flop Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>TQ₈</td>
</tr>
<tr>
<td>Q₈  Q₄  Q₂  Q₁</td>
<td>Q₈  Q₄  Q₂  Q₁</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>0  0  0  0</td>
<td>0  0  0  1</td>
<td>0</td>
<td>0  0  0  1</td>
</tr>
<tr>
<td>0  0  0  1</td>
<td>0  0  1  0</td>
<td>0</td>
<td>0  0  1  1</td>
</tr>
<tr>
<td>0  0  1  0</td>
<td>0  1  0  1</td>
<td>0</td>
<td>0  0  0  1</td>
</tr>
<tr>
<td>0  0  1  1</td>
<td>0  1  1  0</td>
<td>0</td>
<td>0  0  1  1</td>
</tr>
<tr>
<td>0  1  0  0</td>
<td>0  1  1  0</td>
<td>0</td>
<td>0  0  0  1</td>
</tr>
<tr>
<td>0  1  0  1</td>
<td>1  0  0  0</td>
<td>0</td>
<td>0  0  0  1</td>
</tr>
<tr>
<td>0  1  1  0</td>
<td>1  0  0  1</td>
<td>0</td>
<td>0  0  0  1</td>
</tr>
<tr>
<td>0  1  1  1</td>
<td>1  0  0  1</td>
<td>0</td>
<td>0  0  0  1</td>
</tr>
<tr>
<td>1  0  0  0</td>
<td>0  0  0  1</td>
<td>0</td>
<td>0  0  0  1</td>
</tr>
<tr>
<td>1  0  0  1</td>
<td>0  0  0  1</td>
<td>0</td>
<td>0  0  0  1</td>
</tr>
</tbody>
</table>

\[
T₂ = Q₈Q₁ \\
T₄ = Q₂Q₁ \\
T₈ = Q₈Q₁ + Q₄Q₂Q₁ \\
y = Q₈Q₁
\]

4 T flip-flops, 5 AND gates, and 1 OR gate
Binary Counter with Parallel Load

Load an initial binary number into the counter prior to the count operation

It can be used to generate any desired count sequence
A BCD Counter using a Binary Counter with Parallel Load

The AND detects the occurrence of state 1001 and then the counter reloads 0

The NAND detects the occurrence of state 1010 and then the counter is cleared to 0

Fig. 6-15 Two ways to Achieve a BCD Counter Using a Counter with Parallel Load
6-5 Other Counters

• Divide-by-N counter (modulo-N counter): a counter that goes through a repeated sequence of N states
• Counters can be used to generate timing signals to control the sequence of operations in a digital system
• Counters can be constructed also by means of shift registers
• The sequence of counters may follow the binary count or may be any other arbitrary sequence
• non-binary counters
  – Ring counter
  – Johnson counter
Counter with Unused States

- Outside interference may cause a circuit to enter one of the unused states
- Example:

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>Flip-Flop Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A  B  C</td>
<td>A  B  C</td>
<td>J_A  K_A  J_B  K_B  J_C  K_C</td>
</tr>
<tr>
<td>0  0  0</td>
<td>0  0  0</td>
<td>X  X  X  X  X  X</td>
</tr>
<tr>
<td>0  0  1</td>
<td>0  1  0</td>
<td>X  X  X  X  X  X</td>
</tr>
<tr>
<td>0  1  0</td>
<td>1  0  0</td>
<td>X  X  X  X  X  X</td>
</tr>
<tr>
<td>1  0  1</td>
<td>0  1  0</td>
<td>X  X  X  X  X  X</td>
</tr>
<tr>
<td>1  0  0</td>
<td>0  0  0</td>
<td>X  X  X  X  X  X</td>
</tr>
<tr>
<td>1  0  1</td>
<td>1  0  0</td>
<td>X  X  X  X  X  X</td>
</tr>
</tbody>
</table>

Two unused states: 011 and 111

Simplified equations:

- \( J_A = B \) \( K_A = B \)
- \( J_B = C \) \( K_B = 1 \)
- \( J_C = B' \) \( K_C = 1 \)

We need to analyze the circuit to determine the effects of unused states!

Self correcting counter: if it happens to be in an unused state, it eventually reaches the normal counter sequence after one or more clock pulses.


**Ring Counter**

- ring counter: a circuit shift register with only one flip-flop being set at any particular time, all others are cleared. The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals.
- Two approaches: (a) ring-counter (b) counter and decoder.
- k-bit ring counter: k flip-flops to provide k distinguishable states.

---

![Diagram of Ring Counter and Timing Signals](image)

---

Fig. 6-17 Generation of Timing Signals
Johnson Counter

**switch-tail ring counter**: a circular shift register with the complement output of the last flip-flop connected to the input of the first flip-flop – double the number of states for a ring counter (Figure 6-17a)

**Johnson counter**: a k-bit switch-tail counter with 2k decoding gates to provide outputs for 2k timing signals

Connecting Figure 6-18a with 8 AND gates listed Figure 6-18b to complete the construction of the Johnson counter

- Disadvantage: it never finds its way to a valid state if it is at an unused state
  - Correcting: \( D_C = (A+C)B \)
- \# of FF = \( \frac{1}{2} \) \# of timing signals
- \# of 2-input decoding gates = \# of time signals
Summary

Chapter 6 Registers and Counters

6-1 Registers
   4-bit register, register with parallel load

6-2 Shift Registers
   4-bit shift register, serial shift register, serial adder, second-form serial adder, universal shift register

6-3 Ripple Counters
   4-bit binary ripple counter, count-down counter, BCD ripple counter, multi-decade BCD counter

6-4 Synchronous Counters
   4-bit synchronous binary counter, count-down counter, up-down binary counter, BCD counter, binary counter with parallel load

6-5 Other Counters
   ring counter, Johnson counter