Logic and Computer Design Fundamentals

Chapter 5 – Registers and Register Transfers

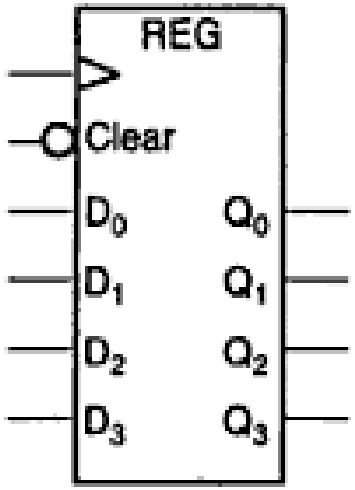
Charles Kime & Thomas Kaminski

Registers

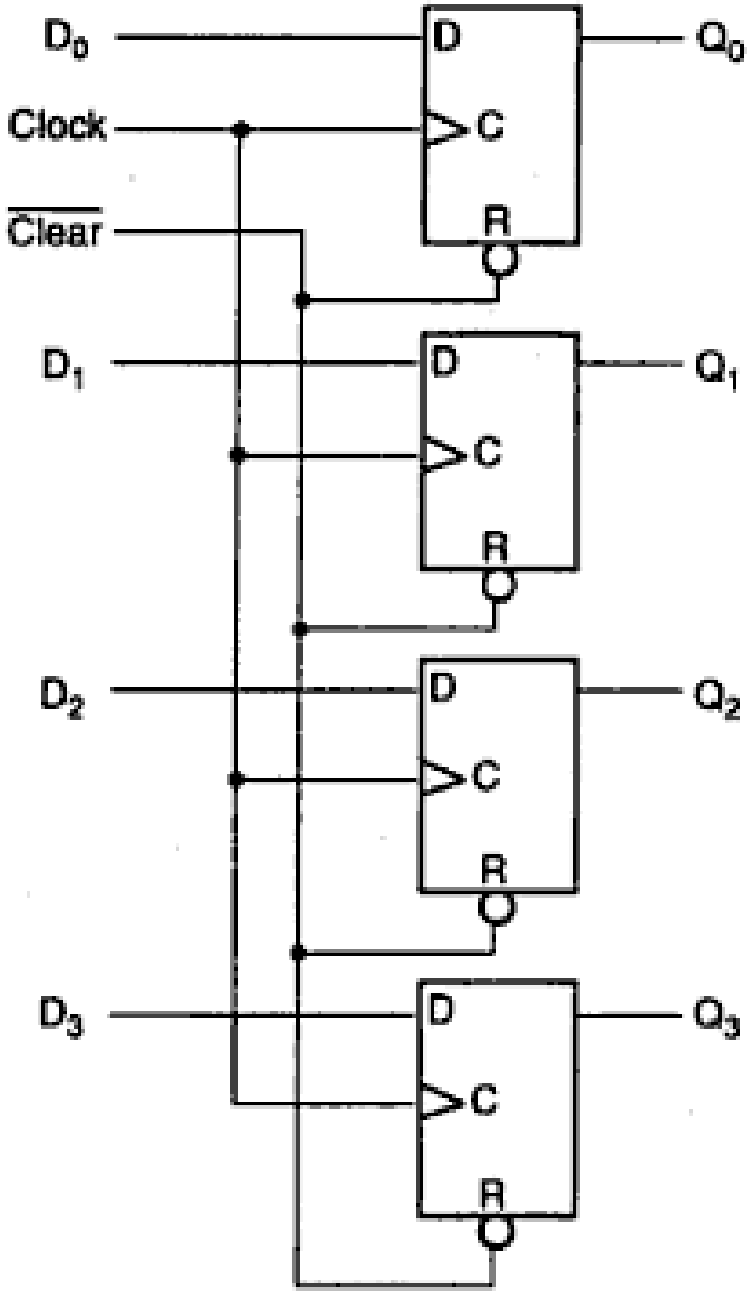
- Register – a collection of binary storage elements
- In theory, a register is sequential logic which can be defined by a state table
- More often think of a register as storing a vector of binary values
- Frequently used to perform simple data storage and data movement and processing operations

4-Bit Register

- $\overline{\text{Clear}}$ goes to $\overline{\text{R}}$
- We put '1' all the time
- Put '0' if want to clear the values



(b) Symbol



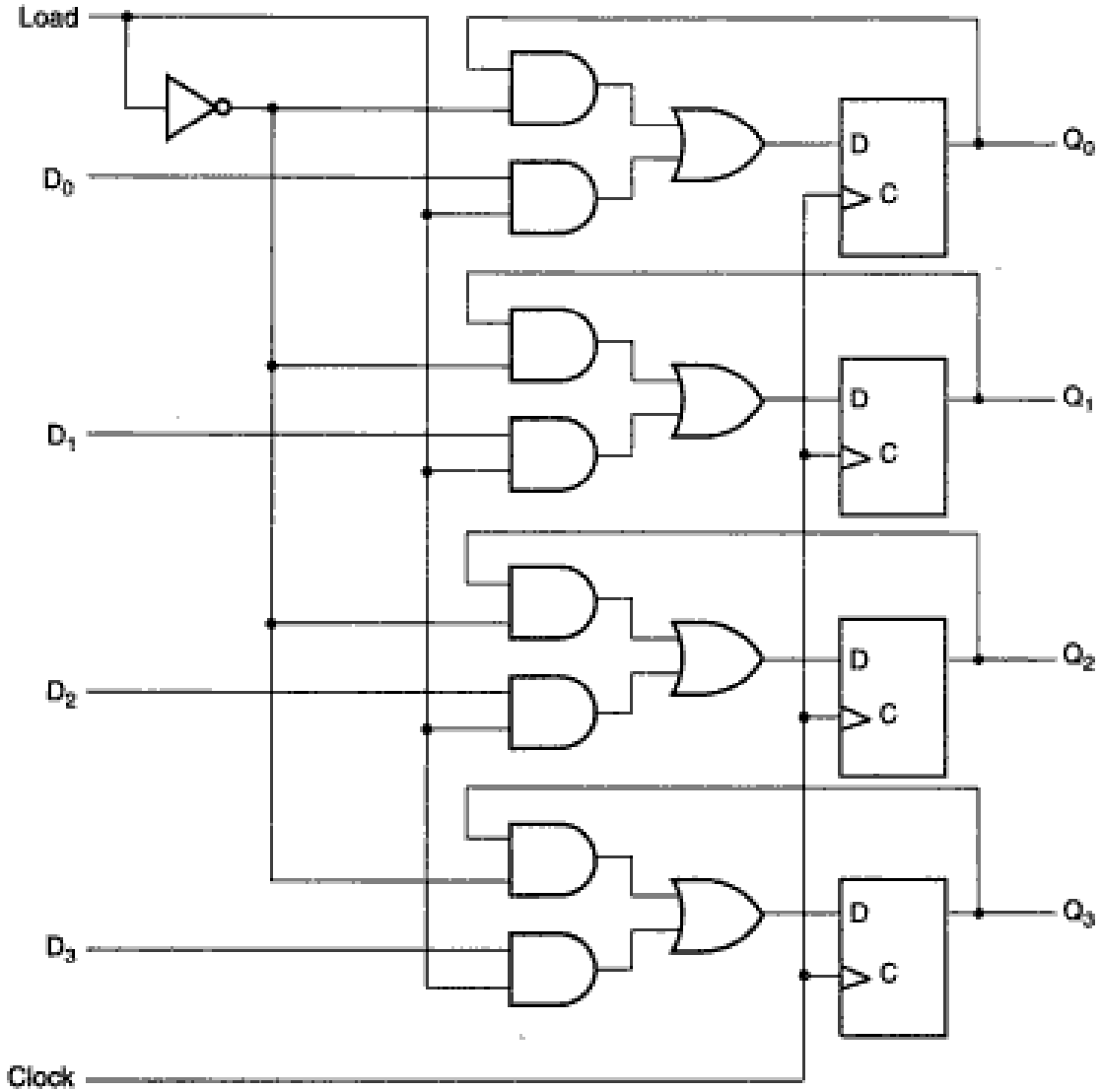
Register with Parallel Load

- If data is not changed no need to trigger the clock
- Clock inputs = C inputs

$$C\ inputs = \overline{Load} + Clock$$

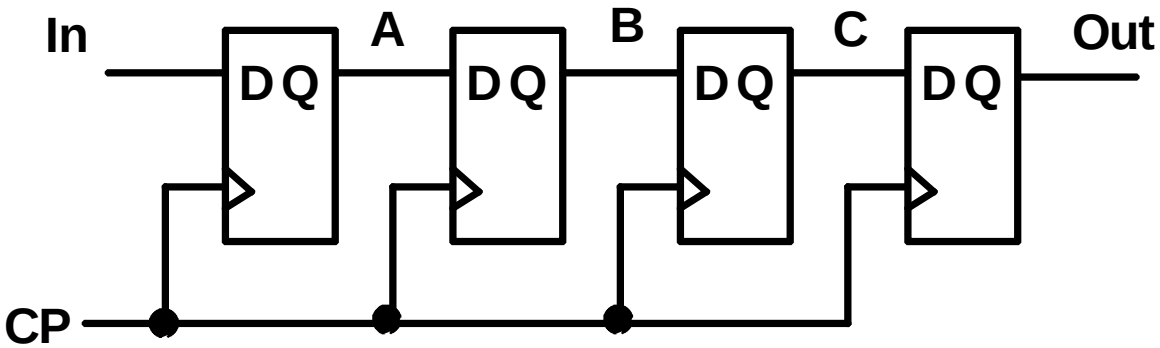
(c) Load control input

- When Load = 1, register clocked normally, new data is loaded
- Note, clock pulses arrive periodically, Load determines if new data would be loaded or not



Shift Registers

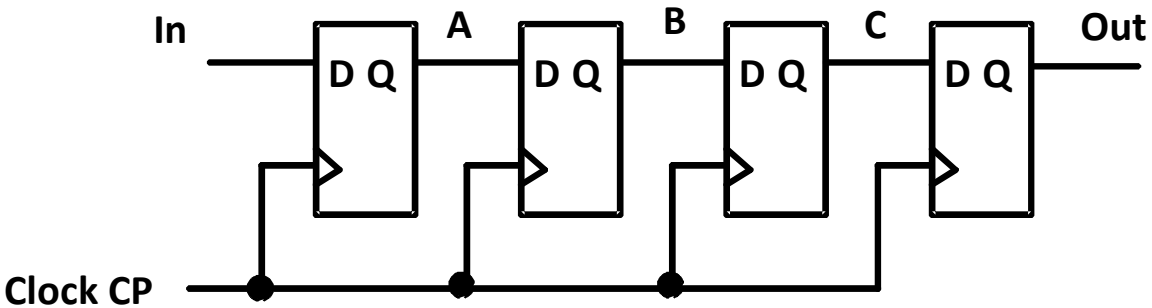
- **Shift Registers move data laterally within the register toward its MSB or LSB position**
- **In the simplest case, the shift register is simply a set of D flip-flops connected in a row like this:**



- **Data input, In, is called a *serial input* or the *shift right input.***
- **Data output, Out, is often called the *serial output.***
- **The vector (A, B, C, Out) is called the *parallel output.***

Shift Registers (continued)

- The behavior of the serial shift register is given in the listing on the lower right
- T0 is the register state just before the first clock pulse occurs
- T1 is after the first pulse and before the second.
- Initially unknown states are denoted by "?"
- Complete the last three rows of the table



CP	In	A	B	C	Out
T0	0	?	?	?	?
T1	1	0	?	?	?
T2	1	1	0	?	?
T3	0	1	1	0	?
T4	1				
T5	1				
T6	1				

Parallel Load Shift Registers

- By adding a mux between each shift register stage, data can be shifted or loaded
- If SHIFT is low, A and B are replaced by the data on D_A and D_B lines, else data shifts right on each clock.
- By adding more bits, we can make *n*-bit parallel load shift registers.
- A parallel load shift register with an added "hold" operation that stores data unchanged is given in Figure 7-10 of the text.