Shift Registers

Overview

- Multiple flip flops can be combined to form a data register
- Shift registers allow data to be transported one bit at a time
- Registers also allow for parallel transfer
  - Many bits transferred at the same time
- Shift registers can be used with adders to build arithmetic units
- Remember: most digital hardware can be built from combinational logic (and, or, invert) and flip flops
  - Basic components of most computers

Register with Parallel Load

- Register: Group of Flip-Flops
- Ex: D Flip-Flops
- Holds a Word (Nibble) of Data
- Loads in Parallel on Clock Transition
- Asynchronous Clear (Reset)

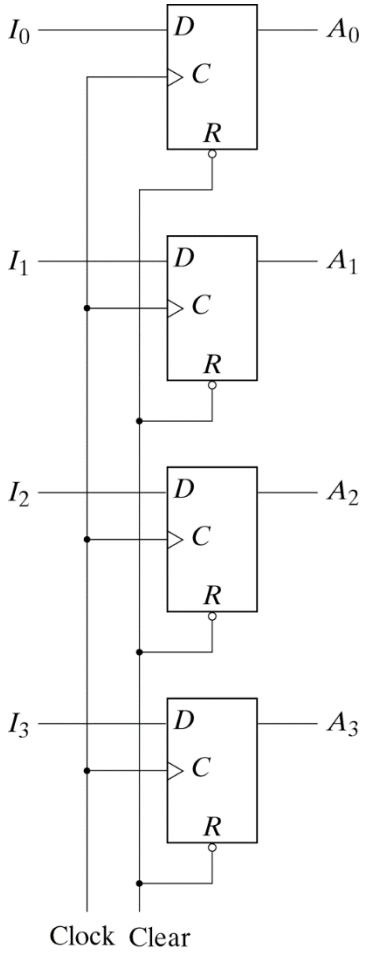


Fig. 6-1 4-Bit Register

Register with Load Control

- **Load Control = 1**
  - **New data loaded on next positive clock edge**
- **Load Control = 0**
  - **Old data reloaded on next positive clock edge**

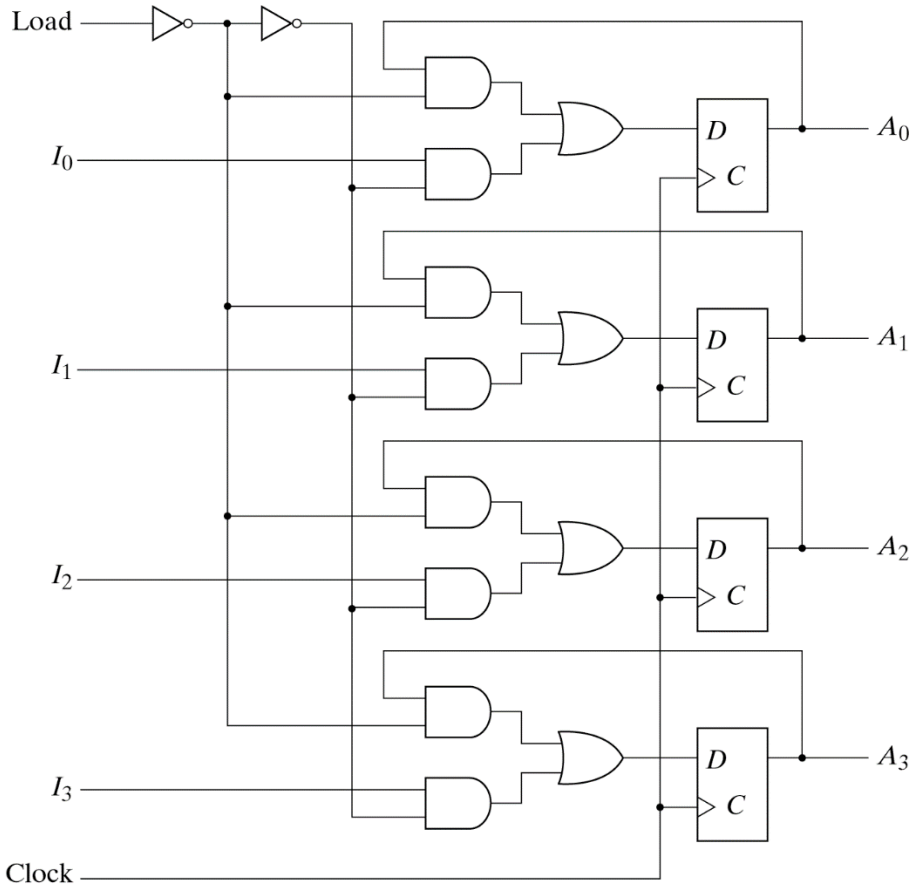


Fig. 6-2 4-Bit Register with Parallel Load

Shift Registers

- Cascade chain of Flip-Flops
- Bits travel on Clock edges
- Serial in – Serial out, can also have parallel load / read

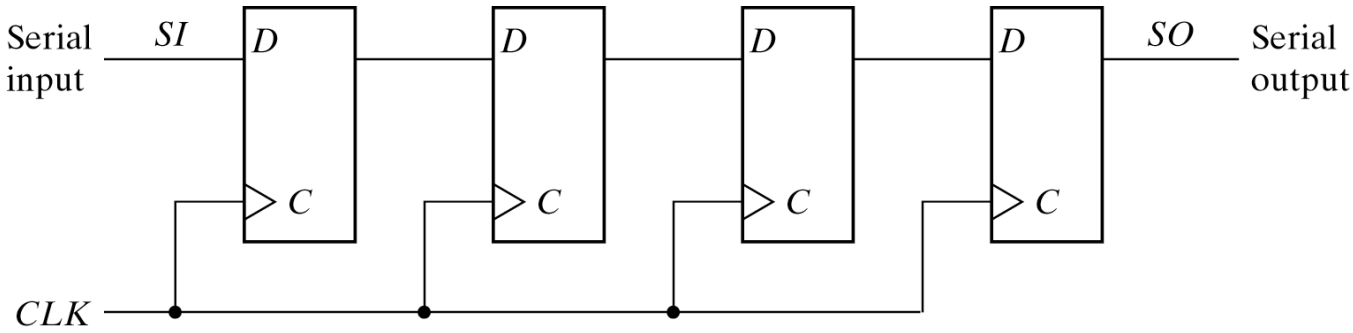
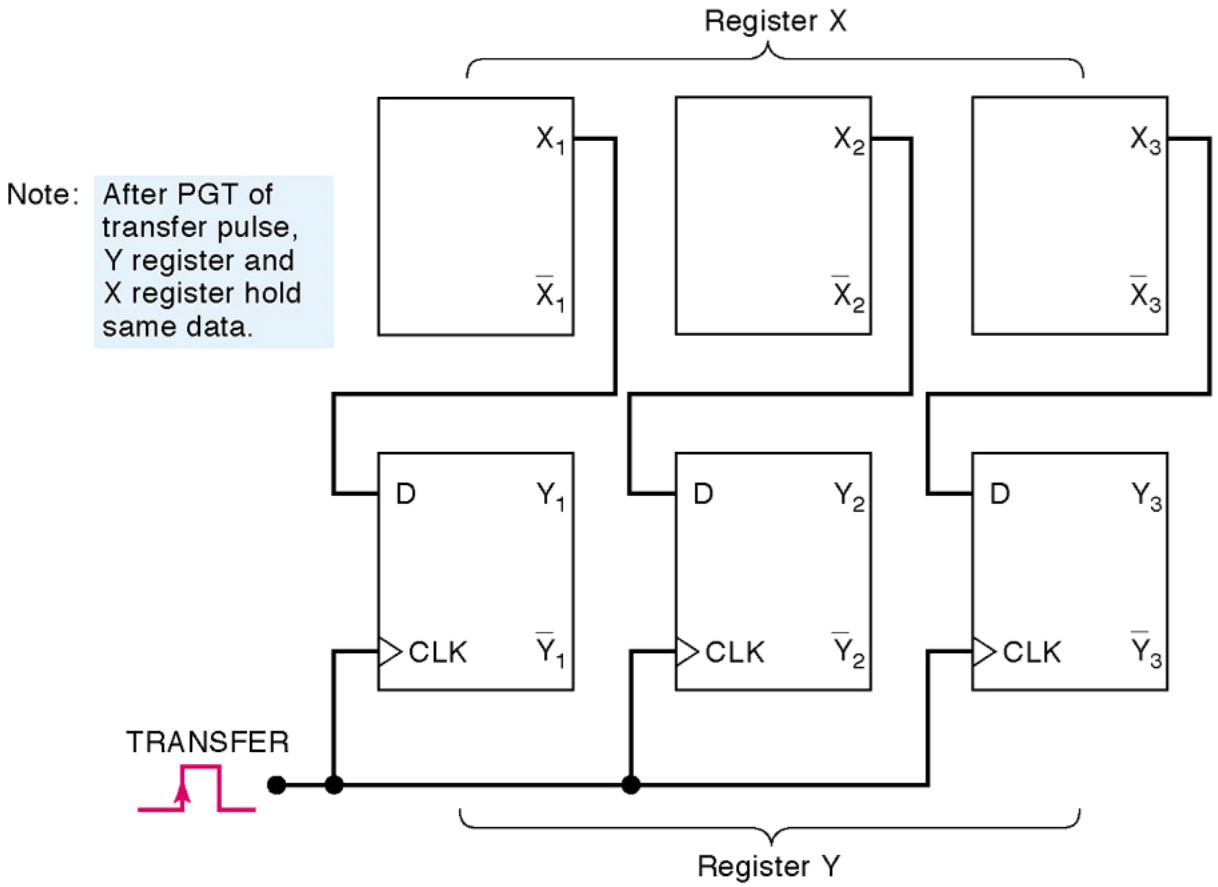


Fig. 6-3 4-Bit Shift Register

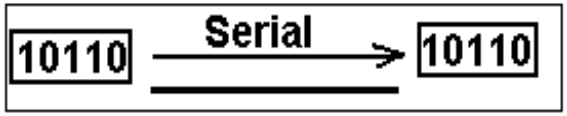
Parallel Data Transfer

- All data transfers on rising clock edge
- Data clocked into register Y

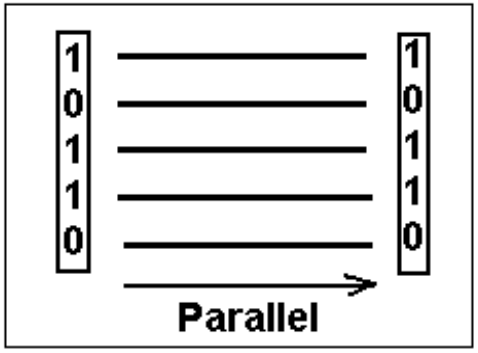


Parallel versus Serial

- **Serial communications is defined as**
  - **Provides a binary number as a sequence of binary digits, one after another, through one data line.**

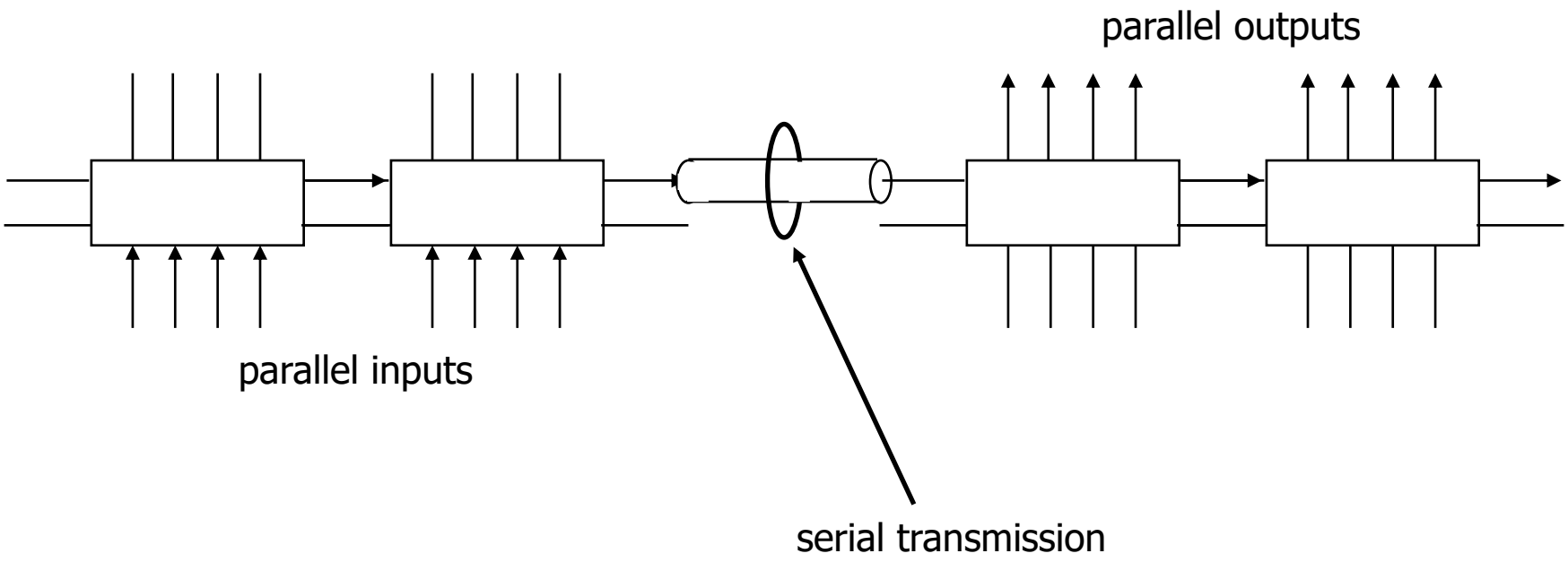


- **Parallel communications**
  - **Provides a binary number through multiple data lines at the same time.**



Shift register application

- **Parallel-to-serial conversion for serial transmission**



Serial Transfer

- Data transfer one bit at a time
- Data loopback for register A

Time	Reg A	Reg B
T0	1011	0011
T1	1101	1001
T2	1110	1100
T3	0111	0110
T4	1011	1011

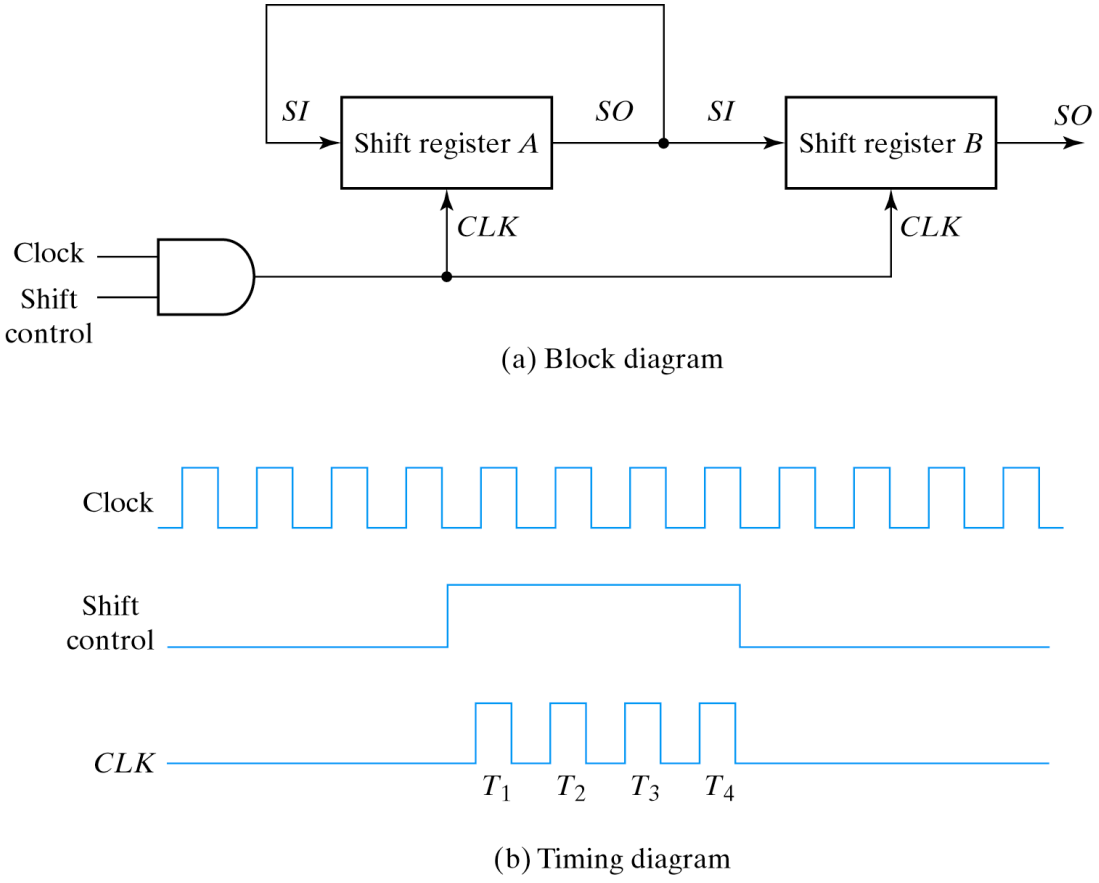
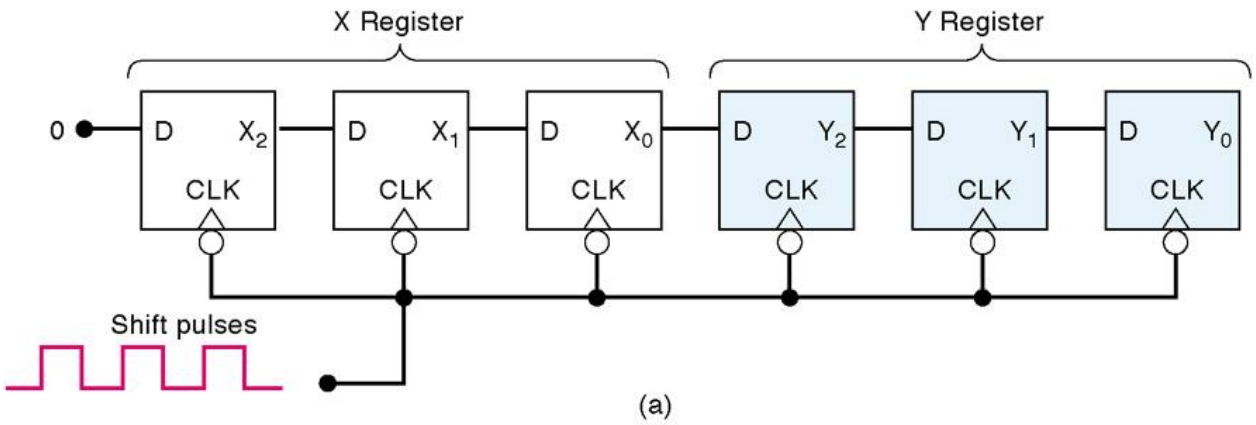


Fig. 6-4 Serial Transfer from Register *A* to register *B*

Serial Transfer of Data

- **Transfer from register X to register Y (negative clock edges for this example)**



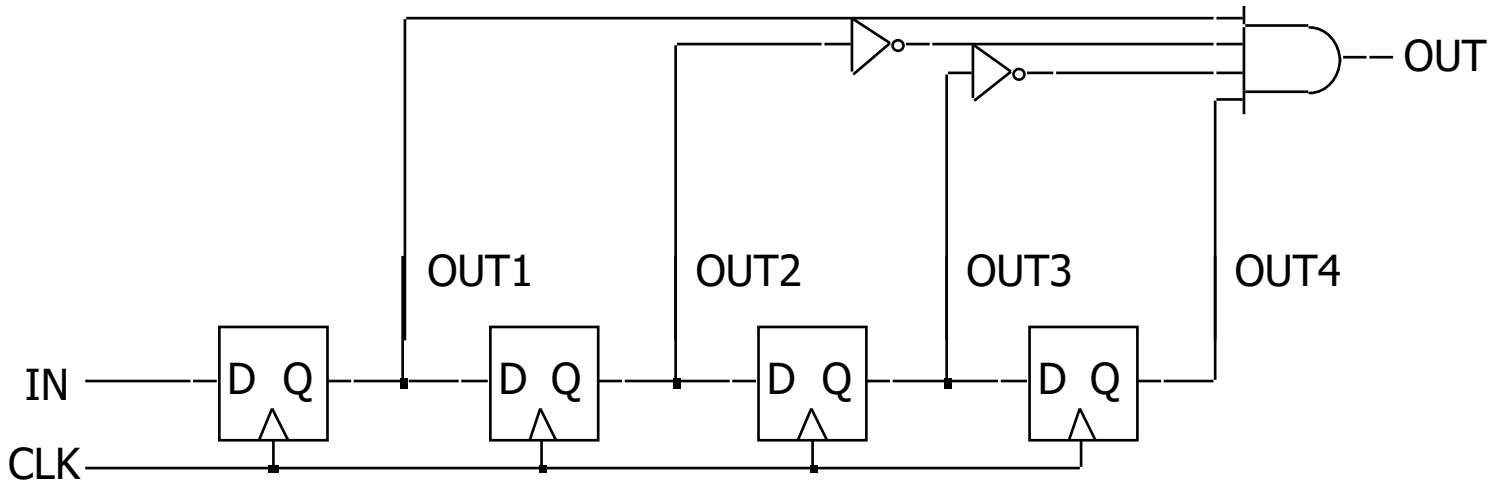
(a)

X_2	X_1	X_0	Y_2	Y_1	Y_0	
1	0	1	0	0	0	← Before pulses applied
0	1	0	1	0	0	← After first pulse
0	0	1	0	1	0	← After second pulse
0	0	0	1	0	1	← After third pulse

(b)

Pattern recognizer

- **Combinational function of input samples**
  - **in this case, recognizing the pattern 1001 on the single input signal**



Clk Before	IN	OUT1	OUT2	OUT3	OUT4	OUT
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	1	0	0	1	0	0
5	0	1	0	0	1	1

Serial Addition (D Flip-Flop)

- Slower than parallel
- Low cost
- Share fast hardware on slow data
- Good for multiplexed data

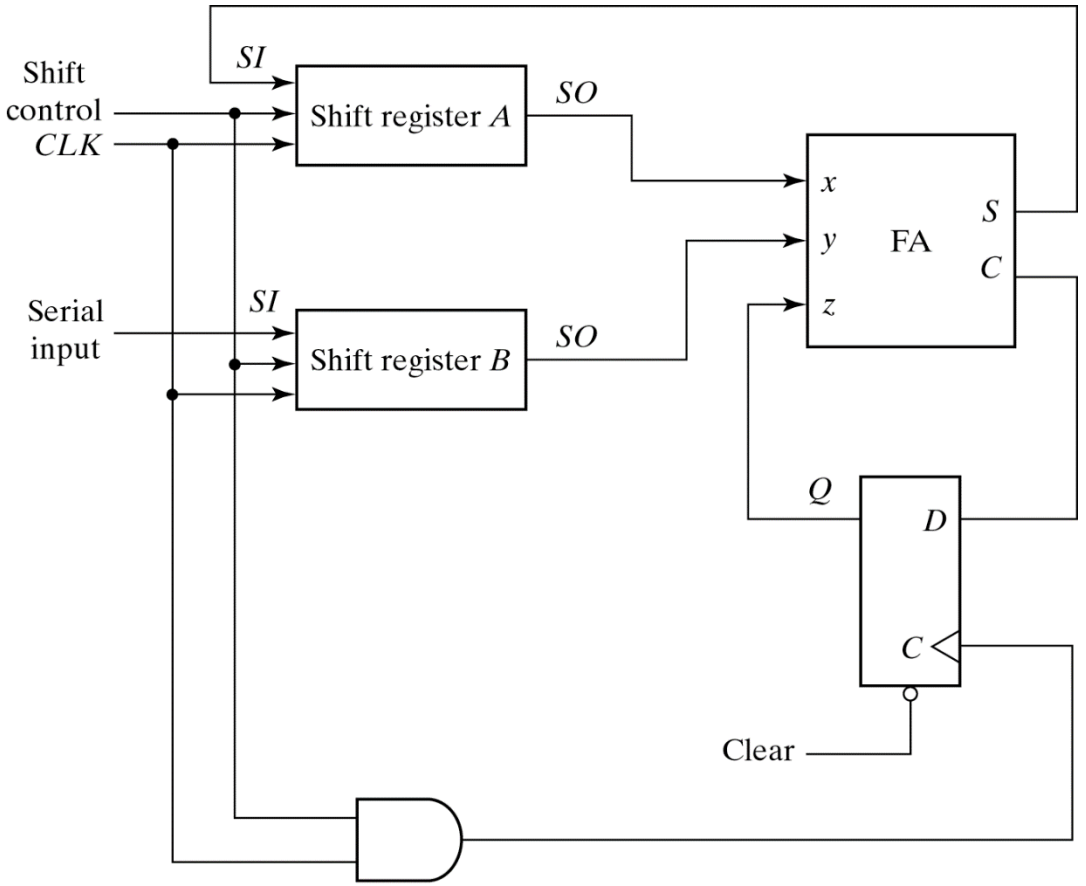


Fig. 6-5 Serial Adder

Serial Addition (D Flip-Flop)

- Only one full adder
- Reused for each bit
- Start with low-order bit addition
- Note that carry (Q) is saved
- Add multiple values.
  - New values placed in shift register B

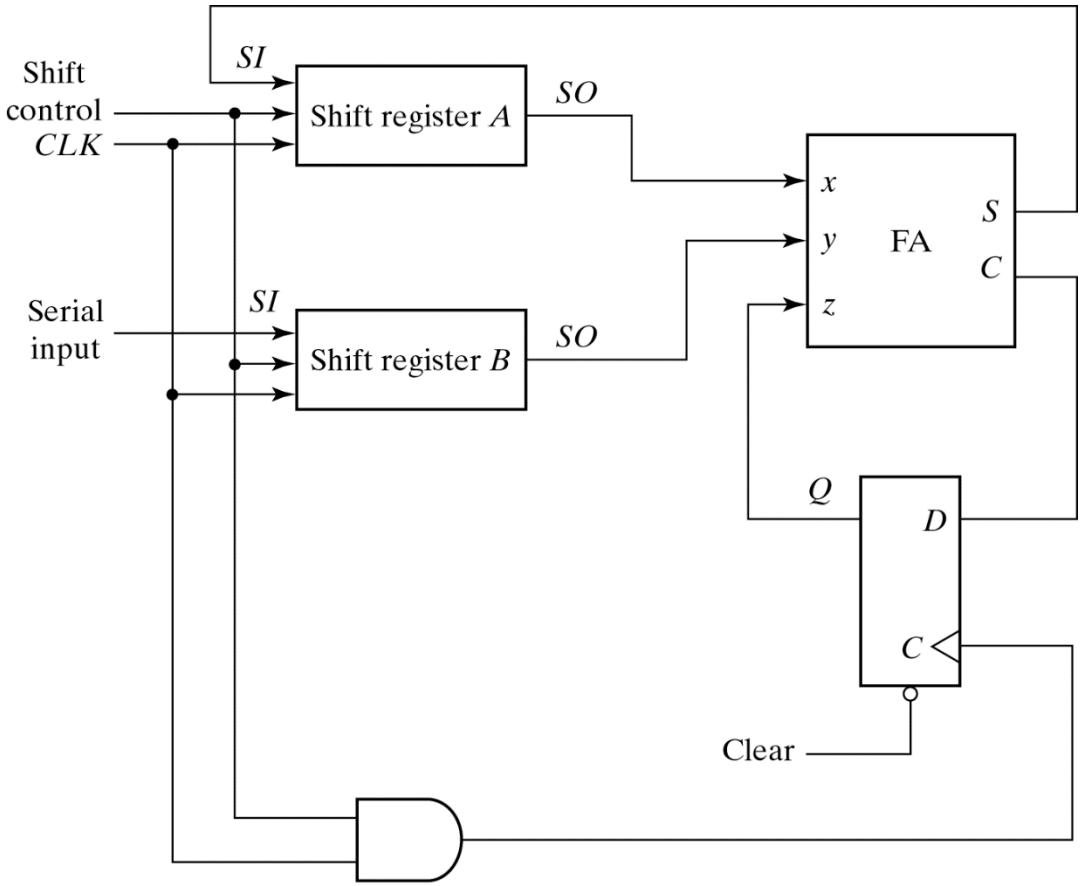


Fig. 6-5 Serial Adder

Serial Addition (D Flip-Flop)

- **Shift control** used to stop addition
- Generally not a good idea to **gate** the clock
- Shift register can be arbitrary length
- **FA** can be built from combin. logic

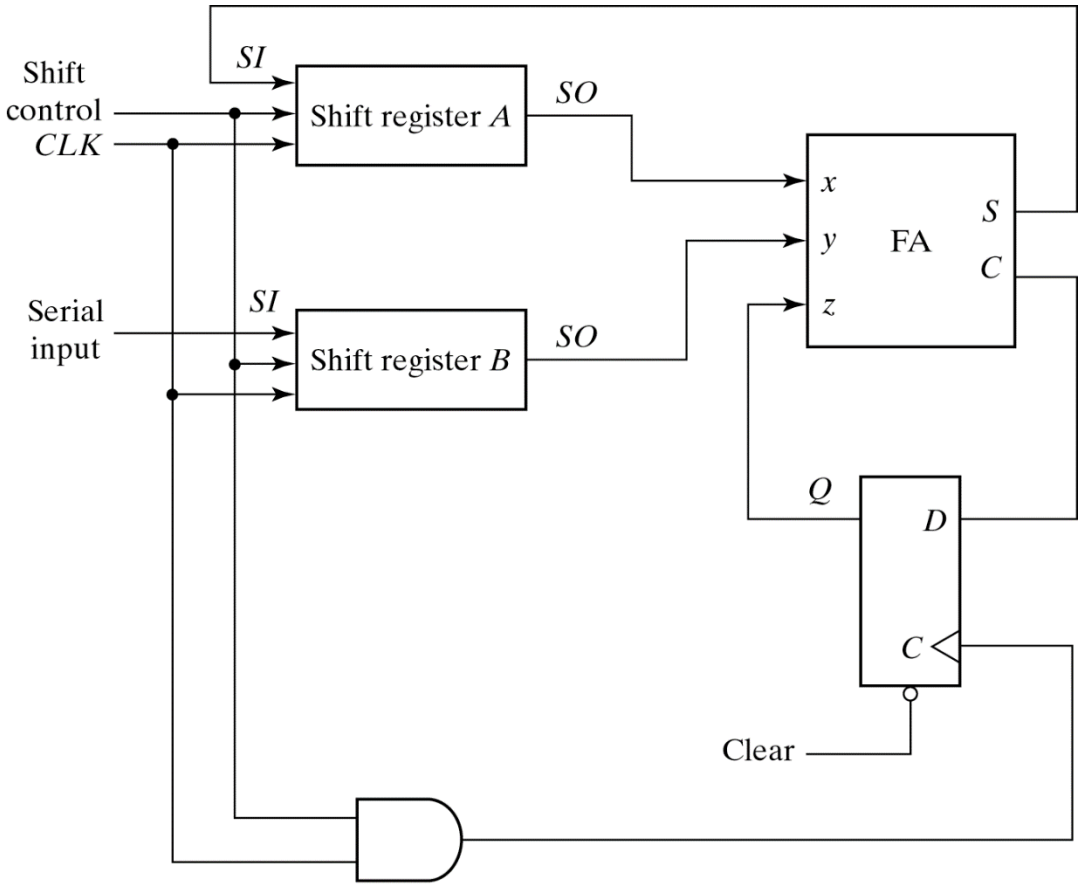


Fig. 6-5 Serial Adder

Universal Shift Register

- Clear
- Clock
- Shift
  - Right
  - Left
- Load
- Read
- Control

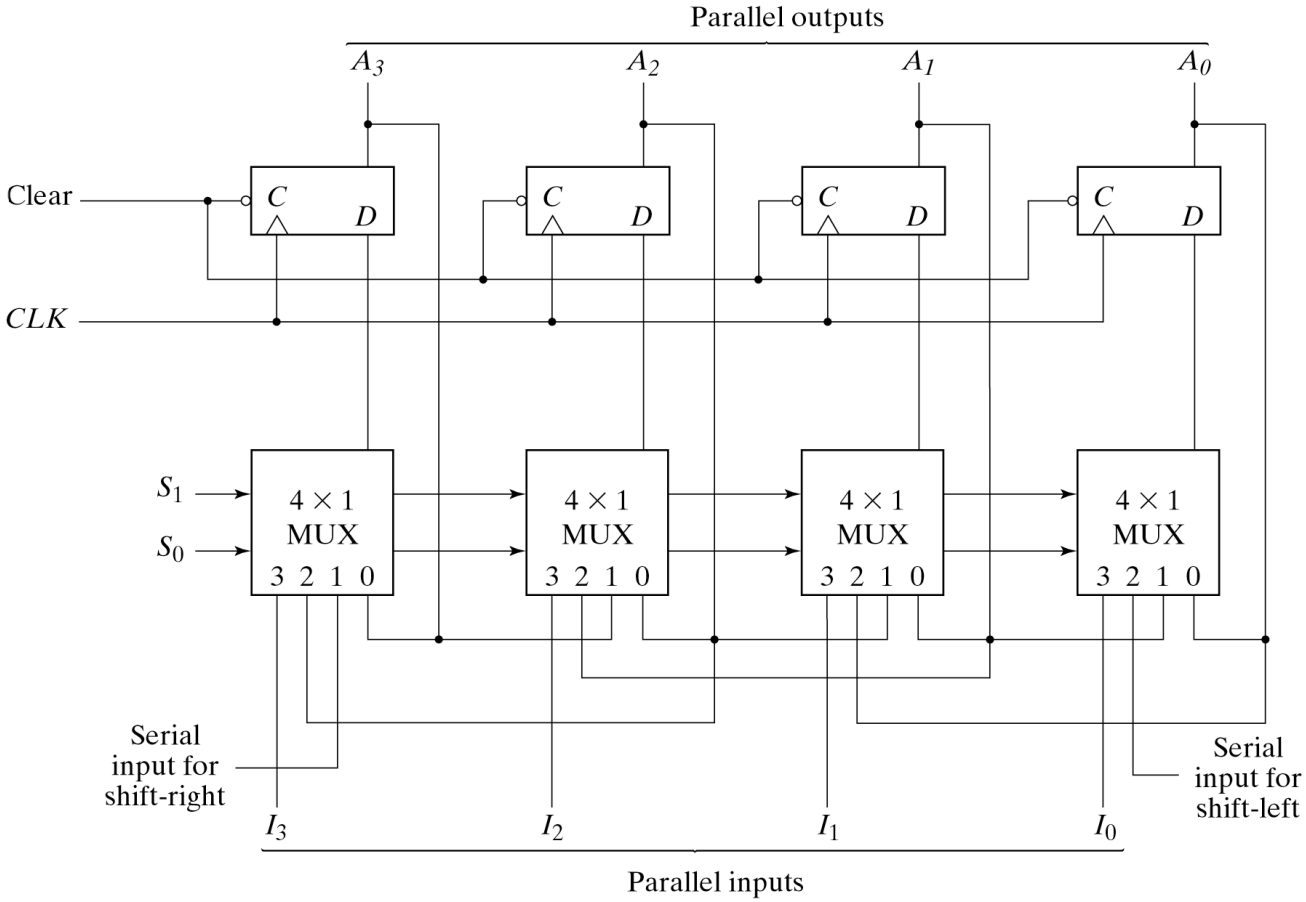


Fig. 6-7 4-Bit Universal Shift Register

Summary

- Shift registers can be combined together to allow for data transfer
- Serial transfer used in modems and computer peripherals (e.g. mouse)
- D flip flops allow for a simple design
  - Data clocked in during clock transition (rising or falling edge)
- Serial addition takes less chip area but is slow
- Universal shift register allows for many operations
  - The register is programmable.
  - It allows for different operations at different times
- Next time: counters (circuits that count!)