Shift Registers

Announcements

- Homework #8 due today
- Homework #9 posted

RIT Resilience

Success depends heavily on your personal health and well-being. **Recognize** that stress is an expected part of the college experience, and it can often be compounded by unexpected setbacks or life changes outside the classroom. I strongly encourage you to **reframe** challenges as opportunities for growth. **Reflect** on your role in taking care of yourself throughout the term, before the demands of exams and projects reach their peak. Please feel free to **reach out** to me about any difficulty you may be having that may impact your performance as soon as it occurs and before it becomes unmanageable. In addition to your academic advisor, you are strongly encouraged to contact a number of other support services on campus that stand ready to assist you.

https://www.rit.edu/studentaffairs/counseling/

Register Review

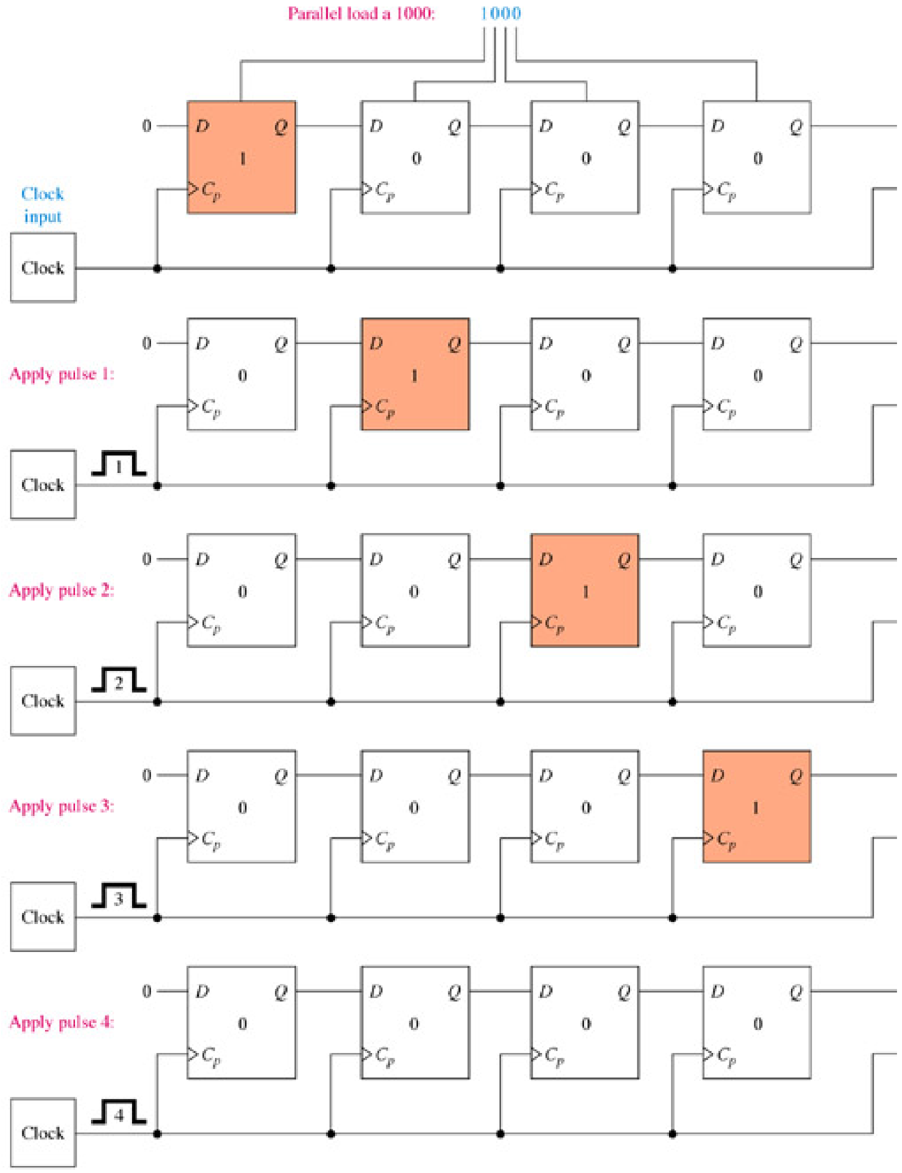
- What is a register?
  - A register is a group of flip flops all tied to the same clock that temporarily stores data as a word
  - Within a digital system, these words moves from register to register
  - The size of the word is based on the system, but is usually 8, 16, 32, or 64 bits
  - A register can only load an entire word at once
    - No mechanism to just load individual bits
    - Called a parallel load

How is a Shift Register Different?

- It’s ability to shift its data
  - A shift register can function as a regular register
  - Additionally it can shift data right and/or left
    - Ex: 1011
      - Shift right: 0101 – each bit moves 1 position to the right, the lsb is lost and 0 shifts into the msb
      - Shift left: 0110 – each bit moves 1 position to the left, the msb is lost and 0 shifts into the lsb
  - The shift is accomplished by connecting the Q from each flip-flop to the D of the flip flop next to it
    - On the clock edge the data moves

Basic Operation

- Initially the register contains 1000
- data shifts 1 bit to the right on each clock

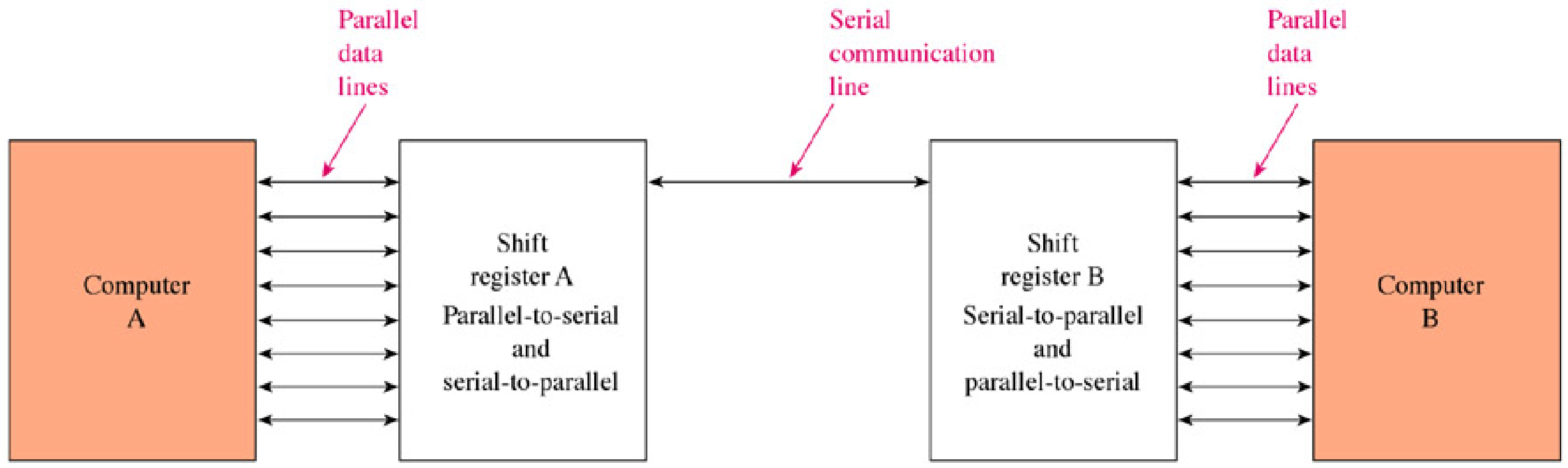


Serial Data

- Serial transmission
  - Data transmitted one bit at a time
  - Usually used when data is transferred from one device to another
    - Far less physical connections necessary
    - USB is an example
- But ….
  - We know that data is stored in a parallel format in the system
  - How does data that is stored in one format get transmitted in another?

Parallel / Serial Conversion

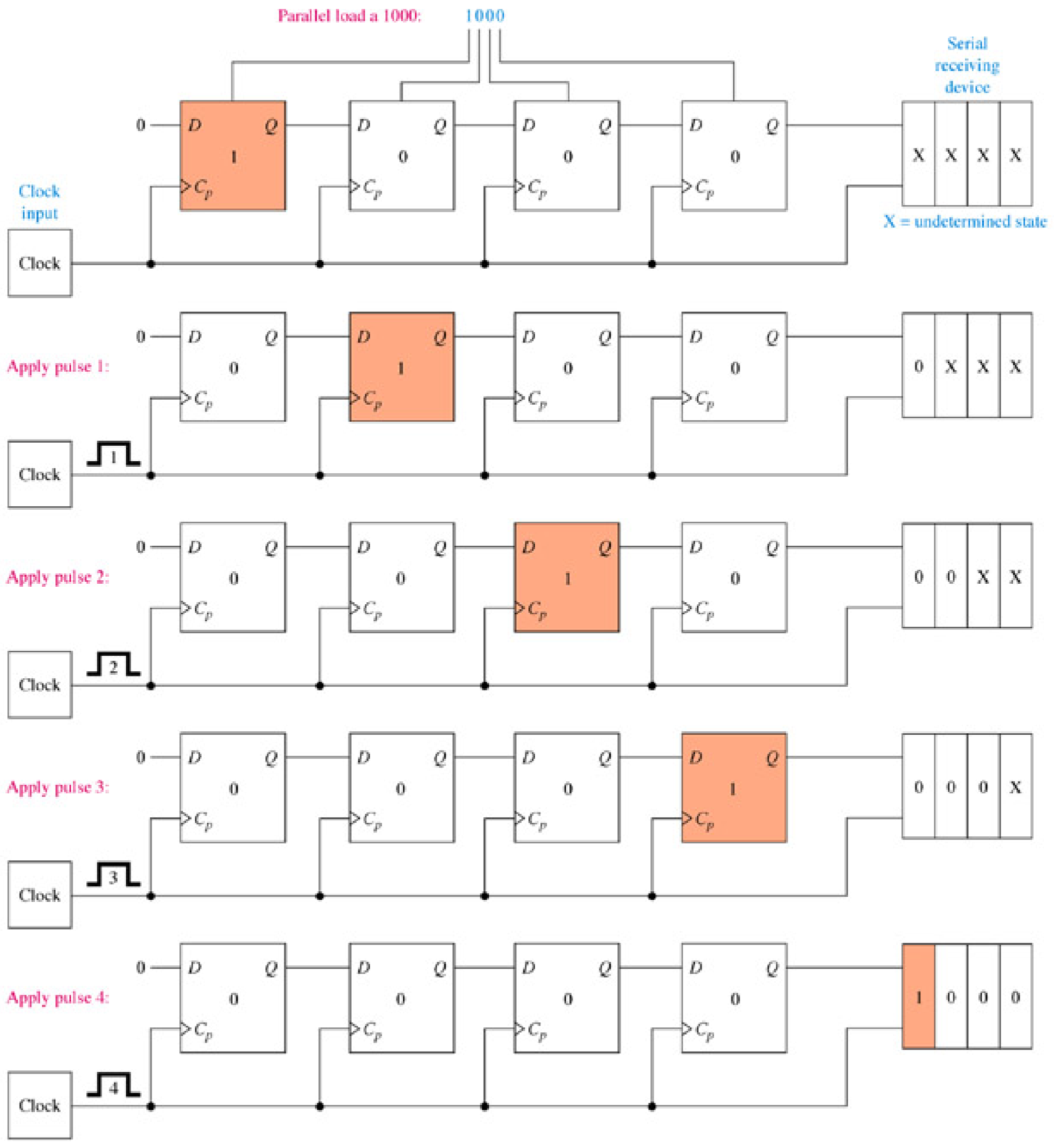
- Data leaving or entering a digital system on a serial interface has to be converted


Parallel data lines
Serial communication line
Parallel data lines
Computer A
Shift register A Parallel-to-serial and serial-to-parallel
Shift register B Serial-to-parallel and parallel-to-serial
Computer B

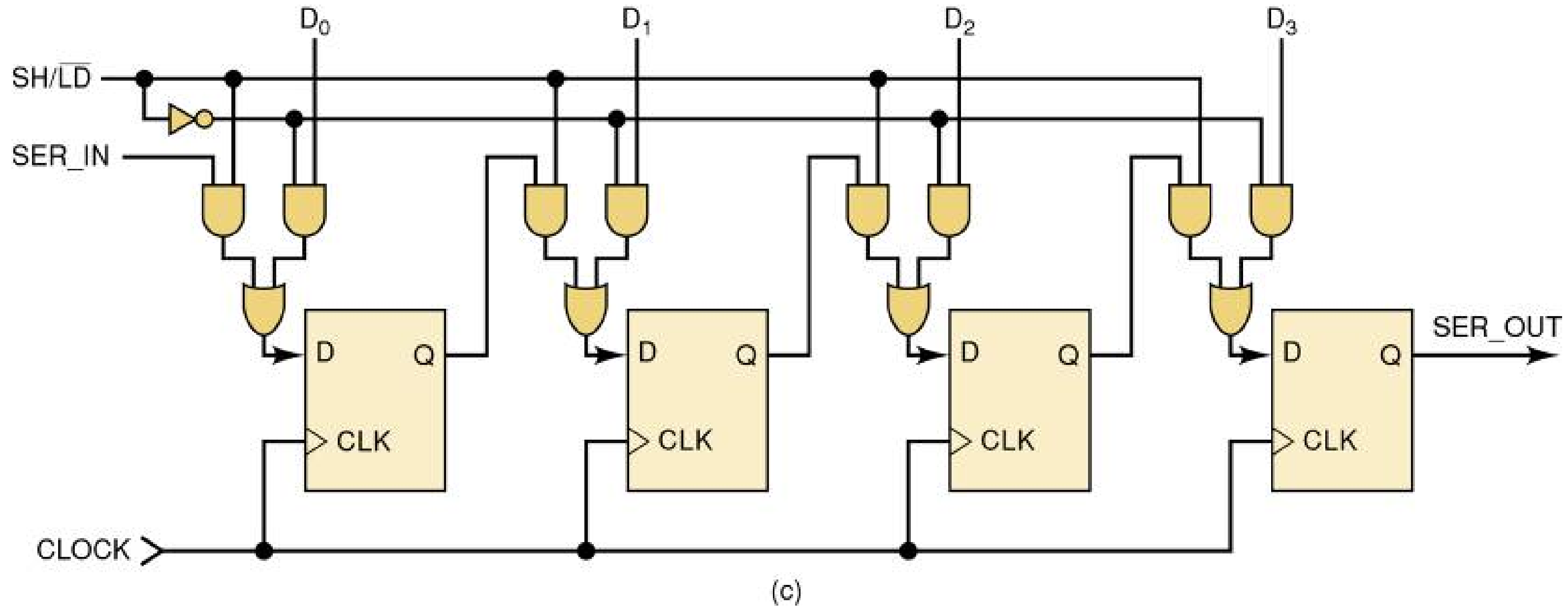
- Conversion is accomplished with a shift register

Parallel –to- Serial Conversion

- Returning to this example, take notice of the data shifted out of the last Q.
- It takes 4 clock cycles for a 4 bit number to transmitted serially



Parallel / Serial conversion



- Evaluate the logic
  - What happens when the SH/$\overline{\text{LD}}$ (Shift or Load) line is low?
  - What happens when it is high

Parallel / Serial Conversion

- 8-bit example
  - Parallel-to-serial
    - Activate load input
    - 8-bit word is loaded into the shift register from internal bus on clock edge
    - Activate shift input
    - Start clocking data. 8 clock cycles later, entire register is shifted out
  - Serial-to-parallel
    - Activate shift input
    - It takes 8 clock cycles for the serial_in data to fill the shift register
    - Now Q outputs can be read as parallel word

Shift Register VHDL

```
entity shift_reg is
port(
       clk : in std_logic;
        reset_n :  in std_logic;
        serial_in :  in std_logic;
        Load :  in std_logic;
        Parallel_in :  in std_logic_vector(7 downto 0);
        Parallel_out : out std_logic_vector(7 downto 0)

       );
end shift_reg;
```

```
architecture rtl of shift_reg is
      signal shift : std_logic_vector(7 downto 0);
Begin

shifter: process(clk,reset_n)
begin
  if (reset_n = '0') then
      shift <= (others => '0');
  elsif (clk'event and clk = '1') then
       if  (load = '1') then
              shift <= parallel_in;
       else
               shift(7 downto 1) <= shift(6 downto 0);
               shift(0) <= serial_in;
       end if;    --load
   end if;       --clk
end process;
Parallel_out <= shift;
End rtl;
```

Notice How a Vector is shifted

Shift Register VHDL

- Edit the previous VHDL to include a serial output

Shift Register

- Which direction is this shifting?
  - shift(0) <= serial_in;
    shift(7 downto 1) <= shift(6 downto 0);

- How would you write the VHDL to shift in the opposite direction?