HACETTEPE UNIVERSITY
Computer Engineering Department
BBM233 Logic Design Laboratory
Fall 2019

Experiment 6
Registers

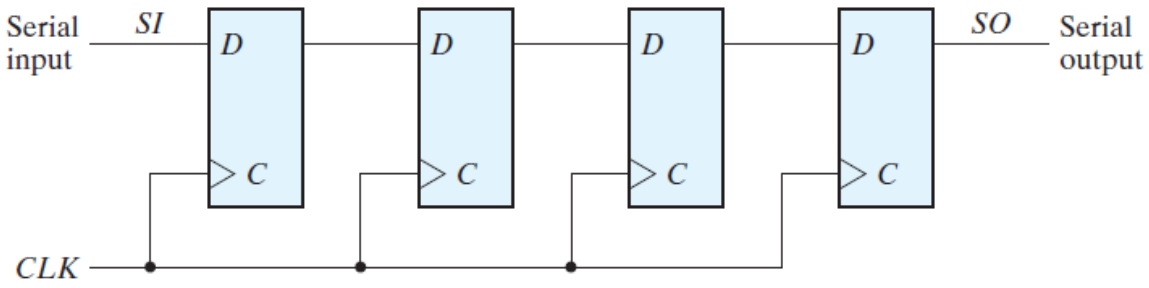
Deadline for report submission:
Friday, 03.01.2020 at 22:00 for all sections.

AIM

In this experiment you will design a serial adder and implement it in Verilog HDL.

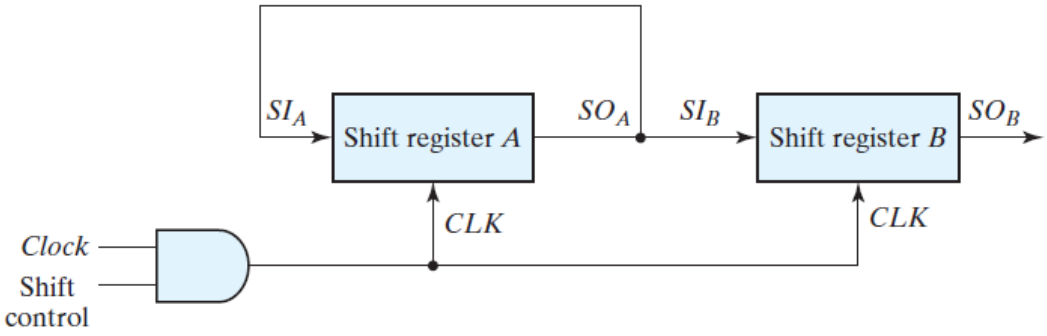
BACKGROUND

Shift registers are registers that are capable of shifting binary information held in each cell to its neighboring cell, either to the left or to the right. The simplest possible shift register is one that uses only flip-flops. A 4-bit unidirectional (left-to-right) shift register is shown in the figure below.

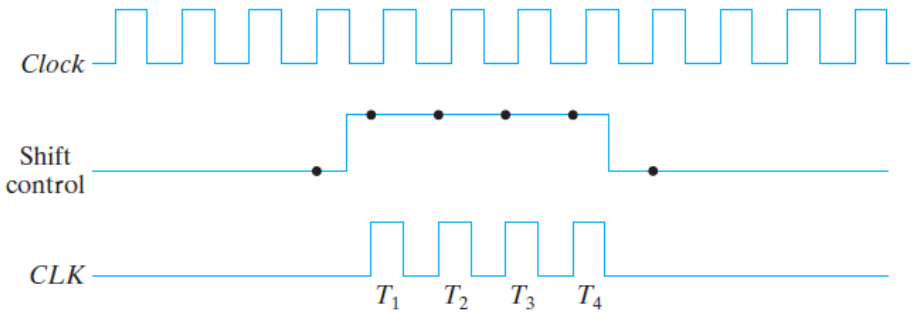


An example application would be serial transfer. A digital system is said to operate in a serial mode when information is manipulated one bit at a time. Computer may operate in a serial mode, a parallel mode, or a combination of both. Serial operations are slower, because information is manipulated one bit at a time. However, serial computers require less hardware, because one common circuit can be used over and over again to manipulate the bits coming out of shift registers.

In serial transfer, information is transferred one bit at a time by shifting the bits out of a source register into a destination register. The serial transfer of information from register A to register B is done with shift registers, as shown in figure below.



Timing Diagram:



The serial output (SO) of register A is connected to the serial input (SI) of register B. To prevent the loss of information stored in the source register, the information in register A is made to circulate by connecting the serial output to its serial input. The initial content of register B is shifted out through its serial output and is lost unless it is transferred to a third shift register.

Let's assume that the shift registers are 4 bit wide. The control unit that supervises the transfer of data must be designed in such a way that it enables the shift registers, through the shift control signal, for a fixed time of four clock pulses in order to pass an entire word. When the shift control signal is active, the output of the AND gate connected to the CLK inputs produces four pulses: T1, T2, T3, and T4. Each rising edge of the pulse causes a shift in both registers. After the fourth pulse, the shift control is changed to 0, and the shift registers are disabled.

Assume that the binary content of A before the shift is 1011 and that of B is 0010. The serial transfer from A to B occurs in four steps, as shown in the table below.

Serial-Transfer Example

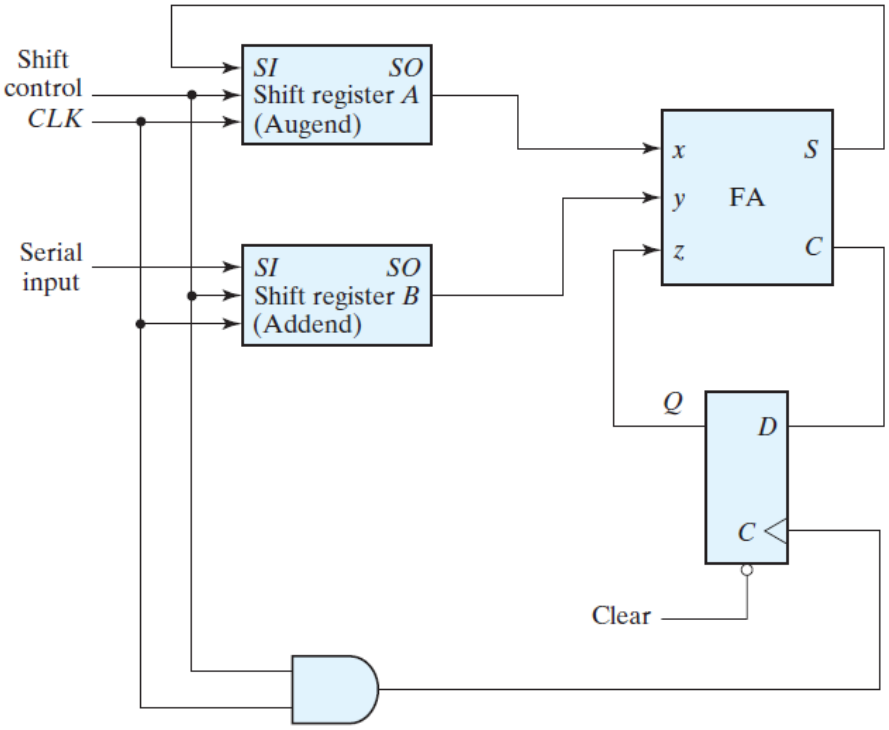
Timing Pulse	Shift Register *A*	Shift Register *B*
Initial value	1 0 1 1	0 0 1 0
After T_1	1 1 0 1	1 0 0 1
After T_2	1 1 1 0	1 1 0 0
After T_3	0 1 1 1	0 1 1 0
After T_4	1 0 1 1	1 0 1 1

For further background on implementing sequential circuits using structural approach in Verilog, please refer to the Lab Experiment 5 instructions.

LAB EXPERIMENT

You will implement a Serial Adder in Verilog using structural design approach.

The two binary numbers to be added serially are stored in two shift registers. Beginning with the least significant pair of bits, the circuit adds one pair at a time through a single full-adder (FA) circuit, as shown in the figure below.



The carry out of the full adder is transferred to a D flip-flop, the output of which is then used as the carry input for the next pair of significant bits. The sum bit from the S output of the full adder could be transferred into a third shift register. However, by shifting the sum into A while the bits of A are shifted out, it is possible to use one register for storing both the augend and the sum bits.

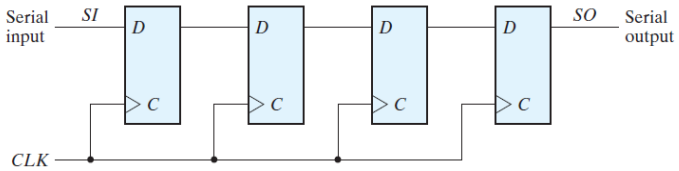
The operation of the serial adder is as follows: Initially, register A holds the augend, register B holds the addend, and the carry flip-flop is cleared to 0. The outputs (SO) of A and B provide a pair of significant bits for the full adder at x and y. Output Q of the flip-flop provides the input carry at z. The shift control enables both registers and the carry flip-flop, so at the next clock pulse, both registers are shifted once to the right, the sum bit from S enters the leftmost flip-flop of A, and the output carry is transferred into flip-flop Q. The shift control enables the registers for a number of clock pulses equal to the number of bits in the registers. For each succeeding clock pulse, a new sum bit is transferred to A, a new carry is transferred to Q, and both registers are shifted once to the right. This process continues until the shift control is disabled. Thus, the addition is accomplished by passing each pair of bits together with the previous carry through a single full-adder circuit and transferring the sum, one bit at a time, into register A.

Note that, unlike parallel adders (e.g. Ripple-Carry Adder) which require the number of full-adder circuits to be the same as the number of bits in the binary numbers (a 4-bit ripple-carry adder requires four full adder circuits), a serial adder requires only one full-adder circuit and a carry flip-flop. Excluding the registers, the parallel adder is a combinational circuit, whereas the serial

adder is a sequential circuit which consists of a full adder and a flip-flop that stores the output carry. This design is typical in serial operations because the result of a bit-time operation may depend not only on the present inputs, but also on previous inputs that must be stored in flip-flops.

Experiment Steps:

1. Implement a D flip-flop in Verilog.
2. Implement a Full Adder in Verilog. You may use any design approach you wish.
3. Implement a 4-bit Shift Register in Verilog using D flip-flip module and a structural design approach based on the figure below.



4. Use the figure of a Serial Adder circuit given above to implement a 4-bit serial adder in Verilog using structural design approach. Use D flip-flop, full adder, and shift register modules you implemented in the previous steps as necessary.
5. Write testbenches that test your Serial Adder module for the following initial states of the shift registers A and B:
    a. Shift-register-A = 4'b0001, Shift-register-B = 4'b1110
    b. Shift-register-A = 4'b0101, Shift-register-B = 4'b0011
    c. Shift-register-A = 4'b1111, Shift-register-B = 4'b0001
    d. Shift-register-A = 4'b1111, Shift-register-B = 4'b1111
6. Write a report that explains your design steps in detail, Verilog codes, and proof of correct results (e.g. screenshots of waveforms, variable changes from the console, etc.).

Report Submission:

The deadline for submission is Friday, 03.01.2020 at 22:00 for all sections. Zip your files (not .rar, only .zip files are supported by the system) and submit your work through https://submit.cs.hacettepe.edu.tr/index.php with the following file hierarchy:

```
- <studentID>.zip
    - D_ff.v
    - full_adder.v
    - shift_register.v
    - serial_adder.v
    - serial_adder_tb.v
    - report.pdf
```

Important notes:

- **One submission per group is enough. Clearly state the names and student IDs of the group members on the first page of your reports.**
- **Don't share your codes with other groups as we will perform the plagiarism check.**