

Combinational Circuits & Sequential Circuits

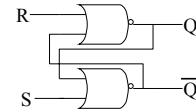
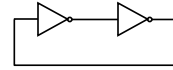
Two main classes of circuits:

- Combinational circuits**
 - Circuits without memory
 - Outputs depend only on current input values
- Sequential Circuits** (also called *Finite State Machine*)
 - Circuits with memory
 - Memory elements to store the state of the circuit
 - The state represents the input sequence in the past
 - Outputs depend on both circuit state and current inputs

1

Latches, Flip-Flops and Registers

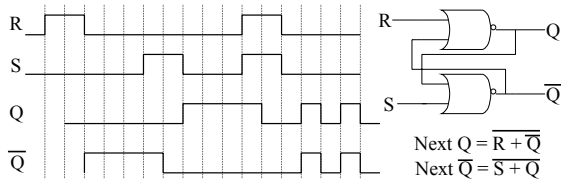
- These are devices to store information.
 - Latches and Flip-Flops – single bit
 - Registers – multiple bits
- Basic structure for storing a bit:
 - a pair of cross-coupled inverters
 - maintain a binary state indefinitely
 - Not useful as it lacks some practical means for changing its state
- Usually constructed by two cross-coupled NOR (or NAND) gates to provide some control signals.



2

SR Latch (Basic Latch)

- SR latch can act as a storage device (S : Set, R : Reset)
 - If R = 1 and S = 0, then Q goes to 0 and \bar{Q} goes to 1
 - If R = 0 and S = 1, then Q goes to 1 and \bar{Q} goes to 0
 - If R = 0 and S = 0, then Q and \bar{Q} remain where they are
 - If R = 1 and S = 1, then will not have a stable state (a bad idea for us)

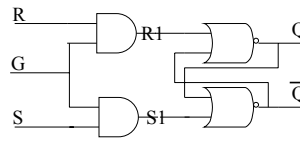


(Assume each step is one gate delay in time)

3

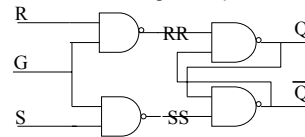
Gated SR Latch

- Adding an enable control input G (sometimes called CLK)



G	S	R	Next Q
0	x	x	Q (no change)
1	0	0	Q (no change)
1	0	1	0
1	1	0	1
1	1	1	Undesirable

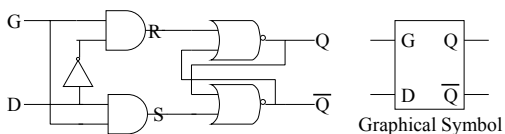
- Equivalent circuit using NAND (less transistors):



4

Gated D Latch (or called D Latch)

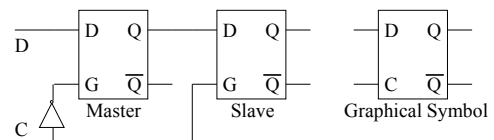
- We do not want R and S both to be 1
- But both can be (and should be) zero to store a value
- So we can force a zero when we want them to be zero together
- But only one of them will be 1 at a time
- These facts can be used to make a gated D latch
- G acts as a control signal
 - G = 0 means no writing, G = 1 allows writing
 - The value written is that of input D



5

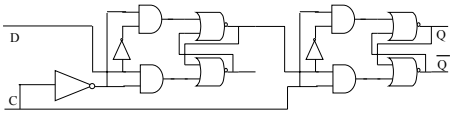
Master-Slave D Flip-Flop (D Flip-Flop)

- D Latch manages timing based on levels of signals (called a Level Sensitive circuit)
- We like to define a precise point in time when data gets stored (called a Edge-Triggered circuit)
- Data is written in flip-flop when an edge of clock signal C arrived
- This can be achieved by connecting two gated latches as below
- When C is low, first latch gates data on D, second does nothing
- When C goes high, second latch latches what is Q of first latch

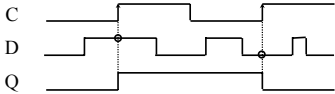


6

Operation of D Flip-flop (Edge-triggered FF)



- When clock C is low,
 - the first D latch samples the D input
 - the second D latch does not record any new value
- When C changes from low to high (i.e., at the up-going edge of C),
 - the first D latch store the D input value just before the edge
 - the second D latch copies the value in the first D latch into itself

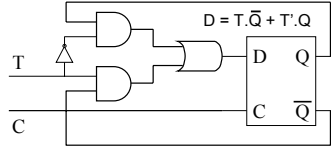


Assume there is no gate delay!!

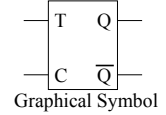
7

T Flip-Flop

- Remain the same when T=0
- Toggle the state when T=1



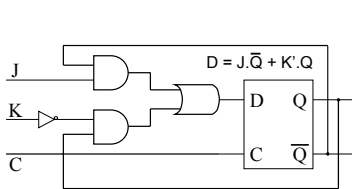
T	Next Q
0	Q
1	Q-bar



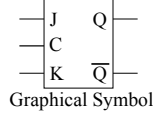
8

JK Flip-Flop

- Combines the behaviors of SR and T Flip-Flops
- It behaves as the SR flip-flop where J=S and K=R (except J=K=1)
- If J=K=1, it toggles its state like the T flip-flop



J	K	Next Q
0	0	Q
0	1	0
1	0	1
1	1	Q-bar



9

Timing Consideration

- Circuit timing is a very important consideration in the design of any electronic systems
- So far, we have ignore any timing problems

- We will consider the following timing issues:

For Flip-flops:

- Set-up time
- Hold time
- Propagation delay

For Combinational circuits:

- Contamination delay
- Propagation delay

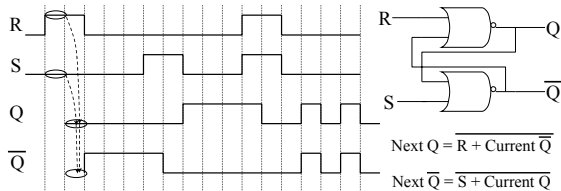
For Sequential circuits:

- Combining the timing of FFs and combinational circuits

10

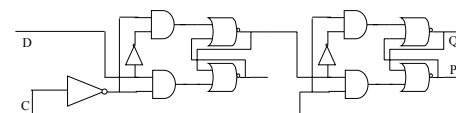
Delays in SR Latches

- A latch is a pair of cross-coupled inverting gates
 - They can be NAND or NOR gates as shown
 - Consider their behavior (each step is one gate delay in time)
 - From R and S to Q and Q-bar stable condition is reached in two gate delays

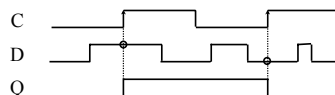


11

Operation of D Flip-flops (Edge-triggered FFs)



- When clock C is low,
 - the first D latch samples the D input
 - the second D latch does not record any new value
- When C changes from low to high (i.e., at the up-going edge of C),
 - the first D latch store the D input value just before the edge
 - the second D latch copies the value in the first D latch into itself



Assume there is no gate delay!!

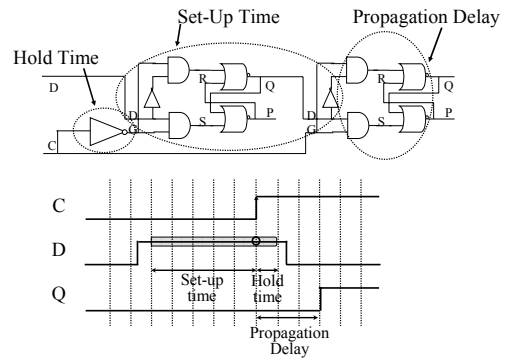
12

Timing Issues in D Flip-flops

- **Set-up time:**
 - Changes in input D propagate through many gates to the AND gates of the second D latch
 - Therefore D should be stable (i.e., set up) for at least *five* gate delays before the clock changes from low to high
- **Hold time:**
 - When clock changes from low to high, the first latch may still sample for one gate delay time.
 - Therefore, D should remain stable (i.e., hold) for at least *one* gate delay even after clock changes
- **Propagation delay:**
 - After clock changes from low to high, the value fetched by the second latch takes *three* gate delays to propagate to the output Q

13

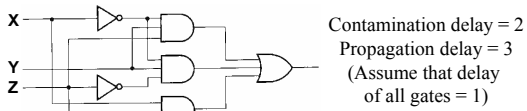
Set-up Time, Hold Time, Propagation Delay of FFs



14

Timing Issues of a Combinational Circuit

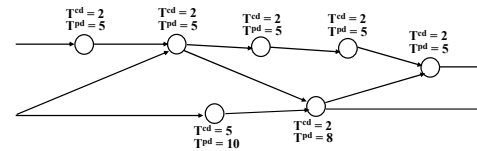
- **Contamination delay:**
 - Minimum delay before any output starts to change once input changes
- **Propagation delay:**
 - Maximum delay after which all outputs are stable once input changes



15

A Complicated Example

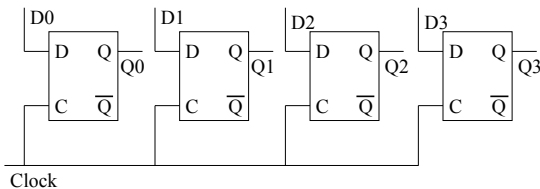
- Suppose a combinational circuit with several inputs and several outputs is constructed using several components
- The contamination delay (t_{cd}) and propagation delay (t_{pd}) of each component are given
- There are various paths from input to output in the circuit
 - We need to find the shortest path for contamination delay
 - We need to find the longest path for propagation delay
- For the circuit given below
 - Contamination delay =
 - Propagation delay =



16

Making a register

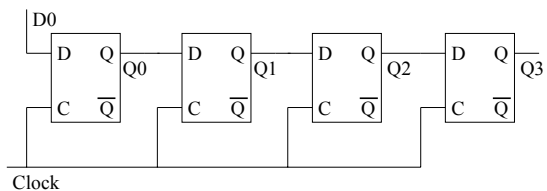
- Flip-flops can be connected to act as a register
- All clock signals are connected together to one clock
- All flip-flops get different input
- They all store one-bit information
- A 4-bit register is shown



17

Making a shift-register

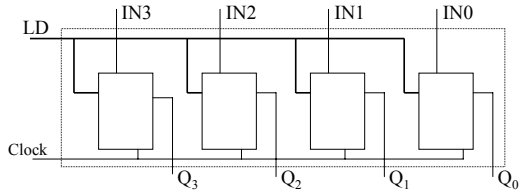
- Flip-flops can also be connected to act as a shift register
- All clock signals are connected together to one clock
- First flip flop gets a new input
- Others get input from previous flip-flop
- A 4-bit shift register is shown



18

Parallel-Access Register

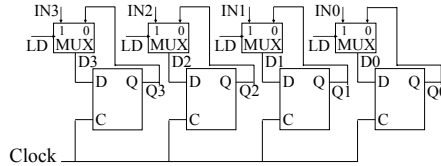
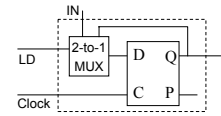
- We can add some logic to registers to create different device behaviors
- So far, the registers we have designed cannot hold a specific data value for more than one clock pulse
- A parallel-access register (also called register with parallel load) can hold a specific data value for more than one clock cycle
- A load signal LD is added



19

Implementation of Parallel-Access Register

- At the input of D flip-flops, a MUX is used to select whether to load a new input or to retain the old value
- All flip-flops get the same clock cycle
- signal LD = 1 means load new value
- signal LD = 0 means retain old value

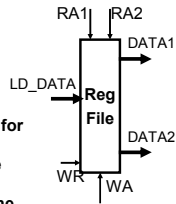


- Please refer to textbook Sec. 7.8.2 for parallel-access shift register

20

Register File

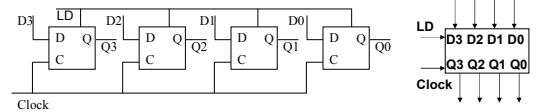
- Register file is a unit containing r registers
 - r can be 4, 8, 16, 32, etc.
- Each register has n bits
 - n can be 4, 8, 16, 32, etc.
 - n defines the data path width
- Output ports (DATA1 and DATA2) are used for reading the register file
 - Any register can be read from any of the ports
 - Each port needs a $\log_2 r$ bits to specify the read address (RA1 and RA2)
- Input port (LD_DATA) is used for writing data to the register file
 - Write address is also specified by $\log_2 r$ bits (WA)
 - Writing is enabled by WR signal



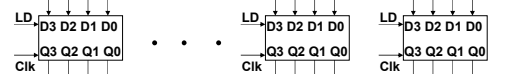
21

Register file design

- We will design an eight-register file with 4-bit wide registers
- A single 4-bit register and its abstraction are shown below



- We have to use eight such registers to make an eight register file

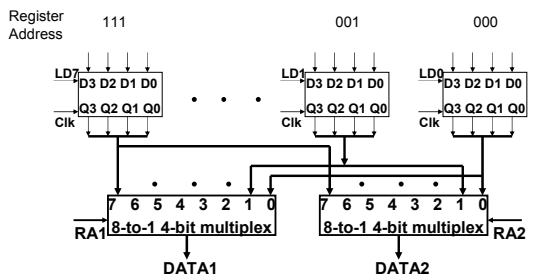


- How many bits are required to specify a register address?

22

Reading circuit

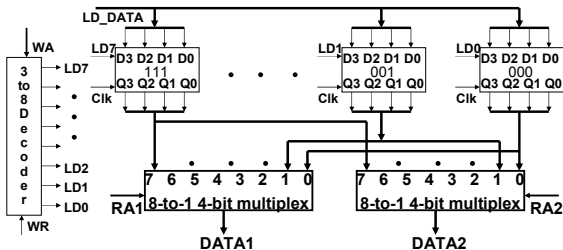
- A 3-bit register address, RA, specifies which register is to be read
- For each output port, we need one 8-to-1 4-bit multiplier



23

Adding write control to register file

- To write any register, we need register address (WA) and a write register signal (WR)
- A 3-bit write address is decoded if write register signal is present
- One of the eight registers gets a LD signal from decoder



24