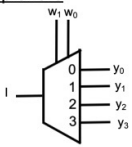
DEMUX

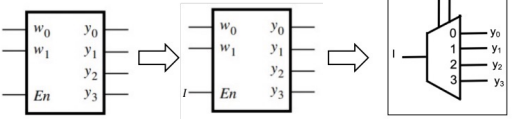
Demultiplexers:



w_1	w_0	y_0	y_1	y_2	y_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

$y_0 = \overline{w_1} \cdot \overline{w_0} \cdot I$
$y_1 = \overline{w_1} \cdot w_0 \cdot I$
$y_2 = w_1 \cdot \overline{w_0} \cdot I$
$y_3 = w_1 \cdot w_0 \cdot I$

Realizing Demultiplexer using a Decoder:



$y_0 = \overline{w_1} \cdot \overline{w_0} \cdot E_n$
$y_1 = \overline{w_1} \cdot w_0 \cdot E_n$
$y_2 = w_1 \cdot \overline{w_0} \cdot E_n$
$y_3 = w_1 \cdot w_0 \cdot E_n$

$y_0 = \overline{w_1} \cdot \overline{w_0} \cdot I$
$y_1 = \overline{w_1} \cdot w_0 \cdot I$
$y_2 = w_1 \cdot \overline{w_0} \cdot I$
$y_3 = w_1 \cdot w_0 \cdot I$

$y_0 = \overline{w_1} \cdot \overline{w_0} \cdot I$
$y_1 = \overline{w_1} \cdot w_0 \cdot I$
$y_2 = w_1 \cdot \overline{w_0} \cdot I$
$y_3 = w_1 \cdot w_0 \cdot I$

A_2, B_2	A_1, B_1	A_0, B_0	A>B	A<B	A=B
$A_2 > B_2$	×	×	1		
$A_2 < B_2$	×	×		1	
$A_2 = B_2$	$A_1 > B_1$	×	1		
$A_2 = B_2$	$A_1 < B_1$	×		1	
$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	1		
$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$		1	
$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$			1

XNOR for A=B

A	B	f
0	0	1
0	1	0
1	0	0
1	1	1

$\therefore f = (A_2 \odot B_2) \cdot (A_1 \odot B_1) \cdot (A_0 \odot B_0)$

A>B

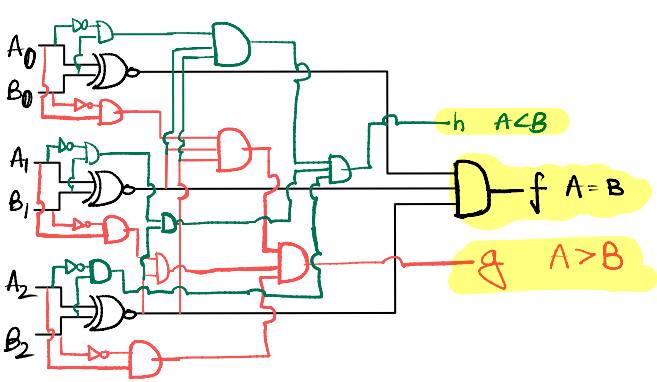
↳ $A_2 > B_2$ (OR) $A_2 = B_2, A_1 > B_1$ (OR) $A_2 = B_2, A_1 = B_1, A_0 > B_0$

$g = (A_2 \cdot \overline{B_2}) + (A_2 \odot B_2) \cdot (A_1 \cdot \overline{B_1}) + (A_2 \odot B_2)(A_1 \odot B_1)(A_0 \overline{B_0})$

$\therefore A < B$

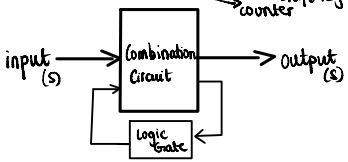
$A_2 < B_2$ (or) $A_2 = B_2, A_1 < B_1$ (OR) $A_2 = B_2, A_1 = B_1, A_0 < B_0$

$h = (\overline{A_2} \cdot B_2) + (A_2 \odot B_2)(\overline{A_1} \cdot B_1) + (B_2 \odot A_2)(A_1 \odot B_1)(B_0 \overline{A_0})$



Sequential circuits

- latch
- flip-flop
- Registers
- Shift registers
- counter

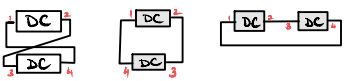


Combinational : present input depends only on present input

Sequential : present output depends on present & previous input

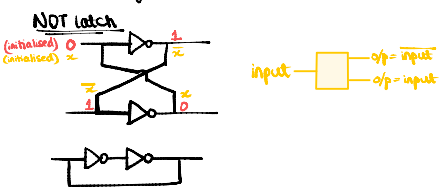
Latch

Structure

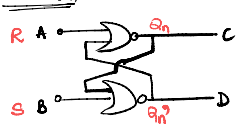


DC → NOT gate → NOT latch
- → NAND gate → NAND latch
- → NOR gate → NOR latch

NOT latch



NOR latch



present output = Q_n
next output = Q_{n+1}
R — DC — Q_n
S — DC — Q_n'

R	S	Q_n	Q_{n+1}	
0	0	0	0	} Q_n
0	0	1	1	
0	1	0	0	} Reset
0	1	1	0	
1	0	0	1	} Set
1	0	1	1	
1	1	0	forbidden	
1	1	1	forbidden	

A	B	C	D	
0	0	Q_n	Q_n'	memory
0	1	1	0	set
1	0	0	1	reset
1	1	0	0	← FORBIDDEN

} SR Latch

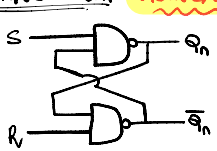
S \ RQn	00	01	11	10
0	0	1	0	0
1	1	1	x	x

$Q_{n+1} = S + \bar{R} Q_n$
characteristic eqn

Excitation table

Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

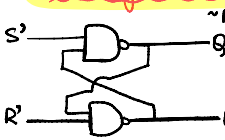
NAND latch — Active low S-R latch



S	R	Q_{n+1}	
0	0	forbidden	
0	1	1	set
1	0	0	reset
1	1	Q_n	memory

Active High S-R latch

~NAND latch w/ inverted input



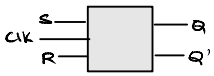
S' — DC — Q_n
R' — DC — Q_n'

Latch → Asynchronous circuit (w/out time, w/out clock)

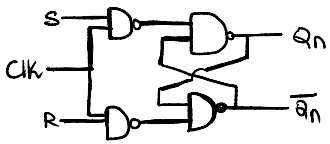
Clocked latch → synchronous circuit (flip flop {FF})

CLOCKED S-R Latch

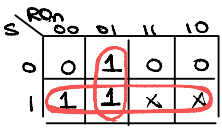
S-R FF (flip flop)
- 2NOR + 2AND
- 4NAND



S-R Latch
- 2NOR gate
- 2NAND gate + 2 NOT gate



Clk	S	R	Q_{n+1}	
1	0	0	Q_n	memory
1	0	1	0	reset
1	1	0	1	set
1	1	1	forbidden	
0	x	x	Q_n	memory

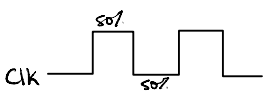


Clk = 1

Clk	S	R	Q_n	Q_{n+1}
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	x
1	1	1	1	x
0	x	x	0	0
0	x	x	1	1

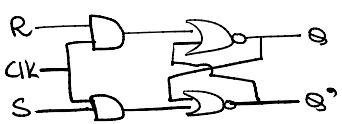
$Q_{n+1} = S + \overline{R}\,Q_n$ {Clk = 1}

$Q_{n+1} = Q_n$ {Clk = 0}

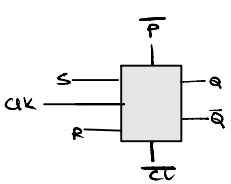


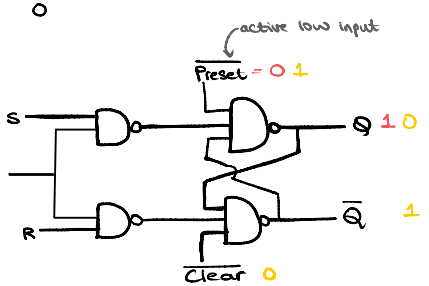
duty cycle = 50%

$f_{clk} = 1/T_{clk}$



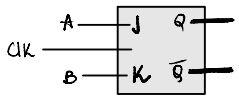
Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0



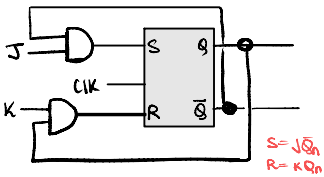


S	R	$\overline{Preset}$	$\overline{Clear}$	F/F operation
x	x	0	0	FORBIDDEN
x	x	0	1	Q=1, FF set
x	x	1	0	Q=0, FF clear
S	R	1	1	Normal FF operation

J-K F/P



clk	J	K	Q_{n+1}
↓	0	0	Q_n
↓	0	1	0
↓	1	0	1
↓	1	1	$\overline{Q_n}$
0	X	X	Q_n



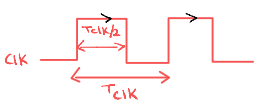
$S = J\overline{Q_n}$

$R = KQ_n$

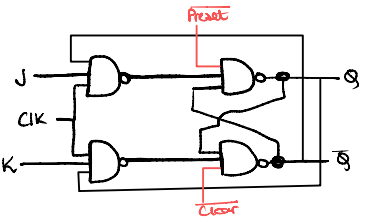
$$Q_{n+1}\big|_{S-R} = S + \overline{R}\,Q_n$$

$$Q_{n+1} = J\overline{Q_n} + (KQ_n)'\,Q_n$$
$$= J\overline{Q_n} + (K' + Q_n')\,Q_n$$

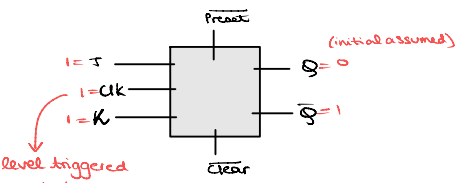
$$Q_{n+1} = J\overline{Q_n} + \overline{K}\,Q_n$$



Race around problem in level triggered clock JK ff.

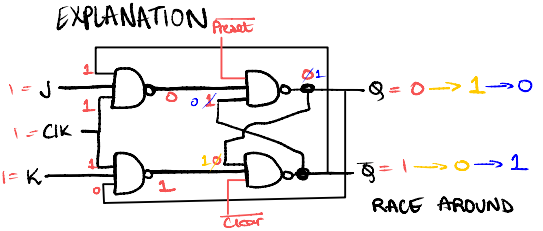


TOGGLE



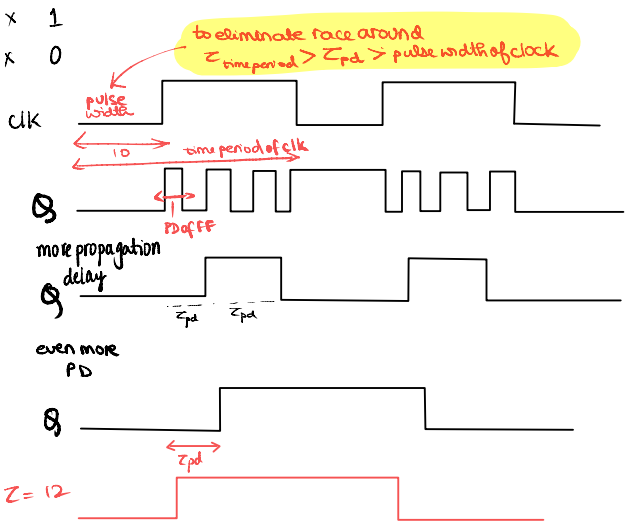
CLK	J	K	Q_n	Q_{n+1}
1	1	1	0	1, 0, 1, 0, ...
1	1	1	1	0, 1, 0, 1, ...

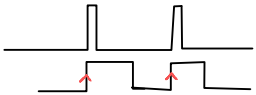
EXPLANATION



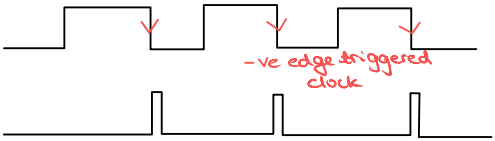
Excitation table

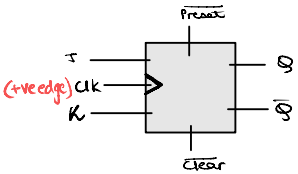
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

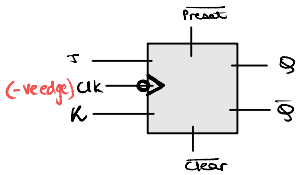


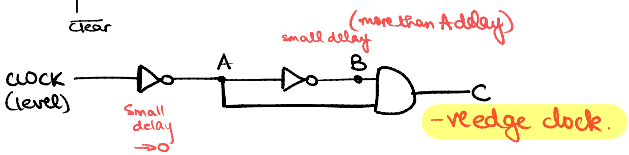


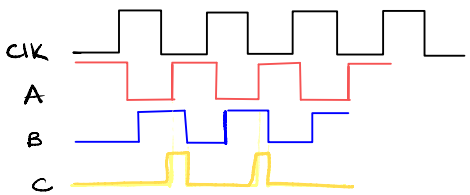
+ve edge triggered clock/
rising edge clock

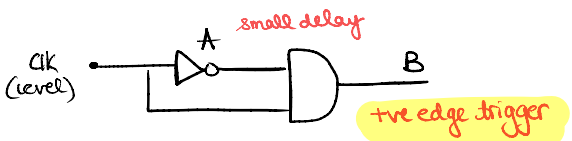


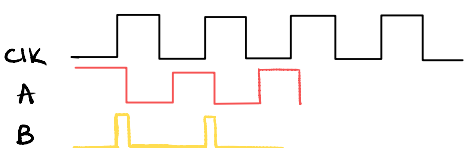






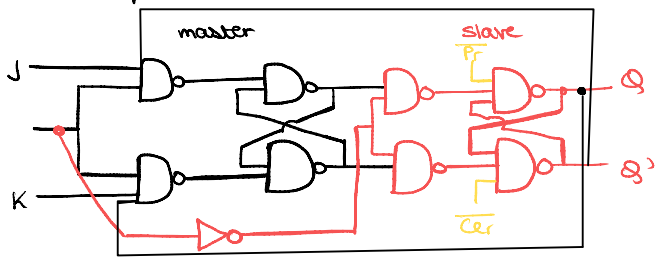


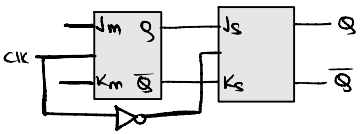


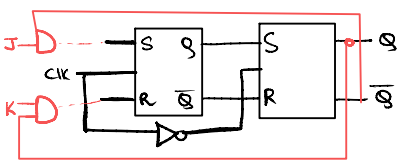


Master-slave J-K/F-F

works as a -ve edge trigger FF



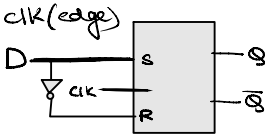




for const input in SR, edge & level have no diff

D/latch if clk (level)

D/FF if clk (edge)

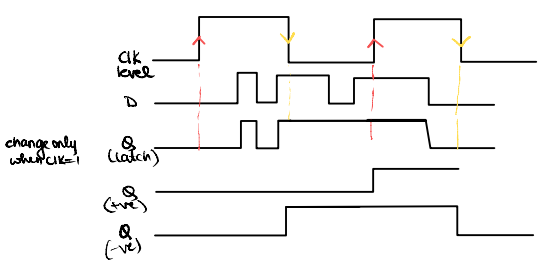


S = D

R = $\overline{D}$

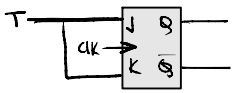
charac-eqn data FF

$Q_{n+1} = D$



T F/F

Toggle flip flop



$J = K = T$

$Q_{n+1} = J\overline{Q}_n + \overline{K}Q_n$

$Q_{n+1} = T \oplus Q_n$

$Q_{n+1} \oplus Q_n = T$

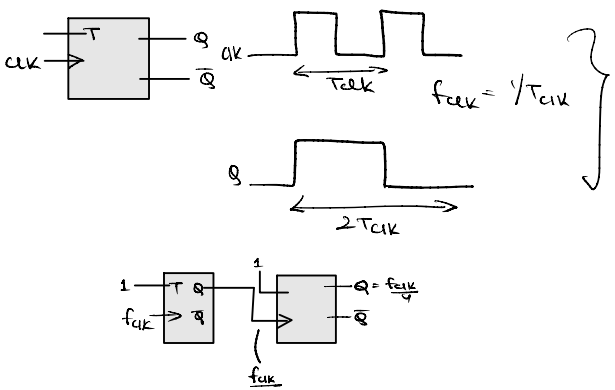
$Q_{n+1}|_{SR} = S + \overline{R}Q_n$

$Q_{n+1}|_{JK} = J\overline{Q}_n + \overline{K}Q_n$ → to construct Register (memory element)

$Q_{n+1}|_{D} = D$

$Q_{n+1}|_{T} = T \oplus Q_n$ → to implement counters

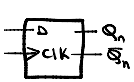
→ frequency divide by 2^n



Design one FF using another

D F/F available

→ implement J/K

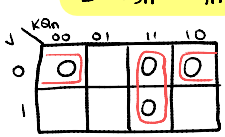


$D = f(J, K, Q_n)$

$Q_{n+1} = J\overline{Q}_n + \overline{K}Q_n$

$Q_{n+1} = D$

$D = J\overline{Q}_n + \overline{K}Q_n$



$D = (J + Q_n) \cdot (\overline{K} + \overline{Q_n})$

→ implement T

$D = T \oplus Q_n$

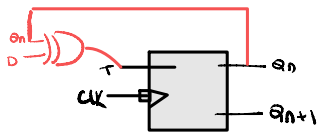
→ implement SR

$D = S + \overline{R}Q_n$

can't implement SR. 11, are shown invalid in approach.

T → D

Q_n	D	Q_{n+1}	$T = Q_n \oplus D$
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0



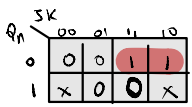
SR → JK.

available : SR

required : JK

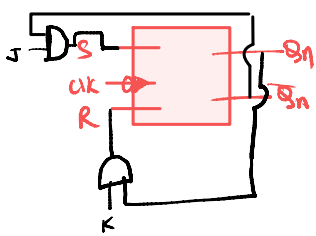
Q_n	J	K	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	X	0
1	0	1	0	0	1
1	1	0	1	X	0
1	1	1	0	0	1

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0



$S = \overline{Q_n} \cdot J$

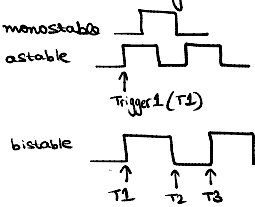
$R = Q_n K$

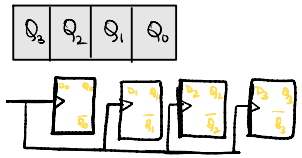


Register : 4bit Register → 4F/F : 4-D/FF

1F/F : store 1 bit binary info

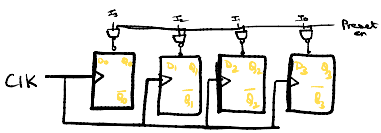
: 1 bit memory, bistable multivibrator





parallel in, parallel out Register

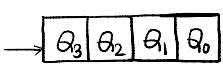
$D_3D_2D_1D_0$ = 1 1 1 1
then apply clock
↳ $Q_3Q_2Q_1Q_0$ = 1 1 1 1
↓
stored until we give clock again

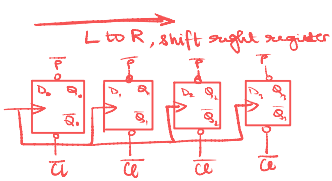


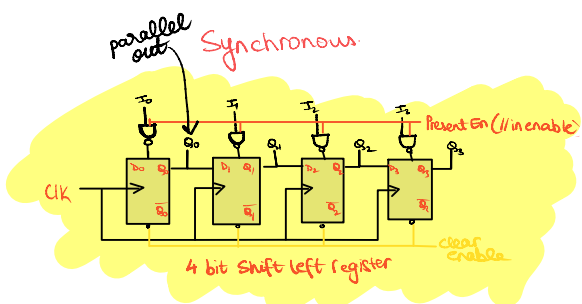
M1 $Q_3Q_2Q_1Q_0$ = x x x x
preset en = 0
Put D_3 D_2 D_1 D_0 = 1 1 1 1
∴ apply clk
$Q_3Q_2Q_1Q_0$ = 1 1 1 1

M2 initial $Q_3Q_2Q_1Q_0$ = 0000 (clr)
$D_3D_2D_1D_0$ = x x x x
$I_3I_2I_1I_0$ = 1 1 1 1
preset en = 1 1 1 1
$Q_3Q_2Q_1Q_0$ = 1 1 1 1

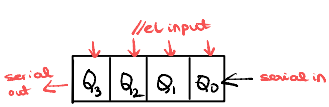
Shift Register → left / right







shift register
- → //el in //el out
- → //el in serial out
- → serial in //el out
- → serial in serial out



① parallel in, parallel out {0 clock is needed}

clear $Q_3 Q_2 Q_1 Q_0 = 0\ 0\ 0\ 0$

input data @ $I_3 I_2 I_1 I_0 = 1010$

② parallel in, serial out {n-1 clock is needed for n bit}

for output, only Q_3 can be accessed

$Q_3 Q_2 Q_1 Q_0 = 1\ 0\ 1\ 0$

clk	1	10 1 x
	2	10 x x
	3	1 x x x

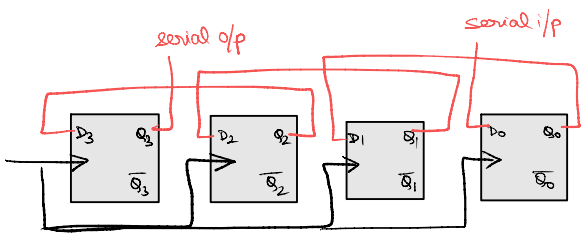
③ serial in, //el out {n clock needed}

1 1 1 0

$Q_3 Q_2 Q_1 Q_0$

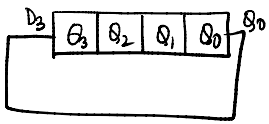
	Q_3	Q_2	Q_1	Q_0
initial	x	x	x	x
Put $D_0=1$, clk=1	1	x	x	x
$D_1=1$, clk=2	1	1	x	x
$D_2=1$, clk=3	1	1	1	x
$D_3=1$, clk=4	0	1	1	1

④ serial in, serial out {n + n-1 clocks}



Application of shift register

1. for parallel to serial data conversion
2. for serial to parallel data conversion
3. used to multiply w/ 2^n {Shift Left Register}
4. used to divide w/ 2^n {Shift Right Register}
5. As a ring counter



serial output & input connected.

{ MAX possible count by n-bit ring counter → n

2bit ring → cannot be started from all bits same

00, 01, 10, 11

	Q_1	Q_0
start	0	0
clk 1	0	0

only one state

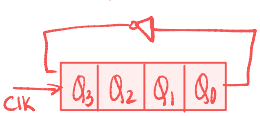
	Q_1	Q_0
start	0	1
clk 1	1	0
clk 2	0	1

2 diff states

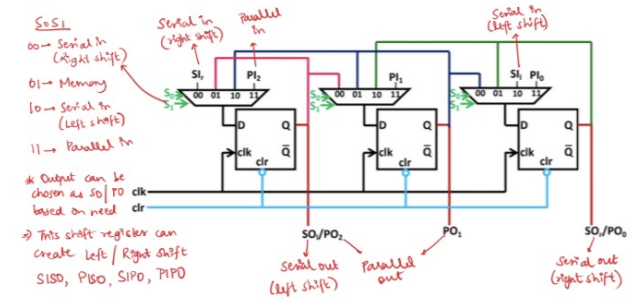
start	1	1
clk 1	1	1

max possible no of used states = 2n

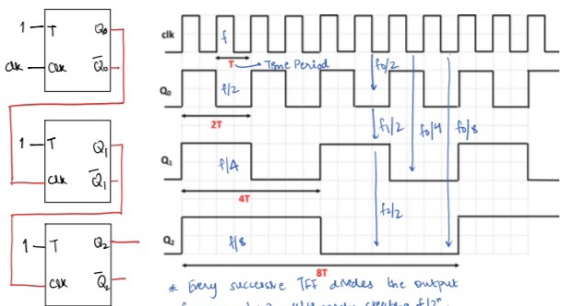
Johnson Counter (twisted ring counter) / (switch tail)
↳ shift register application



	Q_1	Q_0
initial	0	0
clk 1	1	0
2	1	1
3	0	1
4	0	0

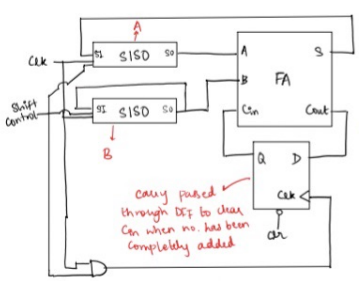


FREQUENCY DIVIDER



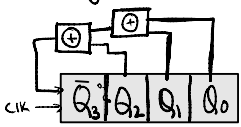
* Every successive TFF divides the output frequency by 2. Ultimately creating f/2ⁿ.

N-BIT ADDER USING SHIFT REGISTERS



* Same shift register is used to store sum & augend wherein sum of each bit is stored at the start of the SISO register.
* Clk AND'ed with shift control activated DFF only when bits are shifted within the shift registers; thus $B_n + A_n$ gets carry from $B_{n-1} + A_{n-1}$

Shift Register as synchronous counter



	Q_3	Q_2	Q_1	Q_0
initial	1	1	1	1
CLK ↓	1	1	1	1

Synchronous : provided to only one F/F (LSB)

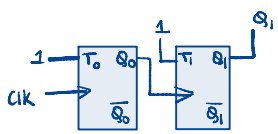
Asynchronous : simultaneously provided to all FF

Counters (T F/F)

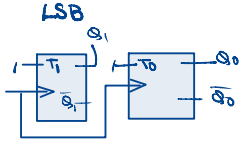
↳ Driven by clock signal, used to count no. of clock cycles. Also known as freq divider circuit

$$f_{o/p} = \frac{f_{CLK}}{N}$$

↳ any integer value



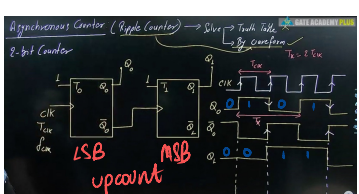
ASYNCHRONOUS Serial/Ripple



SYNCHRONOUS Parallel

up → start 0000 ... 1111

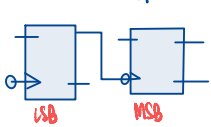
ASYNCHRONOUS

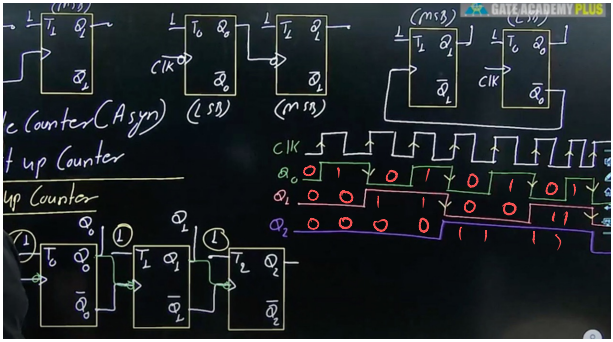


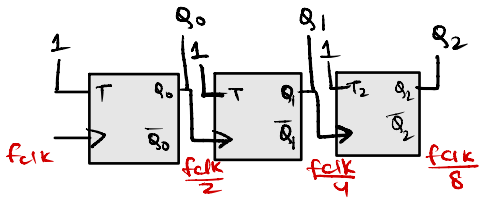
$f_{o/p} = \frac{f_{CLK}}{4}$

Q_1	Q_0
0	0
0	1
1	0
1	1

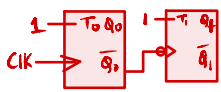
$f_{o/p} = \frac{f_{CLK}}{\text{no. of o/p states}}$

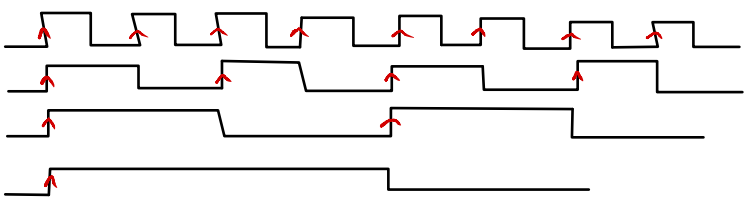






3 bit down counter





MOD-N counter

$T_{clk} \geq$ PD of all the F/F $= n \times$ PD of any one F/F

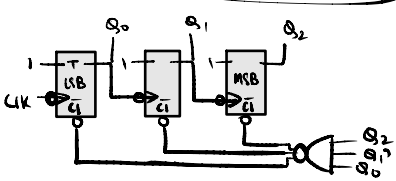
$T_{clk/min}$ = PD of all F/F

$f_{clk} \leq \frac{1}{\text{PD of all the F/F}}$

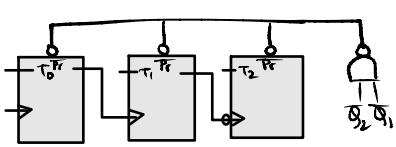
$f_{clk/max} = \frac{1}{n \times \text{PD of any F/F}}$

mod-5

0 → 1 → 2 → 3 → 4

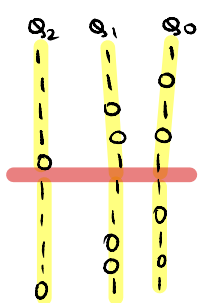


7 → 6 → 5 → 4 → 3 → 2



$f_{Q_2} = f_{Q_1} = f_{Q_0} = \frac{f_{clk}}{5}$

diff waveform



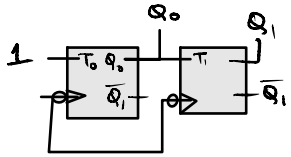
Synchronous Counter

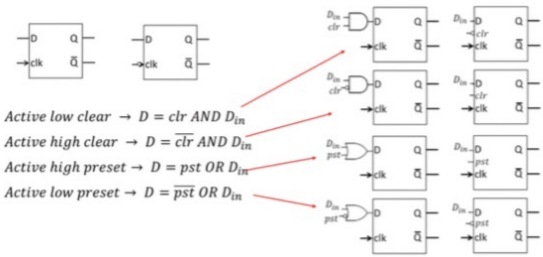
$0 \rightarrow 1 \rightarrow 2 \rightarrow 3$

Q_1	Q_0	Q_1^+	Q_0^+	T_1	T_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1

$T_0 = 1$

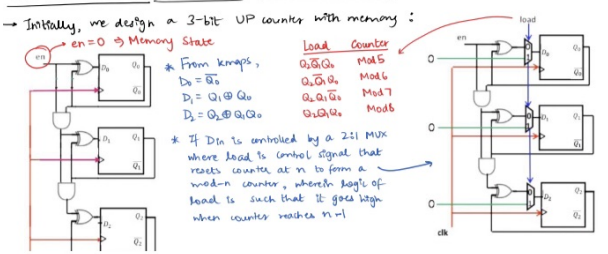
$T_1 = Q_0$





VARIABLE LENGTH SYNCHRONOUS 3-BIT COUNTER

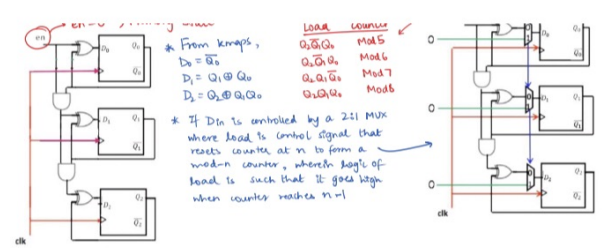
→ Initially, we design a 3-bit UP counter with memory :



* From kmaps,
  $D_0 = \overline{Q_0}$
  $D_1 = Q_1 \oplus Q_0$
  $D_2 = Q_2 \oplus Q_1 Q_0$

Load	Counter
$Q_2\overline{Q_1}Q_0$	Mod5
$Q_2\overline{Q_1}\overline{Q_0}$	Mod6
$Q_2Q_1\overline{Q_0}$	Mod7
$Q_2Q_1Q_0$	Mod8

* If Din is controlled by a 2:1 MUX where load is control signal that resets counter at n to form a mod-n counter, wherein logic of load is such that it goes high when counter reaches n-1



* From kmaps,
  $D_0 = \overline{Q_0}$
  $D_1 = Q_1 \oplus Q_0$
  $D_2 = Q_2 \oplus Q_1 Q_0$

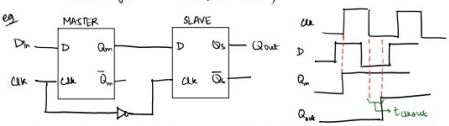
Load	Counter
$Q_2\overline{Q_1}Q_0$	Mod5
$Q_2\overline{Q_1}\overline{Q_0}$	Mod6
$Q_2Q_1\overline{Q_0}$	Mod7
$Q_2Q_1Q_0$	Mod8

* If Din is controlled by a 2:1 MUX where load is control signal that resets counter at n to form a mod-n counter, wherein logic of load is such that it goes high when counter reaches n-1

TIMING DELAY

→ $t_{clk\text{-}out}$: Time taken for final output to see a change after the active clock edge is called ($t_{cm} - t_{cm}$)

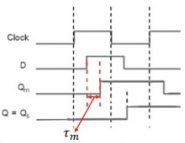
eg



* $t_{clk\text{-}out} \simeq$ Propagation delay of slave latch (T_s)

→ t_{setup}: Time for which Din must remain high before active clock edge of slave latch (i.e. Qm should be stable before slave's active clock edge)

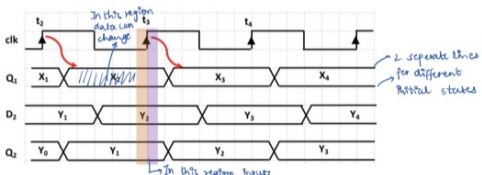
eg For same Master Slave FF above,

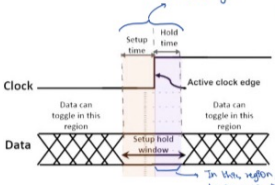


* $t_{setup} \simeq$ Propagation delay of master latch (T_m)

* If input changes in the time period before t_{setup}, the output of slave latch is unpredictable ie. output could be of this value or previous value of input

→ t_{hold}: When the clock signal switches value (for active edge), due to the rise time, there is a delay for the clock to change value; during this time, input must remain constant to avoid ambiguous output (current/prev value)

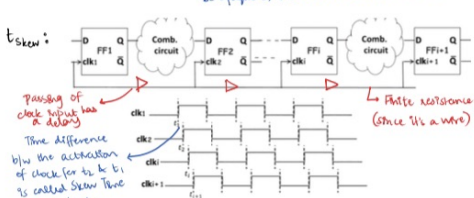




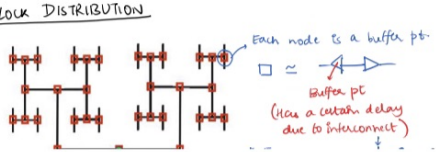
* Reason for "data cannot change":
1. For slave output, we need stable master output
↓
For this we need stable master input during t_{setup}.

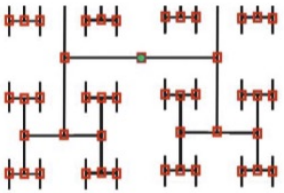
In this region, master is not guaranteed to be opaque ⇒ we need stable master output

→ t_{skew}:



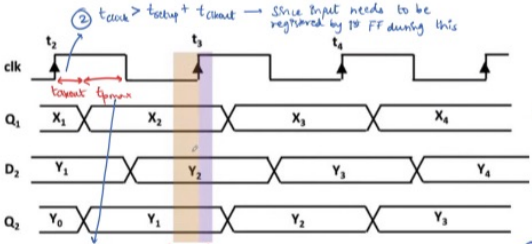
CLOCK DISTRIBUTION





(Has a certain delay due to interconnect)

* For any node, Imagine there were FFs at both ×, then the $t_{interconnect\ delay} = t_{skew}$ for FFs
* This is due to the interconnect having a finite resistance



Possible timing diagram of an arrangement of a combination of 2 FFs

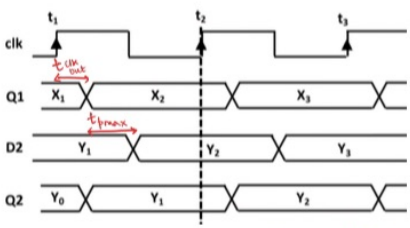
* t_{pmax} → Delay of combinational circuit connecting the 2 FFs

④ Also b/w the 2 FFs, there is a $t_{skew} = t_2 - t_1$

$\Rightarrow t_{clk} > t_{setup} + t_{clkout} + t_{pmax} + t_{skew}$

Maximum logic delay constraint

* This time constraint ensures before t_{setup} is up, the output of the first FF reaches the 2nd FF.
* But this also limits the value of the frequency of the circuit ⇒ circuit can only operate at certain speed.



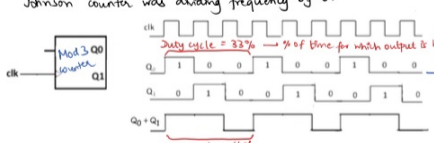
* The delay seen in the change of output at 1st FF and propagation of output to input of 2nd FF through combinational circuit is due to non zero t_{clkout} & t_{pmax}.
* If $t_{clkout} + t_{pmax} = 0$ then the value of D_2 can change in the setup hold window, thus causing an unstable output.
* To overcome this scenario,

Minimum logic delay constraint ← $t_{clkout} + t_{pmin} \geq t_{hold} + t_{skew}$

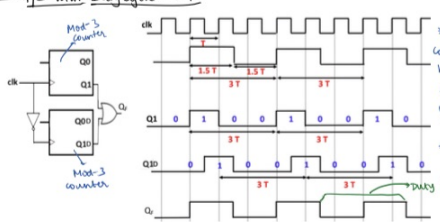
To incorporate clock delay

FREQUENCY DIVISION

→ To build a frequency divider by n, use a mod-n counter similar to how Johnson counter was dividing frequency by 6.



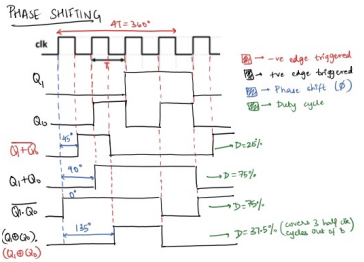
→ f/3 with Duty Cycle = 50%



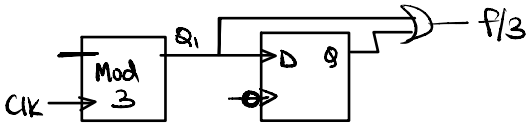
* Output of the combined circuit is high for half the time period of the output i.e. $3T/2 = 1.5T$
⇒ Duty cycle = 50%

* This divider with 50% duty cycle can also be generated using one Mod-3 counter & a DFF (earlier we required 4 DFFs now only 3 DFFs)
* Output of Mod-3 counter can be fed to DFF with opposite active edge thus producing same waveform as Q1D in above timing diagram.

PHASE SHIFTING



* Similarly frequency divider & phase shift can be created for any desired values by using Mod-n counters & taking combinations of input values.

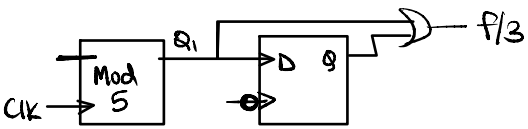


-ve triggered $+T/2$

+ve $+T$

Mod 5

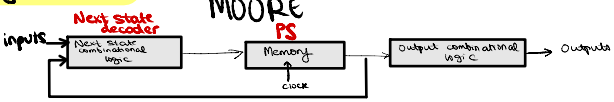
Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0



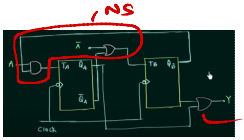
Mealey & Moore → o/p function of Present state only (PS)

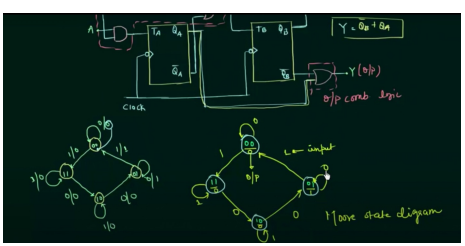
o/p function of PS & Input

MOORE



NS





MEALEY