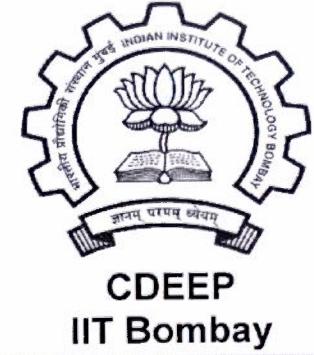
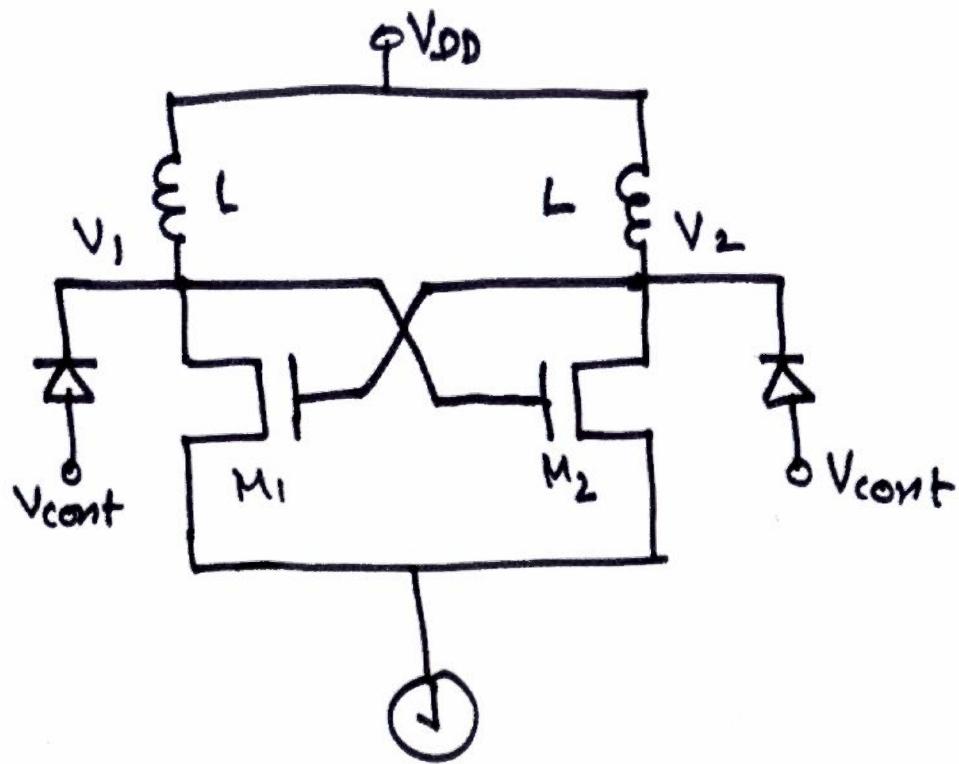


Tuning LC oscillator



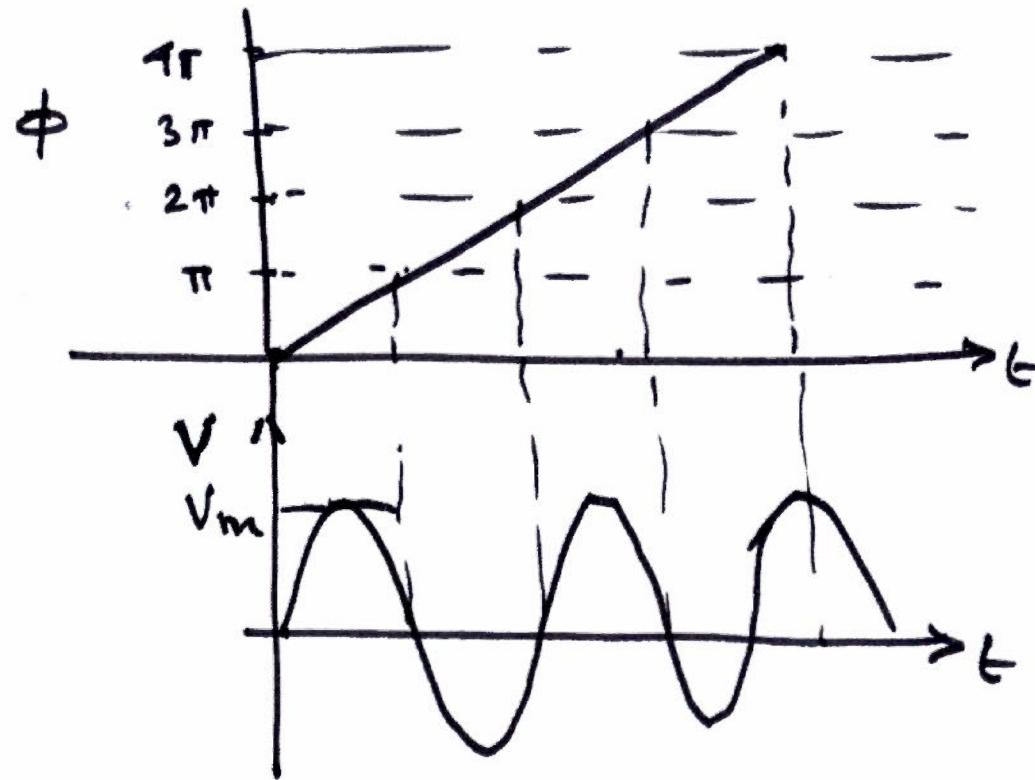
EE 618 L 29 / Slide 01

C_{varactor}

$$= \frac{g_0}{(1 + \frac{V_R}{\phi_B})^m}$$

$m = 0.4$ to 0.5

Mathematical Model of VCO

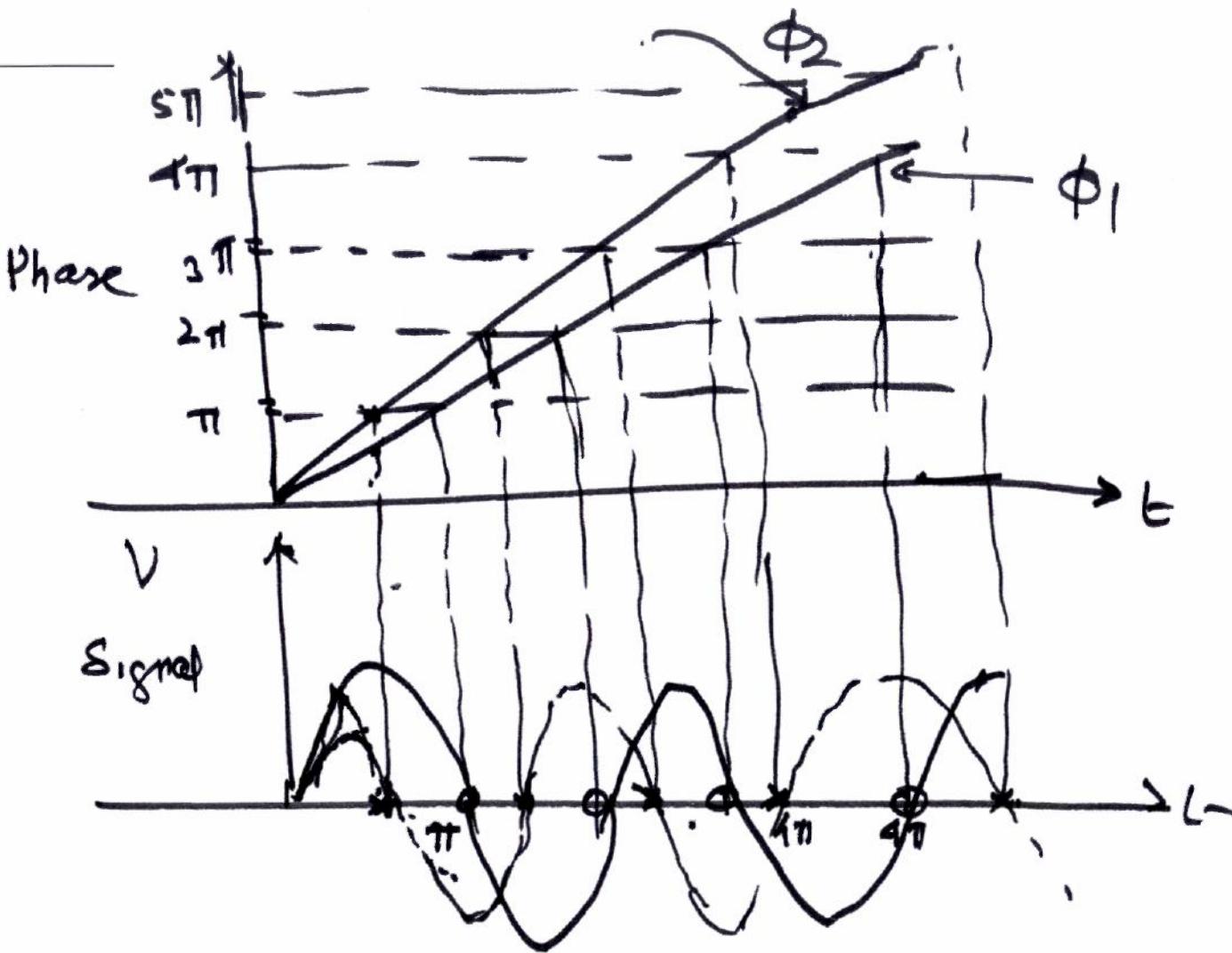


CDEEP
IIT Bombay

EE 618 L 29 / Slide 02

$$\omega = \frac{d\phi}{dt}$$

$$\omega t = \phi$$



CDEEP
IIT Bombay

EE 618 L 29 / Slide 03

$$\therefore \boxed{\frac{d\phi}{dt} = \omega}$$



CDEEP
IIT Bombay

EE 618 L 29 / Slide 09

$$\phi = \int \omega dt + \phi_0$$

In a VCO

$$\omega_{\text{out}} = \omega_0 + K_{\text{VCO}} V_{\text{cont}}$$

We have oscillator output -

$$v_o(t) = V_M \cos \phi t = V_M \cos \int \omega_{\text{out}} dt + \phi_0$$

$$\text{or } V_{\text{out}}(t) = V_m \cos \left\{ \omega_0 t + KV_{\text{co}} \int V_{\text{cont}} dt + \phi_0 \right\}$$

Assume $\phi_0 = 0$

$$\& \quad V_{\text{cont}} = V_m \cos \omega_m t \quad (\text{If } V_{\text{cont}} \text{ varies})$$



CDEEP
IIT Bombay

EE 618 L 29 / Slide 05

$$\text{Then } V_{\text{out}}(t) = V_m \cos \left\{ \omega_0 t + KV_{\text{co}} \int V_m \cos \omega_m t dt \right\}$$

$$= V_m \cos \omega_0 t - V_0 (\sin \omega_0 t) \left(KV_{\text{co}} \frac{V_m}{\omega_m} \sin \omega_m t \right)$$

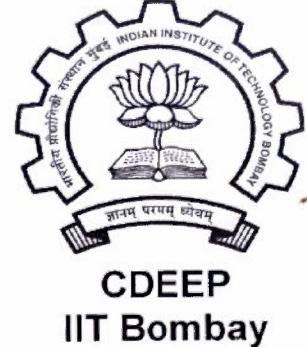
$$= V_m \cos \omega_0 t - \frac{KV_{\text{co}} V_m V_m}{2\omega_m} \left[\cos(\omega_0 - \omega_m)t - \cos(\omega_0 + \omega_m)t \right]$$



Phase Lock Loop

PLL was invented in 1930

1. Locking VCO to a Frequency
 2. Frequency Synthesizers
 3. Used in Mobile phones, TV, Receivers, Pager, Telephony
- (a) Digital PLL (b) Sinusoidal PLL
- 1965 - 1970
- (c) Optical PLL 1965



**Definition: A PLL is a Feedback system
That
Compares the Output Phase
With
The Input Phase.
Comparison is performed
By
PHASE COMPARATOR**

Analog Designers find issues related

to ① Jitter

② Phase Noise

very difficult to handle at high frequencies, which are now used in most Electronic & Communication Systems. Even Digital Systems on Board (PCB) also get critically affected due to two parameters or characteristics as above.



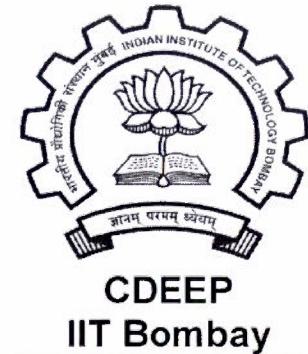
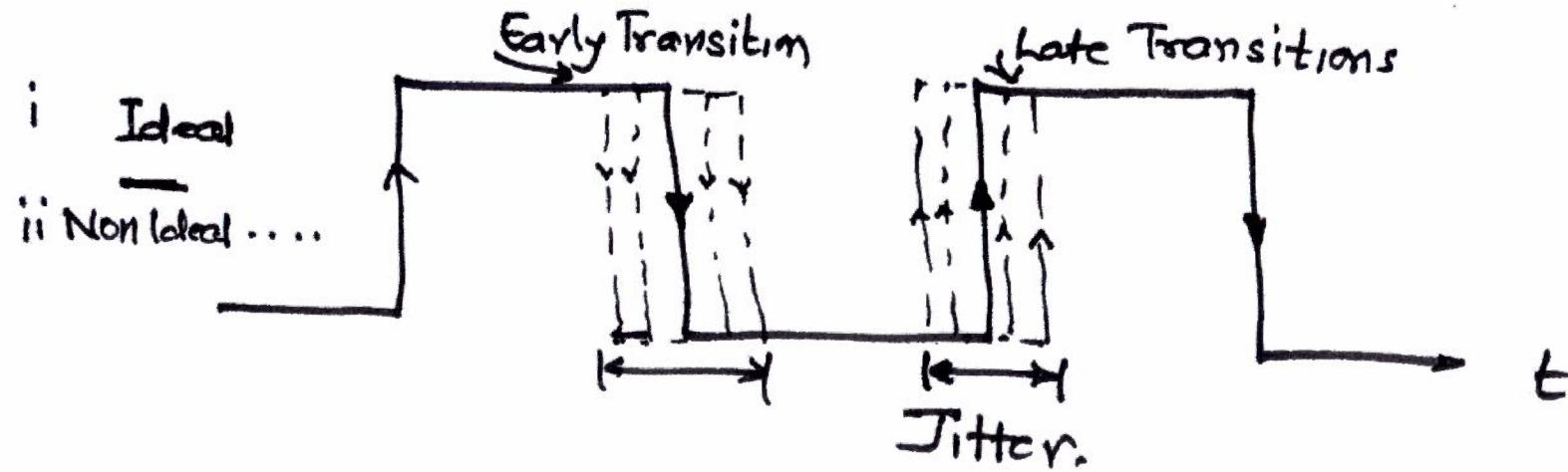
CDEEP
IIT Bombay

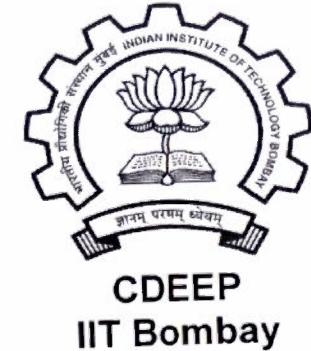
EE 618 L 29 / Slide 08

Example : 100 MHz Pulse waveform (50% Duty cycle), show period of 10 pscc and alternating at 5 pscc at every edge.

But this is only an Ideal Case

Transitions normally do not occur at 5 pscc edge and that creates, what we term as Jitter





Jitter :— i Deterministic
ii Random

i. Deterministic :— Cross talk

EMI radiation on Signal Path

Noise from surroundings

Switching - Power Supply Droop
Ground Bounce

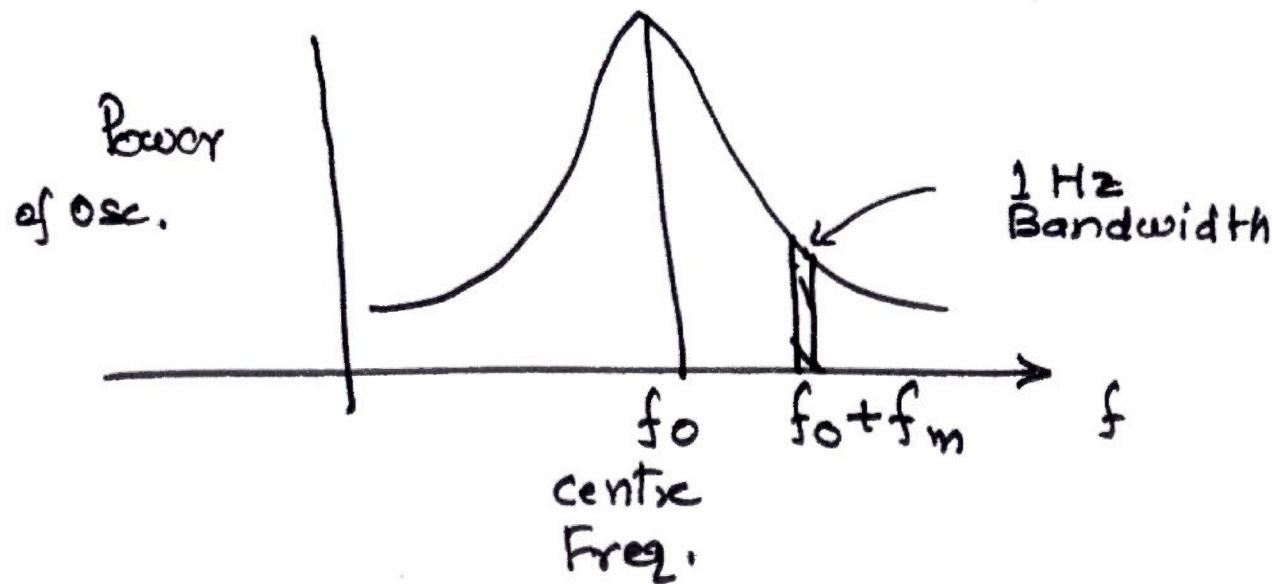
ii Random :— Temperature , Process Variations ,
Interface states

Random Jitter is Gaussian in nature

Multiple random jitter sources add to RMS Jitter .

Phase Noise :

Variation in Signal timings can also be represented in Frequency Domain and resultant Noise distribution is measured as Phase - Noise (PN)



If PN is = 0
Then all Oscillator Power goes to f_o
But PN spreads some power to adjacent frequencies, which results in Sidebands.



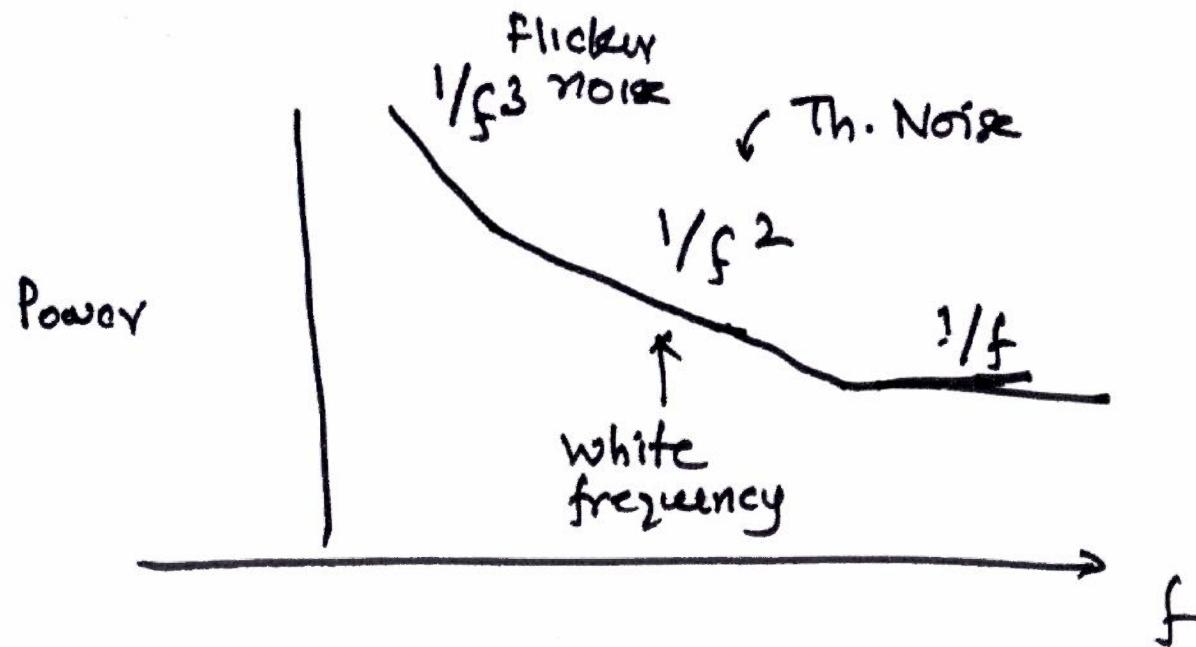
$$\text{Phase Noise} = \frac{\text{Power in } 1\text{ Hz Bandwidth at Offset freq}}{\text{Total Power of the Carrier}}$$

$$= \text{dBc/Hz}$$



CDEEP
IIT Bombay

EE 618 L 29 / Slide 12



17 Terminology

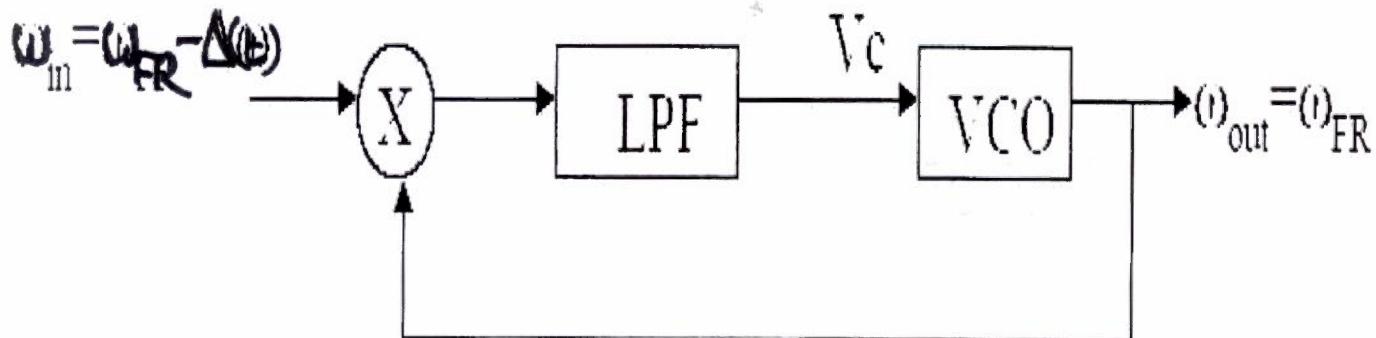
Frequency Locked

Phase error < permissible phase error (e.g. 1.5% of the output freq.)

~~Pull -~~

R-in range (Capture Range)

The frequency range over which a loop can acquire lock.



15

Terminology

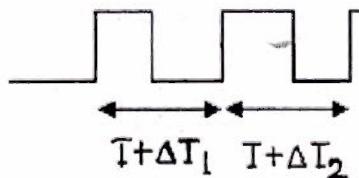
Jitter :

① Deterministic.

Random - specified in rms value or peak to

~~peak~~

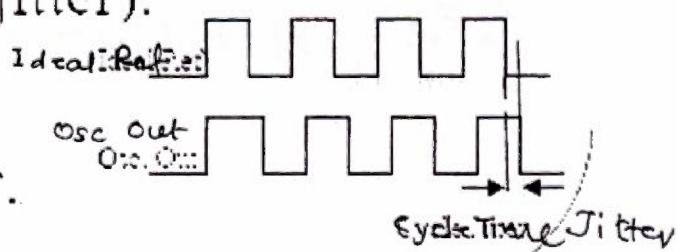
Cycle to cycle jitter.



Accumulated jitter (cycle jitter).

②

Duty cycle distortion jitter.

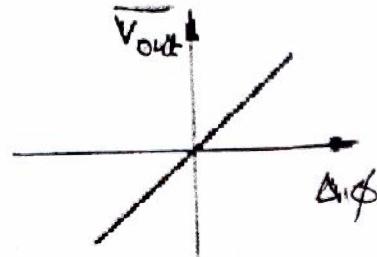
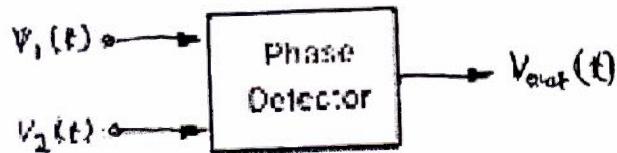


Basic

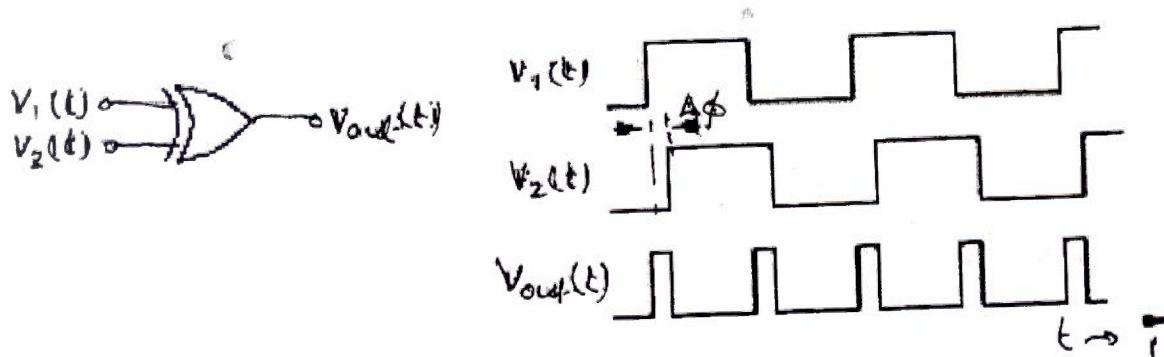
IC PLL

Phase →
Phase Detector

Phase Detector



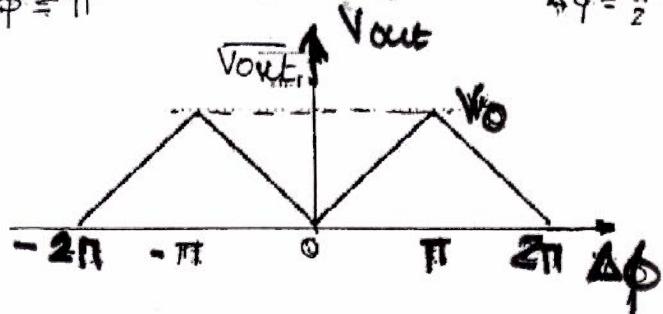
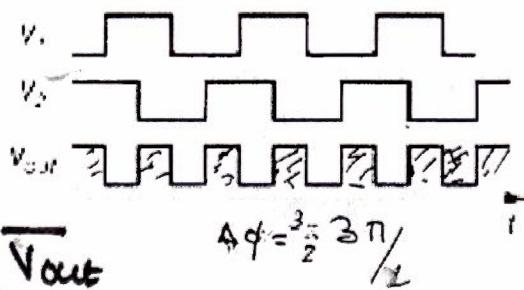
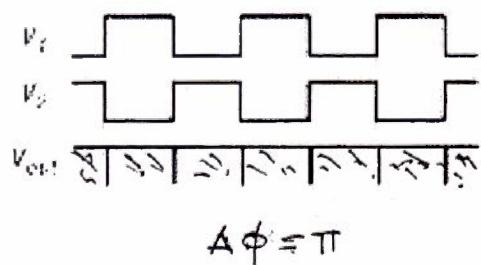
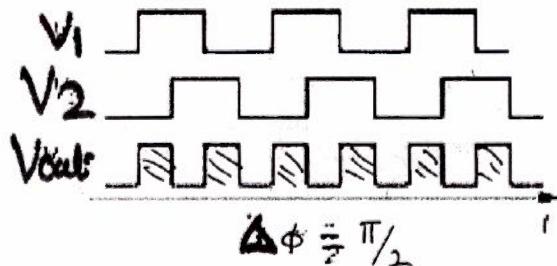
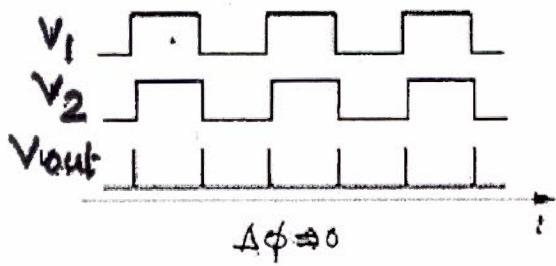
Definition of Phase detector



XOR gate as phase detector

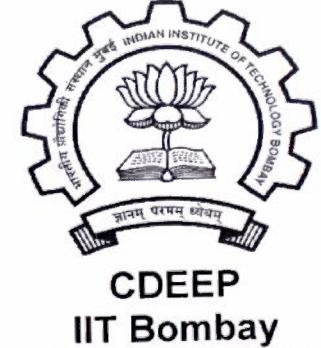
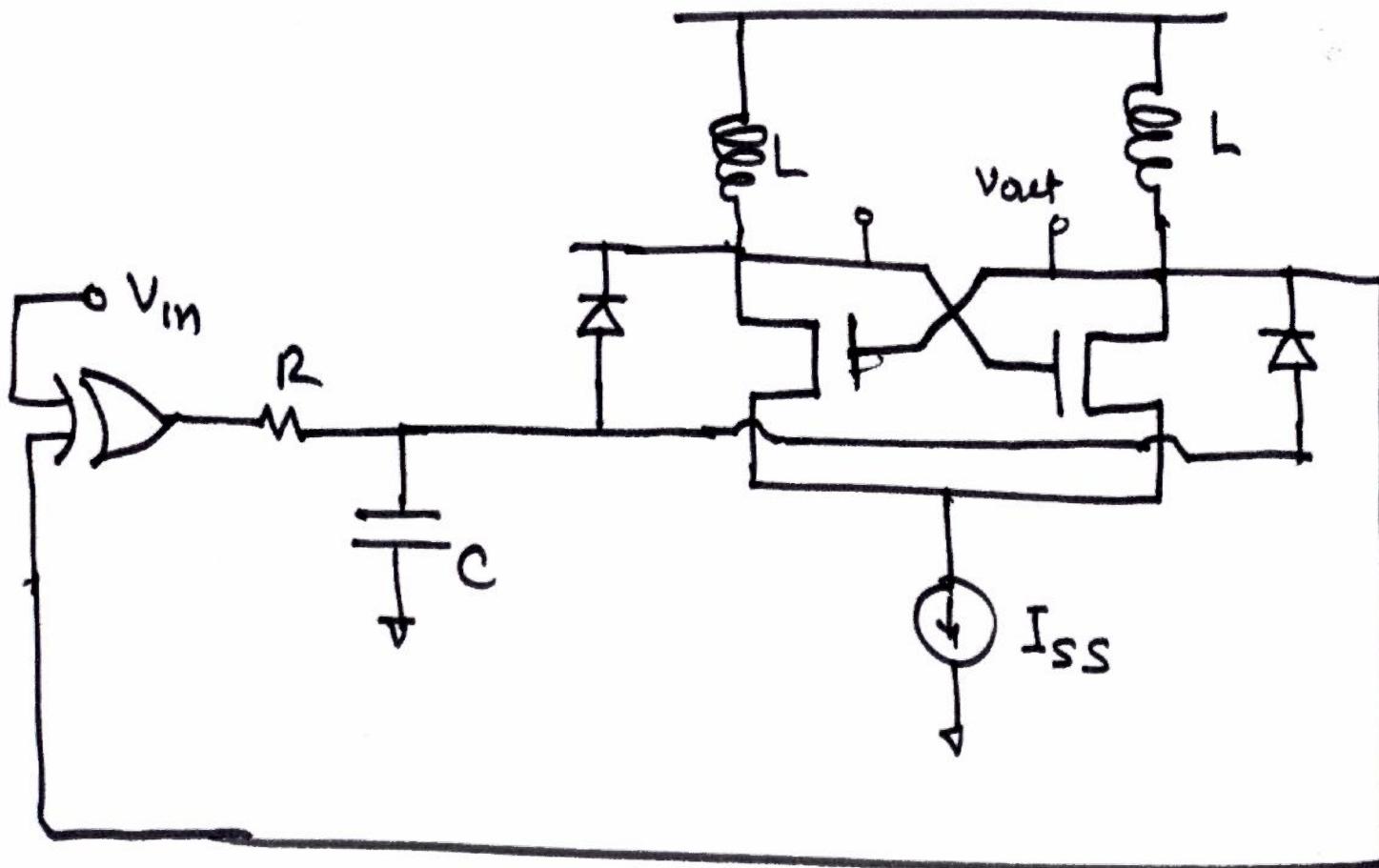
Ref. (Razavi)

Simple PLL



Phase detector output for different input skews

A Typical PLL in CMOS



EE 618 L29 / Slide 17

- \exists_m (LC) oscillator