LECTURE 080 – ALL DIGITAL PHASE LOCK LOOPS (ADPLL)

(Reference [2])

Outline

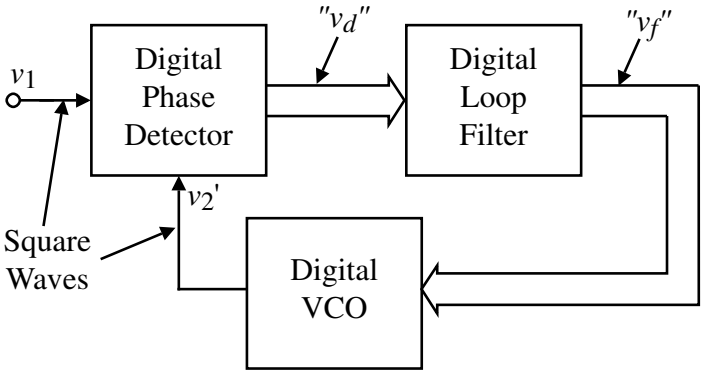
- Building Blocks of the ADPLL
- Examples of ADPLL Implementation
- ADPLL Design
- ADPLL System Simulation

BUILDING BLOCKS OF THE ADPLL

What is an All Digital PLL?

- An ADPLL is a PLL implemented only by digital blocks
- The signal are digital (binary) and may be a single digital signal or a combination of parallel digital signals.

Block Diagram of an ADPLL



Advantages:

- No off-chip components
- Insensitive to technology

DIGITAL PHASE DETECTORS WITH A PARALLEL OUTPUT

All of the phase detectors so far had only a 1-bit or analog output.

Flip-flop Counter PD

This phase detector counts the number of high-frequency clock periods between the phase difference of v_1 and v_2'.

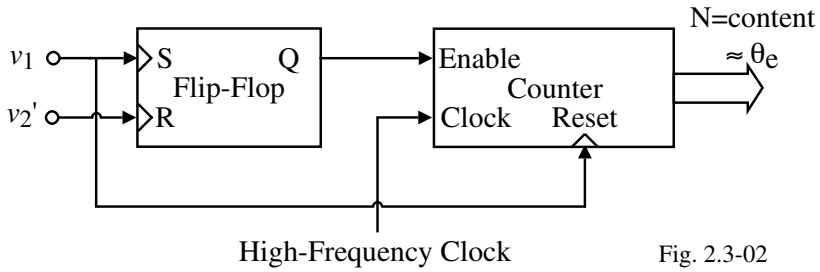


Fig. 2.3-02

Fig. 2.3-03

Nyquist Rate Phase Detector

Uses an analog-to-digital converter.

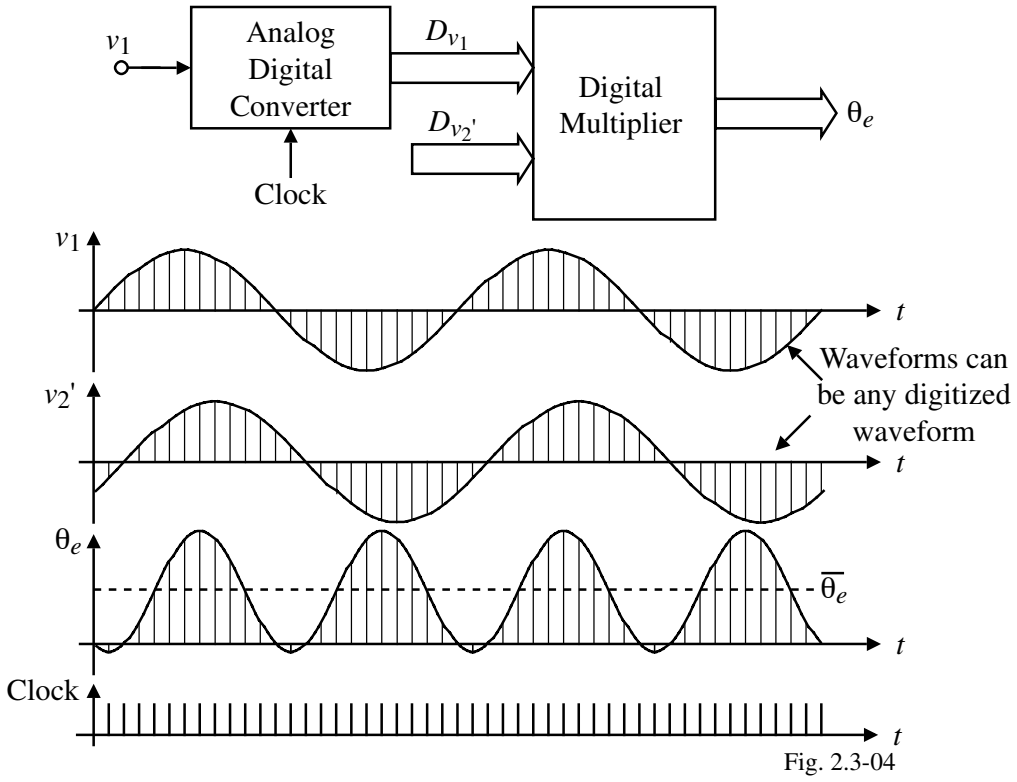


Fig. 2.3-04

Zero-Crossing Phase Detector

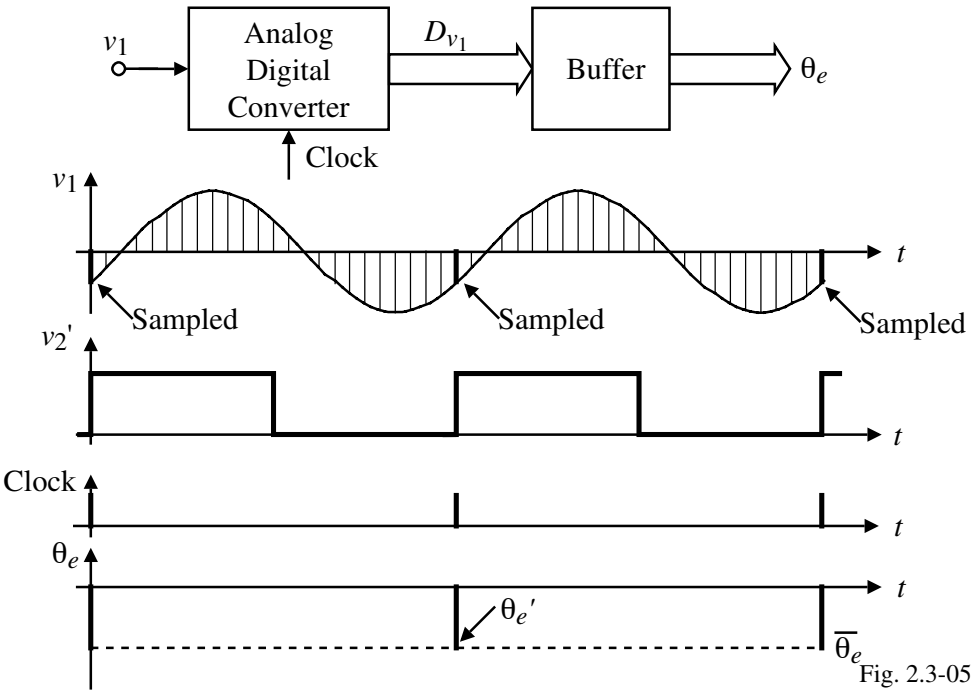


Fig. 2.3-05

Hilbert Transform Phase Detector

This phase detector uses the digital implementation of

$$\theta_e = \tan^{-1}\left[\frac{\cos\omega_o t\,\cos(\omega_o t+\theta_e) + \sin\omega_o t\,\sin(\omega_o t+\theta_e)}{\cos\omega_o t\,\sin(\omega_o t+\theta_e) + \sin\omega_o t\,\cos(\omega_o t+\theta_e)}\right]$$

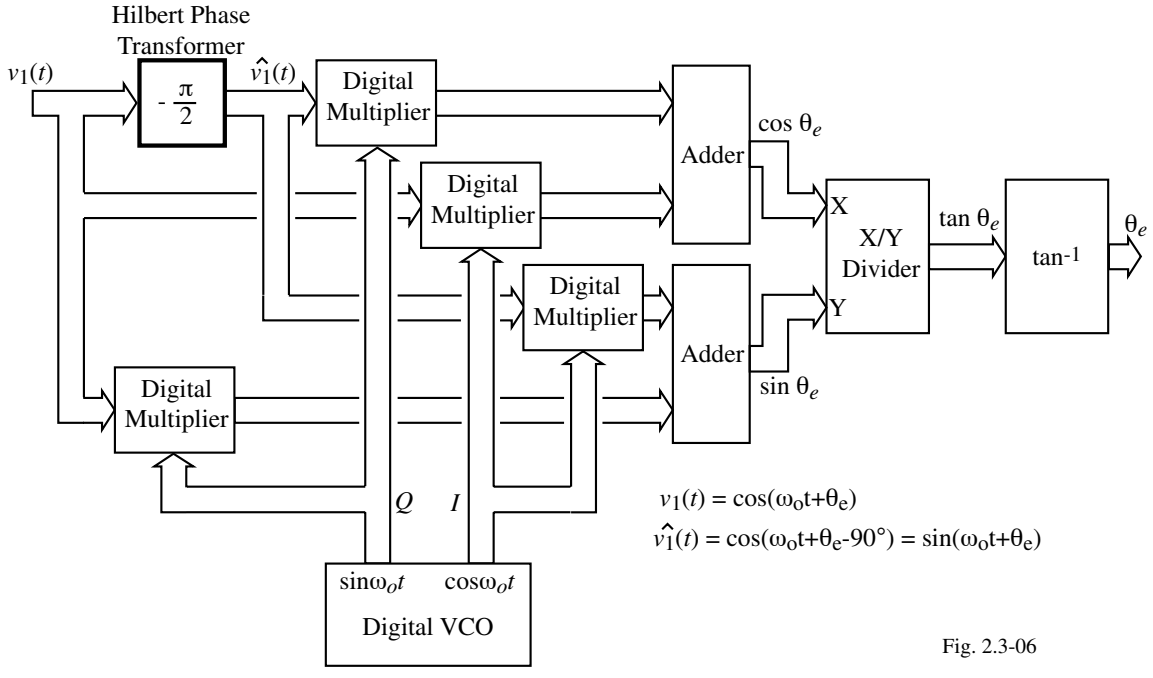


Fig. 2.3-06

Hilbert Transform Phase Detector – Continued

Waveforms:

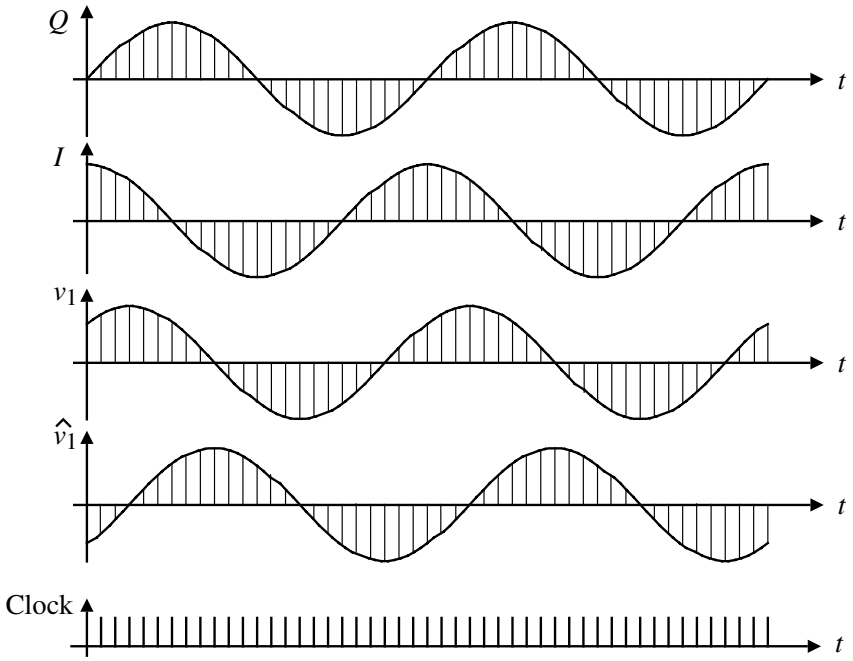


Fig. 2.3-07

Digital-Averaging Phase Detector

Similar to the Hilbert transform but simpler.

$\cos\theta_e$ and $\sin\theta_e$ are implemented by averaging (integrating) the output signals of the multipliers over an appropriate period of time.

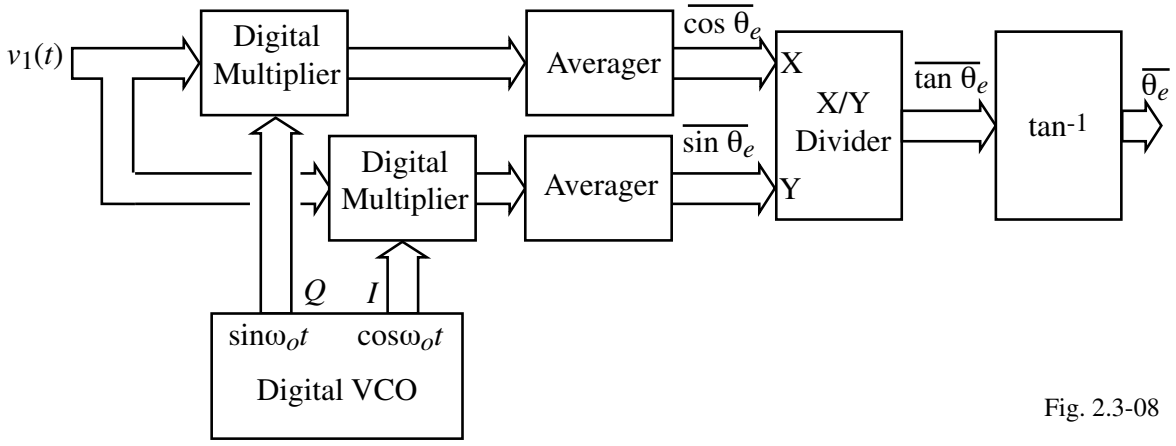


Fig. 2.3-08

This phase detector includes a filter function defined by the impulse function of the averaging circuitry.

LOOP FILTERS FOR THE ADPLL

Categories

1.) PD's not having a parallel digital output.

2.) PD's having a parallel digital output.

UP/DOWN Counter Loop Filter

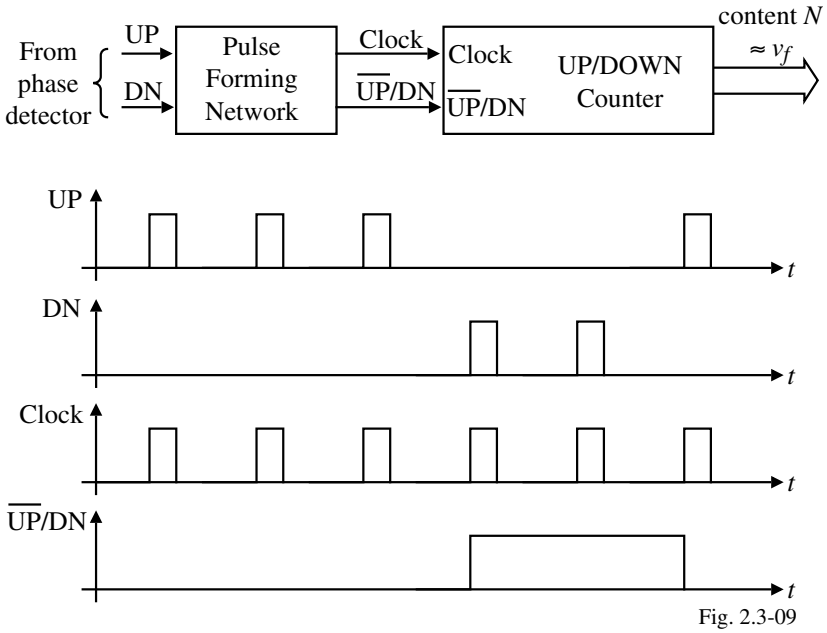


Fig. 2.3-09

The counter is an n-bit parallel output signal which is the weighted sum of the UP and the DN pulses. This signal approximates the function,

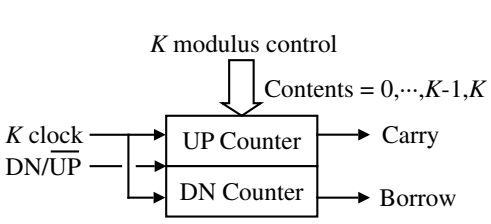
$$H(s) = \frac{1}{sT_i}$$

where

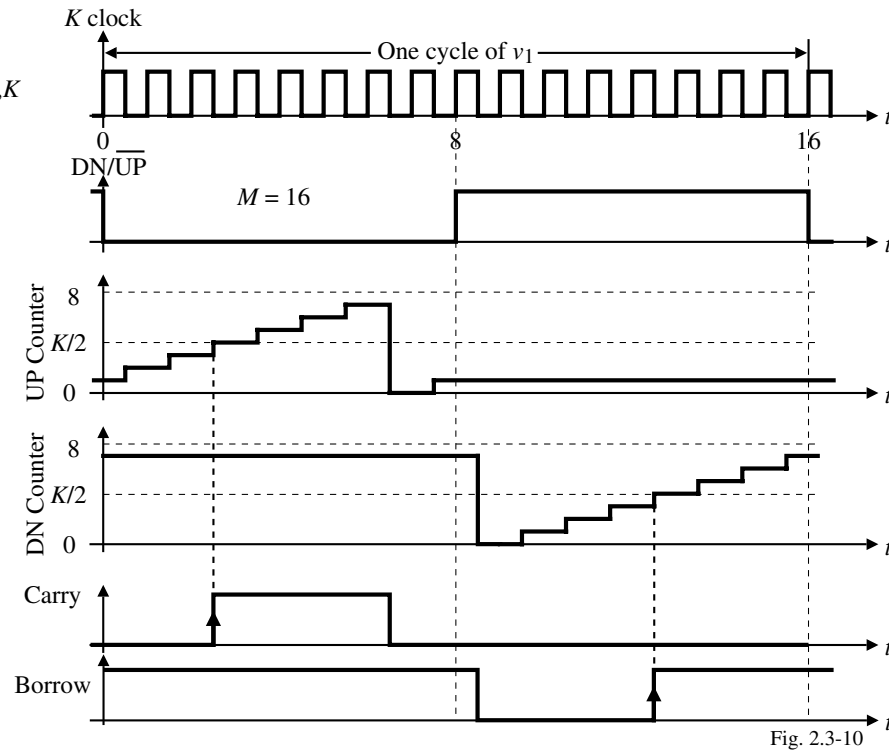
T_i = integrator time constant

K Counter Loop Filter (74xx297)

Works with EXOR or JK Flip-flop PDs. ($f_{clock} = Mf_o$)



Both counters count upwards.
Carry = 1when contents of the UP counter ≥ K/2.
Borrow = 1 when contents of the DN counter ≥ K/2.



Positive going edges of the Carry and Borrow control the DCO.

Fig. 2.3-10

N before *M* Loop Filter

Block diagram:

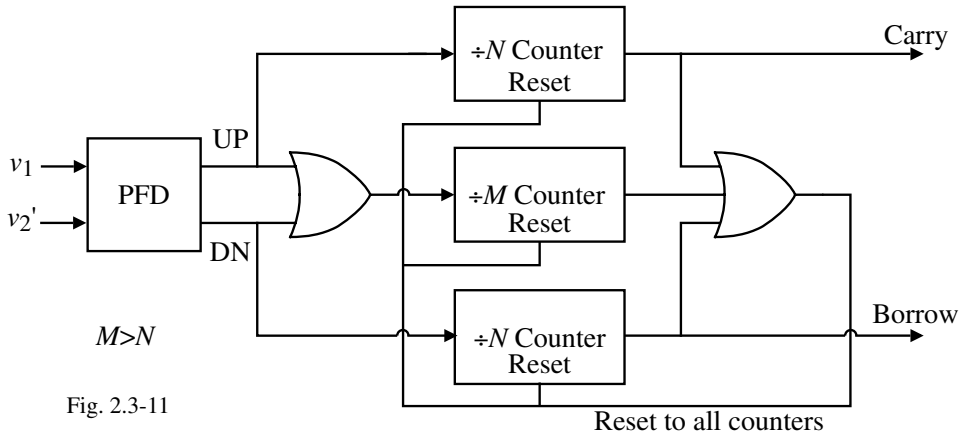


Fig. 2.3-11

Operation:

The upper ÷*N* counter will produce a carry pulse whenever more than *N* pulses of an ensemble of *M* pulses have been UP pulses.

The lower ÷*N* counter will produce a borrow pulse whenever more than *N* pulses of an ensemble of *M* pulses have been DN pulses.

The performance of the filter is very nonlinear.

Digital Loop Filters with an *N*-bit Parallel Input Signal

$$H(s) = \frac{O(s)}{I(s)}$$ If $I(s)$ is $\mathcal{L}[\delta(t)]$, then $O(s) = H(s)I(s) = H(s)$ = Impulse response

Convolution:

$$h^*(t) = T\sum_{n=0}^{\infty} h(n)\delta(t-nT)$$

Frequency Domain:

$$H^*(s) = T\sum_{n=0}^{\infty} h(n)e^{-nT}$$

z-Domain:

$$H(z) = H^*(s)\Big|_{z=e^{sT}} = T\sum_{n=0}^{\infty} h(n)z^{-n}$$

Infinite Impulse Response (*IIR*) Filters-

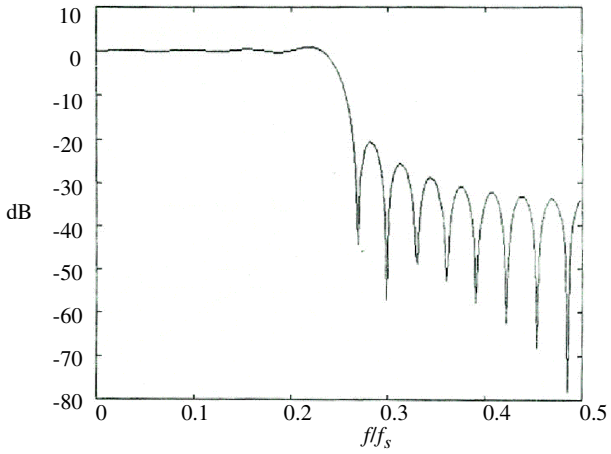
$n \rightarrow \infty$

Finite Impulse Response (*FIR*) Filters-

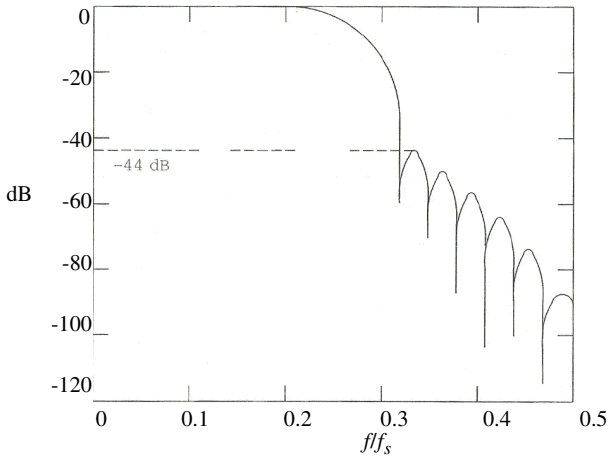
$n = N$

FIR Example

$N = 31$, no windowing



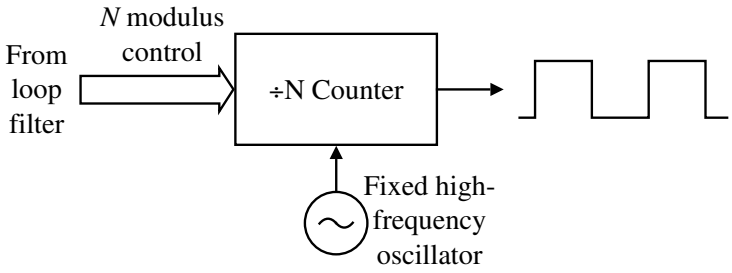
$N = 31$, Hanning window



Other windows: Hamming, Bartlett, Blackman, Kaiser, etc.

DIGITAL CONTROLLED OSCILLATORS

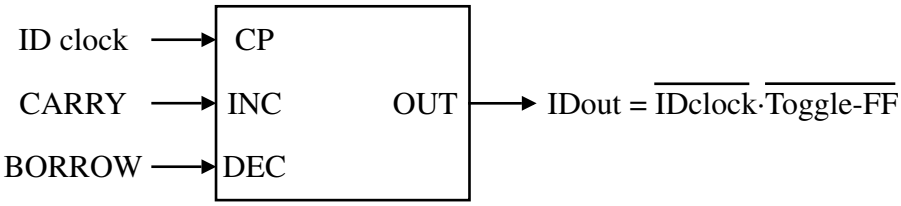
÷ *N* Counter



The *N*-bit output signal of a digital loop filter is used to control the scaling factor *N* of the ÷*N* counter.

Increment-Decrement Counter

Used with loop filters such as the *K* counter or *N* before *M* that output CARRY or BORROW pulses.



a.) No BORROW or CARRY pulses.

The toggle-FF switches on every positive edge of the ID clock if no CARRY or BORROW pulses are present.

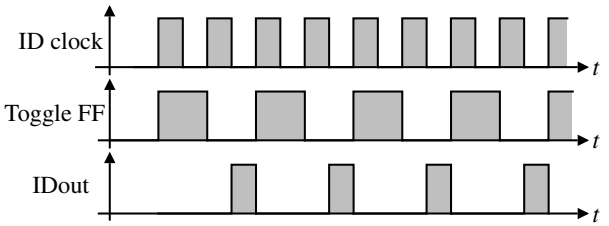


Fig. 2.3-14

b.) CARRY input applied when the toggle-FF is in the low state.

When the toggle-FF goes high on the next positive edge of the ID clock but stays low for the next two clock intervals, the IDout is advanced by one ID clock period.

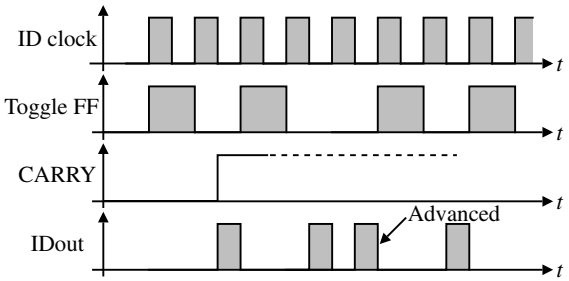


Fig. 2.3-15

Increment-Decrement Counter – Continued

c.) CARRY input applied when the toggle-FF is in the high state.

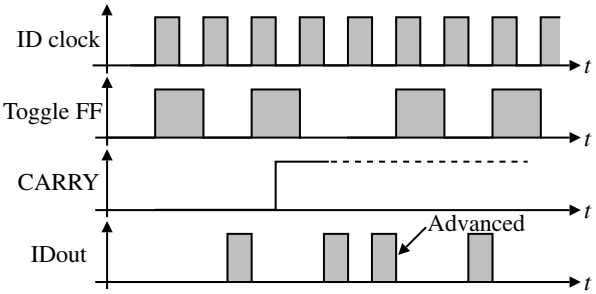


Fig. 2.3-16

The toggle-FF is set low for the next two clock intervals.

Because the CARRY can only be processed when the toggle-FF is in the high-state, the maximum frequency of the IDout signal is reached when the toggle-FF follows the pattern of "high-low-low-high-low-low".
Therefore, the maximum IDout frequency = 2/3 ID clock frequency. This will limit the hold range of the ADPLL

d.) Application of a BORROW pulse.

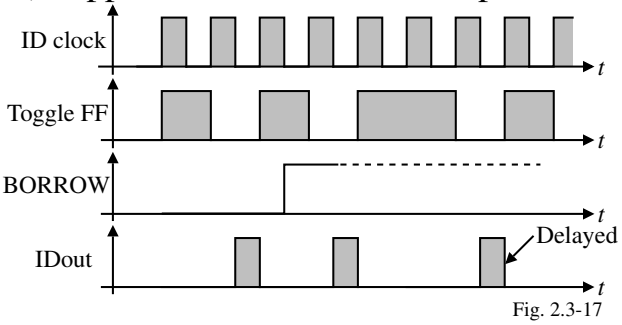


Fig. 2.3-17

A BORROW pulse causes the toggle-FF to be set high on the suceeding two positive edges of the ID clock.

This causes the next IDout pulse to be delayed by one ID clock period. The toggle-FF has the pattern of "low-high-high-low-high-high" which gives the min. IDout frequency = 1/3 ID clock frequency.

Basically, 1 CARRY pulse adds 1/2 cycle and 1 BORROW pulse removes 1/2 cycle.

Waveform Synthesizer DCO

Probably more suitable for software implementation.

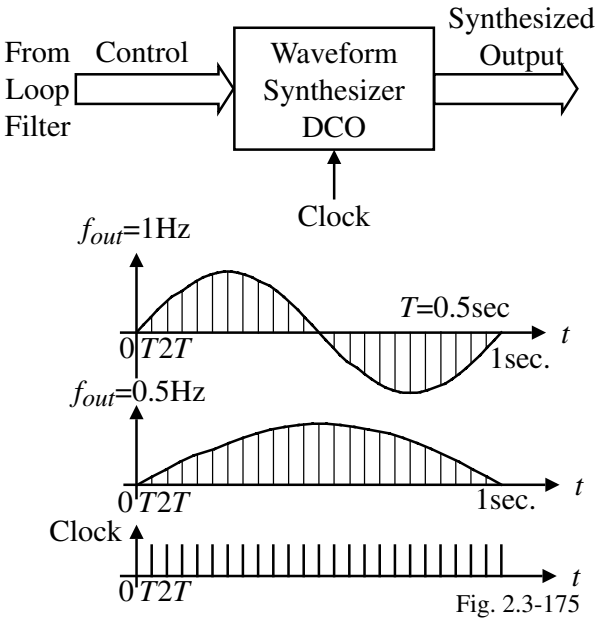


Fig. 2.3-175

EXAMPLES OF ADPLL IMPLEMENTATION

Example 1

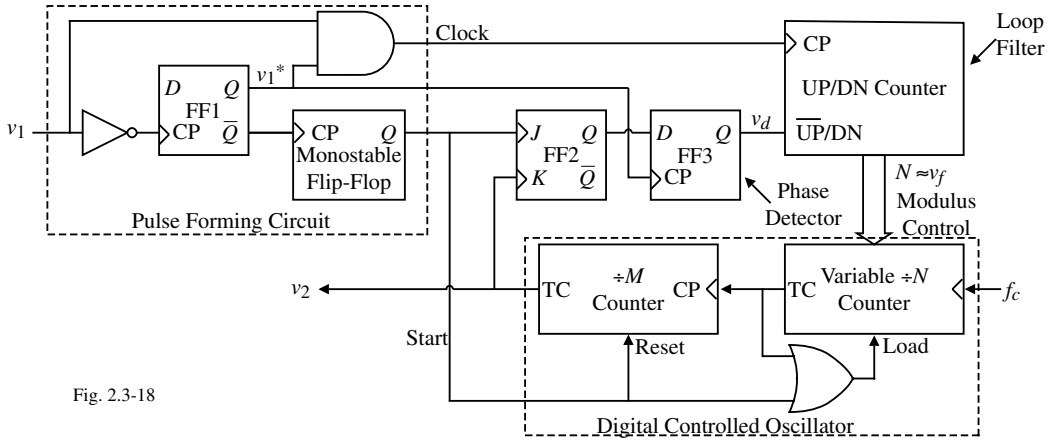


Fig. 2.3-18

Operation:

1.) Pulse forming circuit – Downscales f_1 by two to get v_1*. v_1 and v_1* generate the clock for the loop filter. The negative-going edge of v_1* generates a start pulse.

2.) Digital controlled oscillator – The variable ÷N counter is a down counter. Its content starts with the number N loaded in parallel from the loop filter. The clock, f_c, causes the counter to count down to 0. The content of the ÷N counter at this time is called the terminal count (TC). The output pulse at TC reloads the content N in the ÷N counter and starts the ÷M counter counting up from 0. When the ÷M counter reaches TC, a pulse is delivered at the output which is v_2.

Example 1 – Continued

When the loop is locked, $f_c = MNf_1$. Note that the duration of the start pulse < $1/f_c$.

Waveforms:

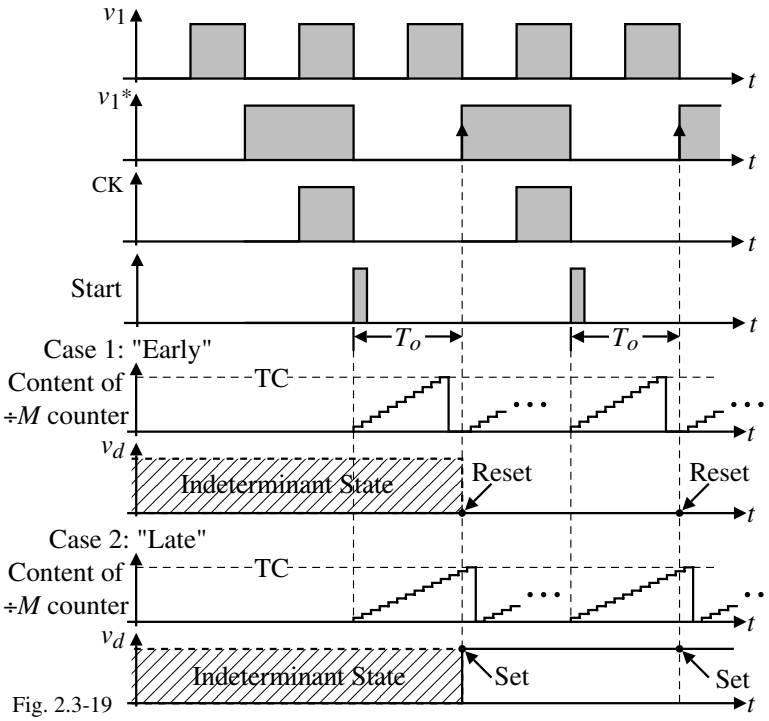


Fig. 2.3-19

Case 1 – "Early": N is too small.

1.) $\div M$ counter reaches TC before T_o.

2.) v_2 causes the loop filter to increase N.

3.) This process continues until the $\div M$ counter reaches TC at the positive edge of v_1*.

Case 2 – "Late": N is too large.

1.) $\div M$ counter reaches TC after T_o.

2.) Under this condition, v_2 causes the loop filter to decrease N.

3.) This process continues until the $\div M$ counter reaches TC at the positive edge of v_1*.

Example 2

Uses the 74xx297

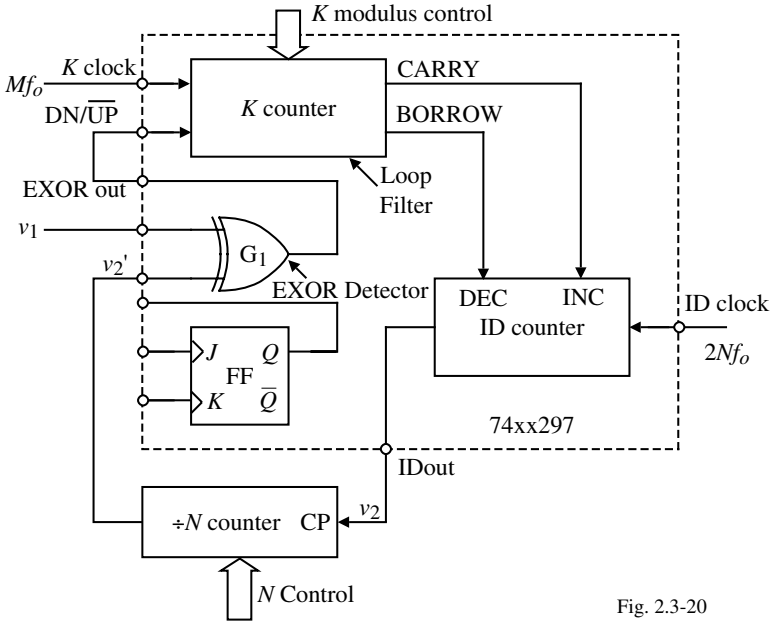


Fig. 2.3-20

In lock, the average number of carry pulses and borrow pulses are equal and no cycles are added or deleted. If f_1 increases, the output of the EXOR detector becomes asymmetrical in order to allow the K counter to produce more carry pulses than borrow pulses on average.

Example 2 – EXOR PD, *M*=16, *K*=4, and *N*=8

Assumptions:

- Loop is in lock
- Both counters count on the negative edges of the *K* clock
- The toggle flip-flop within the ID counter toggles on the positive edge of the ID clock
- All flip-flops of the ÷*N* counter count on the negative edge of the corresponding clock signal

Note that v_2' has a 50% duty cycle which means that it has no ripple or phase jitter.

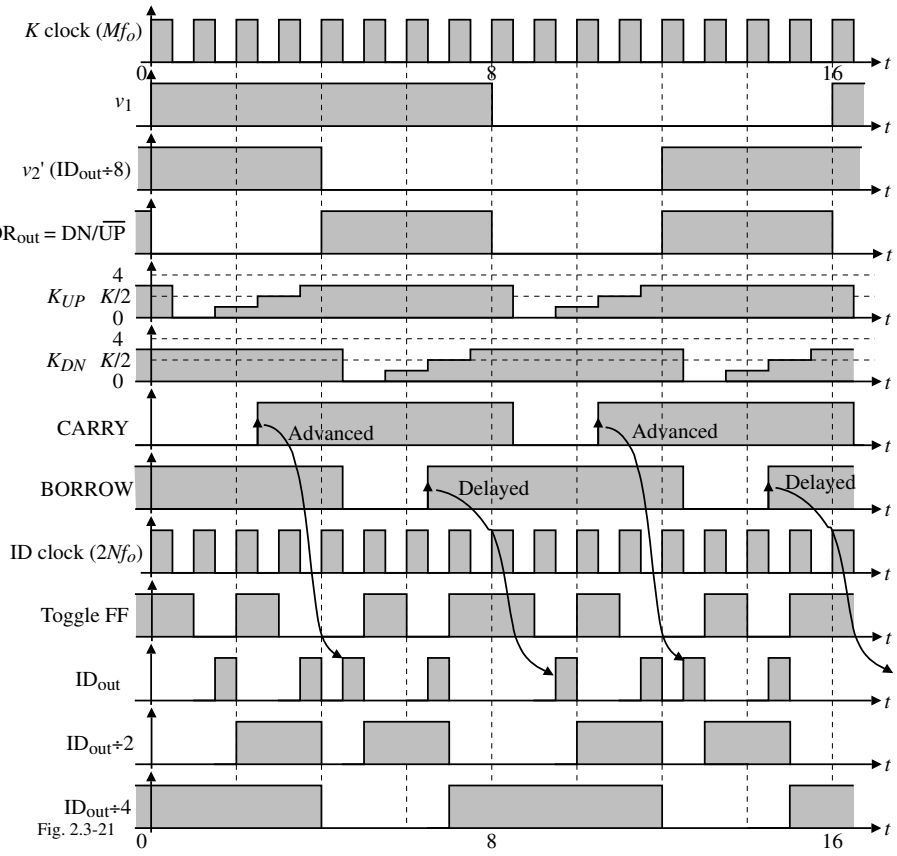


Fig. 2.3-21

Example 2 – EXOR PD, *M*=16, *K*=8, and *N*=8

Waveforms:

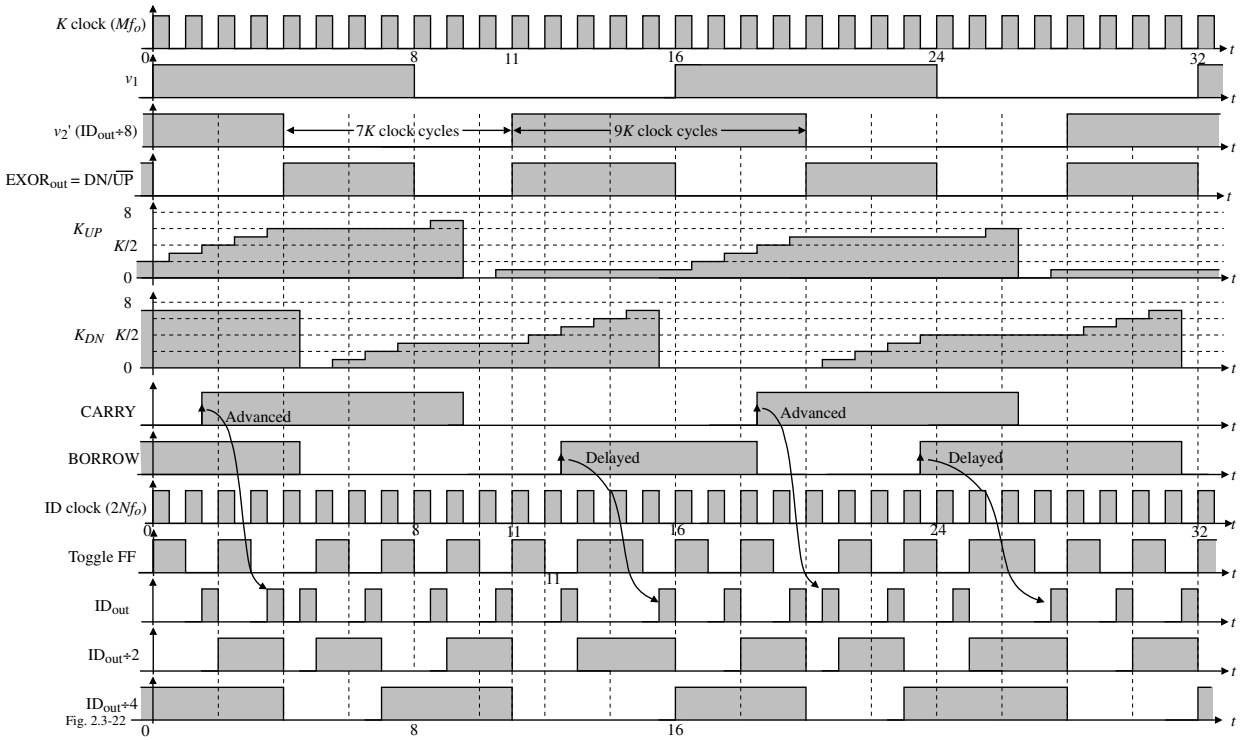


Fig. 2.3-22

If $K \neq M/4$, phase jitter will occur. Duty factor, δ, is $0.5(1-1/N) < \delta < 0.5(1+1/N)$

$\therefore$ maximum deviation is $1/N$ at the worst. Phase jitter can be eliminated.

Example 2 – JK-Flip-flop PD, M=16, K=8, and N=8

Waveforms:

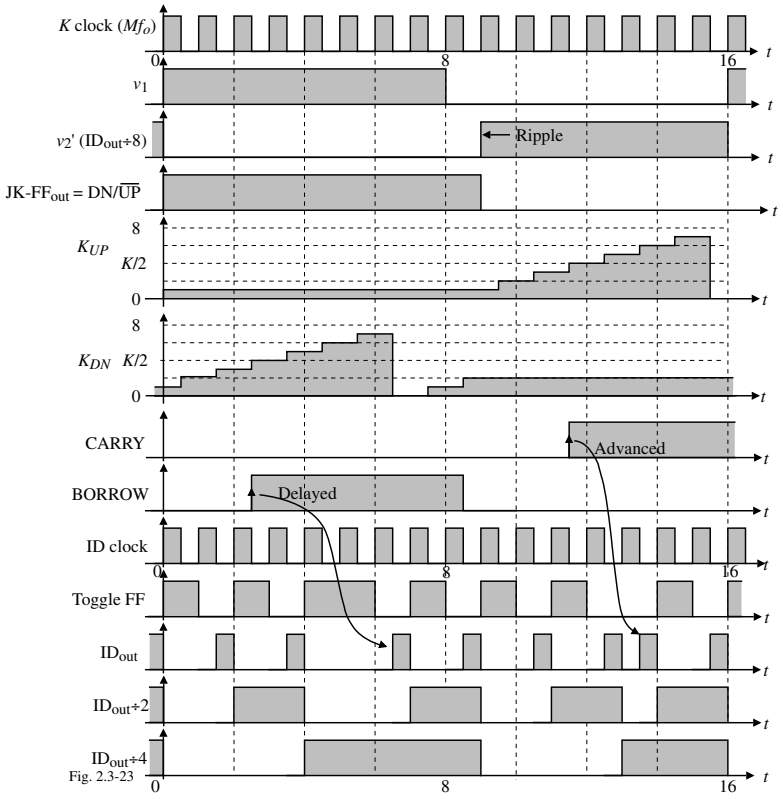


Fig. 2.3-23

Because of the JK-Flip-Flop detector, phase jitter will exist regardless of the value of K.

Duty factor range:

$$0.5\left(1-\frac{M}{2KN}\right) < \delta < 0.5\left(1+\frac{M}{2KN}\right)$$

For minimum ripple, choose

$$K = \frac{M}{2}$$

"Overslept" Carries and Borrows

- If the ID clock frequency is too low, the ID counter is unable to process all the carries and borrows. This condition is called *overslept carries and borrows*.
- If a number of carries have to be processed in succession by the ID counter, the delay between any two carries, K/Mf_o, should be larger than 3 ID clock periods, $1/2Nf_o$.
- The condition for no overslept carries or borrows is given as,

$$\frac{K}{Mf_o} > \frac{3}{2Nf_o} \quad \rightarrow \quad N > \frac{3M}{2K}$$

$$\therefore \qquad N_{min} = \frac{3M}{2K}$$

Since M, K, and N are mostly integer powers of 2, the practical minimum is,

$$N_{practical} = \frac{2M}{K}$$

Hold Range, Δf_H, for the ADPLL

Assume that PD = EXOR, $f_1 = 1.25 f_o$, $M = 16$, $K = 4$, and $N = 8$.

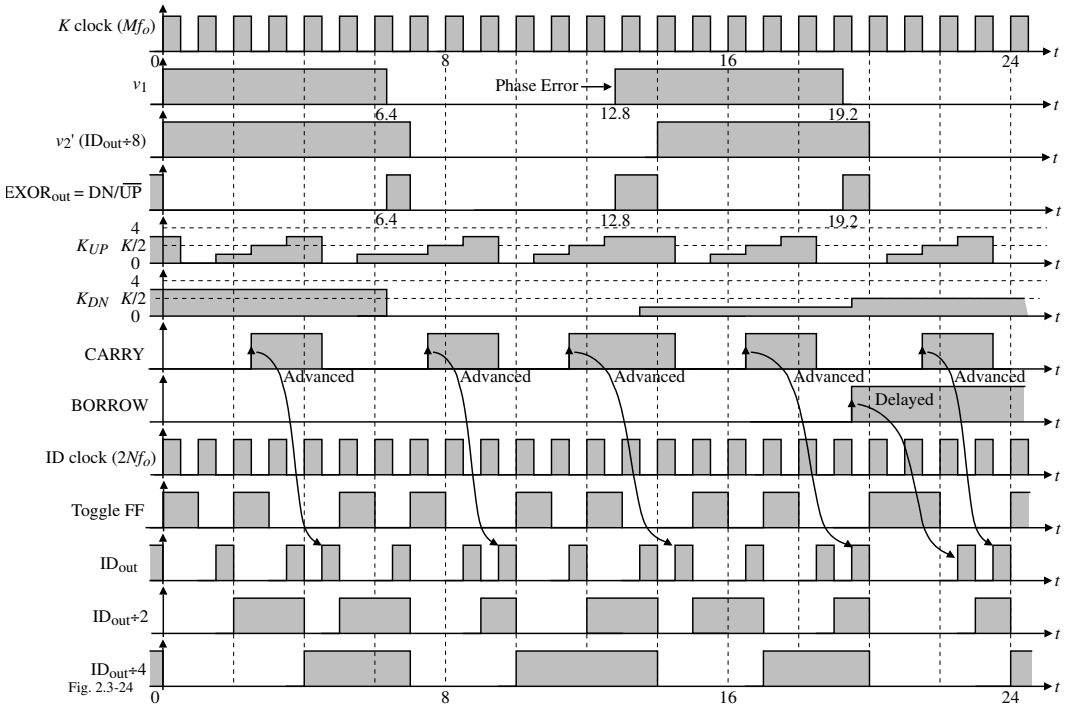


Fig. 2.3-24

The maximum output frequency occurs when the K counter is counting up and is

$$f_{max} = \frac{f_o M}{K}$$

Because each carry applied to the ID counter causes 1/2 cycle to be added to the IDout signal, the output frequency of the ID counter increases by

$$\Delta f_{\text{IDout}} = \frac{f_o M}{2K}$$

Dividing by N gives $\Delta f_H = \dfrac{f_o M}{2KN}$

Frequency Domain Analysis of the ADPLL

Model for the ADPLL:

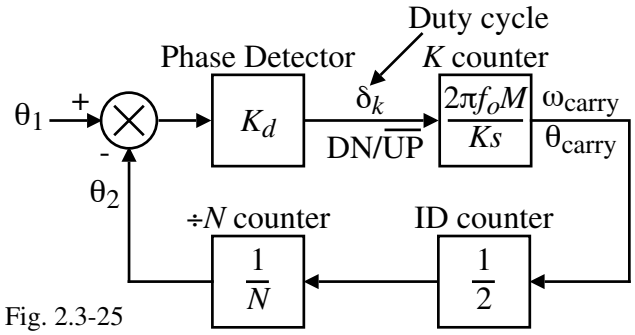


Fig. 2.3-25

$$K_d = \frac{2}{\pi} \text{ (EXOR)}$$

or

$$K_d = \frac{1}{\pi} \text{ (JK-Flip flop)}$$

Model for the K-counter:

$$f_{carry} = \delta_k \frac{Mf_o}{K} \rightarrow \omega_{carry} = 2\pi\delta_k \frac{Mf_o}{K}$$

Transfer function of the K-counter:

$$K_K(s) = \frac{\theta_{carry}(s)}{\Delta_K(s)} = \frac{1}{s}\frac{\omega_{carry}}{\delta_k} = 2\pi\delta_k \frac{Mf_o}{sK}$$

$$\theta_2(s) = \frac{1}{N}\cdot\frac{1}{2}\cdot\frac{2\pi Mf_o}{sK}\cdot K_d[\theta_1(s)-\theta_2(s)] = \frac{K_d \pi Mf_o}{sNK}[\theta_1(s)-\theta_2(s)]$$

$$\theta_2(s) = \frac{\omega_o}{s}[\theta_1(s)-\theta_2(s)] \rightarrow \frac{\theta_2(s)}{\theta_1(s)} = H(s) = \frac{\omega_o}{s+\omega_o}$$

where

$$\omega_o = \frac{K_d \pi Mf_o}{NK} \text{ or } \tau = \frac{1}{\omega_o} = \frac{NK}{K_d \pi Mf_o}$$

Note: $\tau(\text{EXOR}) = \dfrac{NK}{2Mf_o}$ and $\tau(\text{JK}) = \dfrac{NK}{Mf_o}$

Ripple (Phase Jitter) Reduction Techniques

A ripple cancellation scheme that uses the enable feature of the *K* counter is shown below.

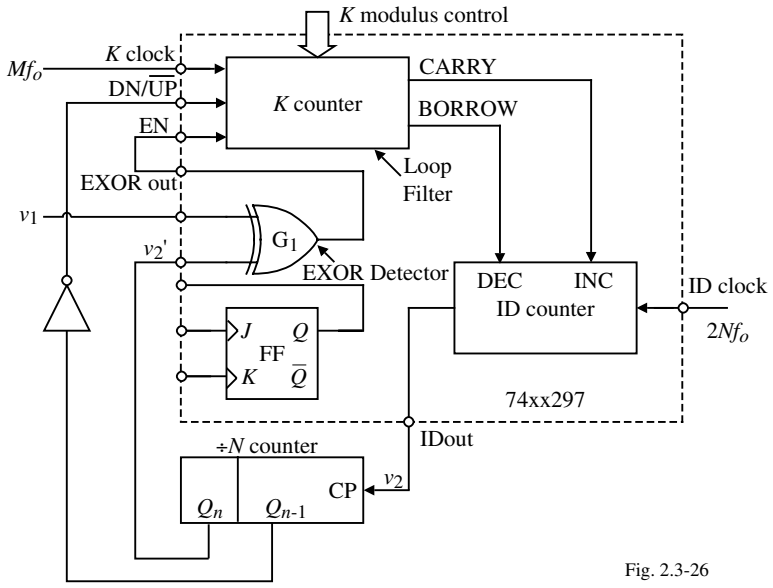


Fig. 2.3-26

DN/$\overline{\text{UP}}$ is driven by Q_{n-1} whose frequency is twice v_2'.

EXOR drives the ENABLE of the *K*-counter

$\therefore$ v_1 and v_2' are nearly in phase when the ADPLL operates at its center frequency.

Ripple Reduction Techniques – Continued

ADPLL operates at its center frequency:

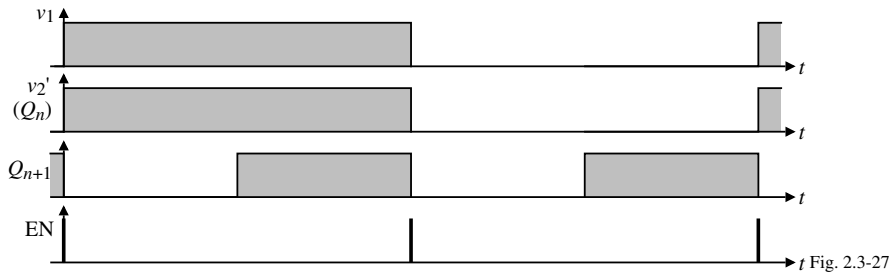


Fig. 2.3-27

The reference frequency > center frequency:

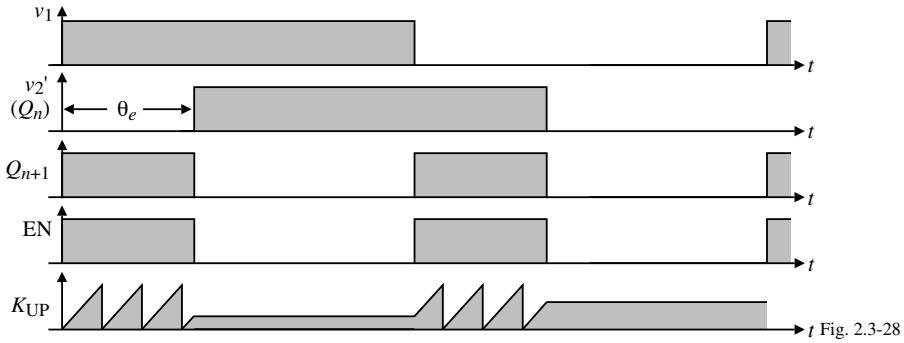


Fig. 2.3-28

The average number of carries is reduced approximately by 2.

$$\therefore \quad \Delta f_H = \frac{Mf_o}{2N(2K+1)} \qquad \text{If } M = 2N \text{ and } K>>1, \text{ then} \qquad \Delta f_H = \frac{f_o}{2K}$$

ADPLL DESIGN

Designing an ADPLL FSK Decoder using the 74xx297

FSK Decoder Diagram:

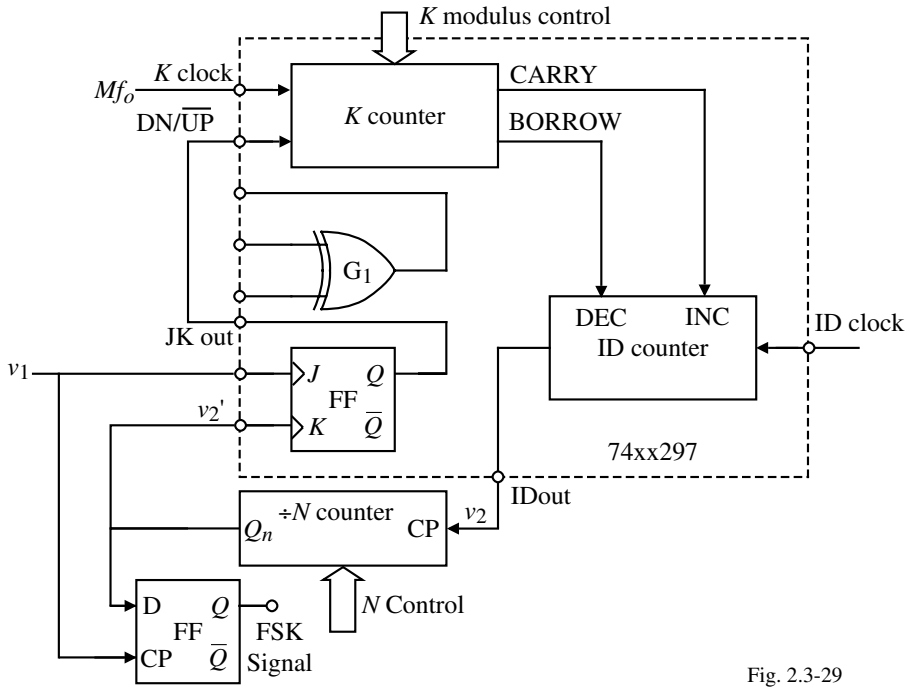


Fig. 2.3-29

ADPLL FSK Decoder Design – Continued

1.) Assume the FSK transmitter uses the frequencies of $f_{11} = 2100\text{Hz}$ and $f_{12} = 2700\text{Hz}$.

Let $f_o = \sqrt{f_{11} f_{12}} \approx 2400\text{Hz}$

To ensure that both frequencies of the FSK transmitter are within the hold range of the ADPLL we specify that $\Delta f_H = 600\text{Hz}$.

2.) The PD has been selected as a JK Flip-flop.

3.) For minimum ripple let $M = 2K$.

4.) Select N.

a.) For the simplest circuit, let $M = 2N$.

If $K = 4$, then $\Delta f_H = \dfrac{Mf_o}{2NK} = \dfrac{f_o}{K} = \dfrac{2400}{4} = 600\text{Hz}$

However, the 74xx297 requires that $K \geq 8$.

b.) Therefore, choose $K = 8$ which gives $\underline{M = 2K = 16}$ and

$$\Delta f_H = \frac{2Kf_o}{2NK} = \frac{f_o}{N} \rightarrow \underline{N = 4}$$

5.) To avoid overslept carries and borrows (this is not realized in this design),

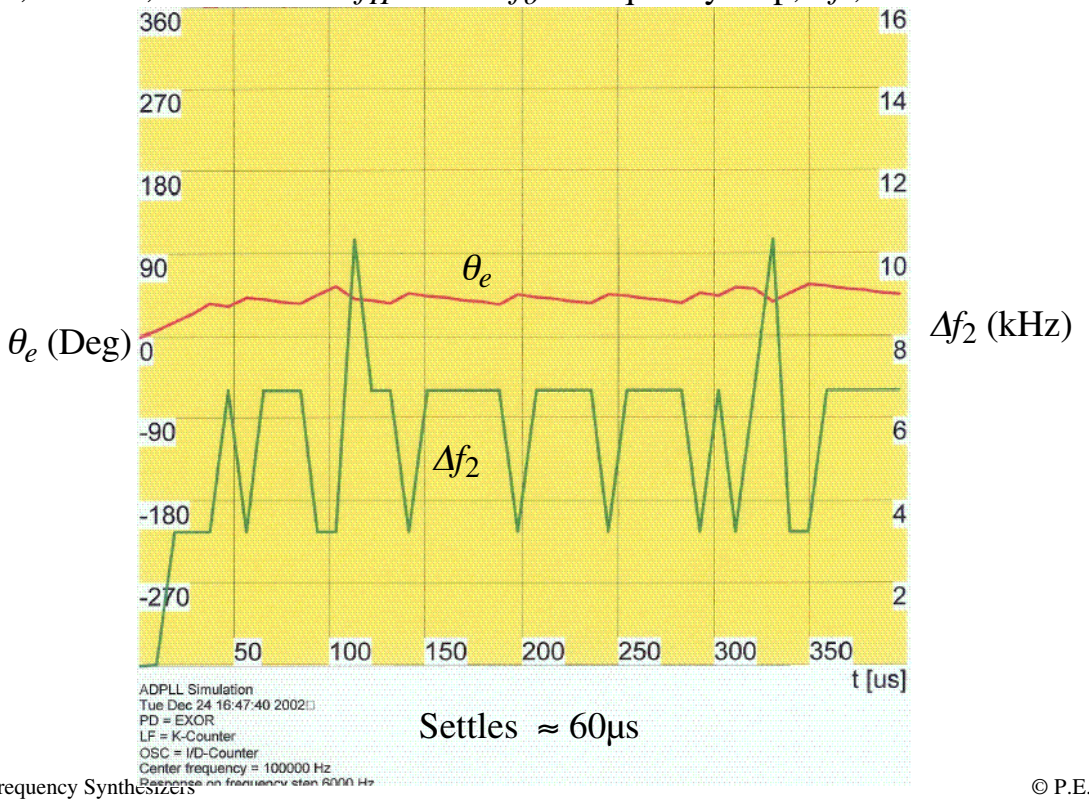
$N > N_{min} = \dfrac{3M}{2K} = \dfrac{3 \cdot 16}{2 \cdot 8} = 3$ Therefore, $N = 4$ is okay.

6.) Settling time, $\tau = \dfrac{2}{f_o} = \dfrac{2}{2400} = 0.833\text{ms}$

ADPLL SYSTEM SIMULATION

Example 1 – Dynamic Performance of the ADPLL using an EXOR PD

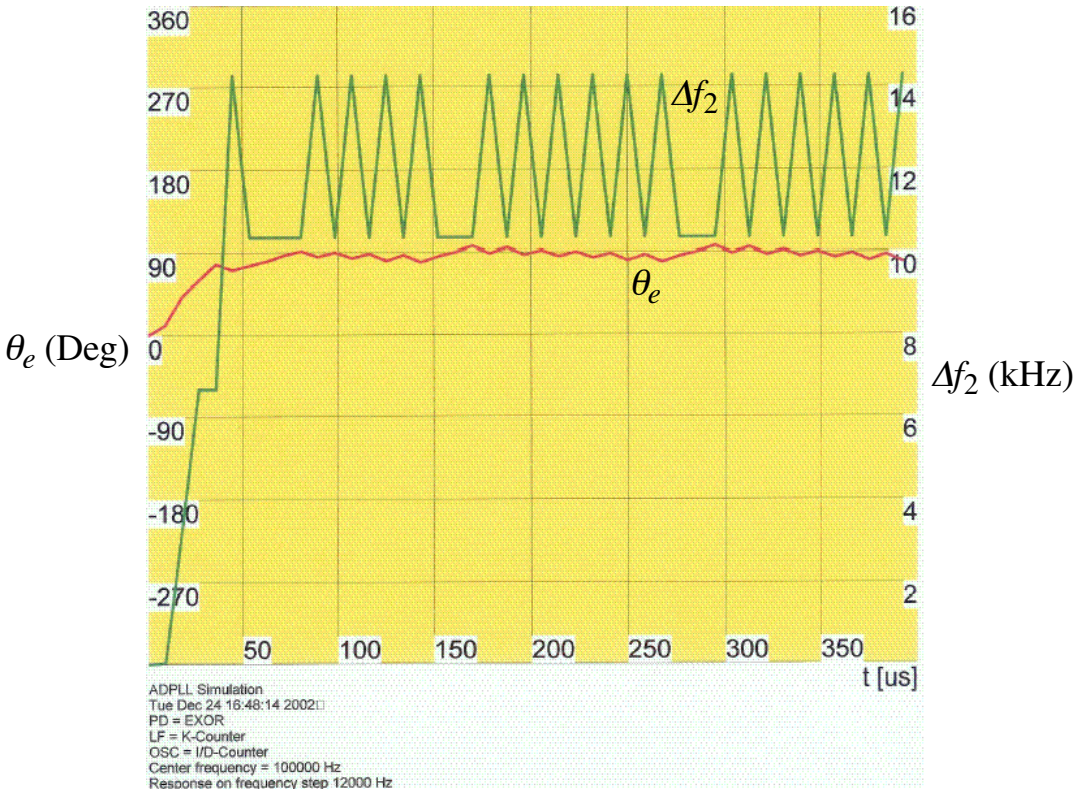
$K = 8$, $M = 32$, $N = 16$ and $\Delta f_H = 0.125 f_o$. Frequency step, Δf, is 6kHz.



Settles ≈ 60μs

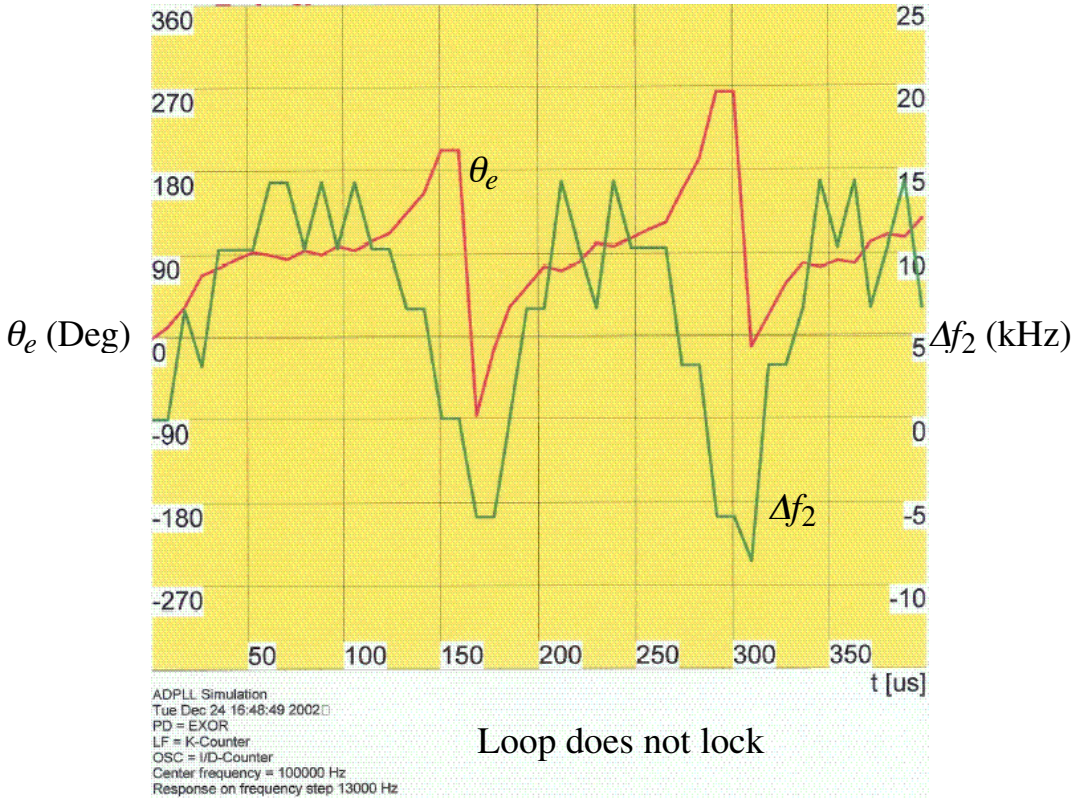
Example 1 – Continued

$\Delta f = 12$ kHz $\Delta f_H = 0.125$x100kHz = 12.5kHz



Example 1 – Continued

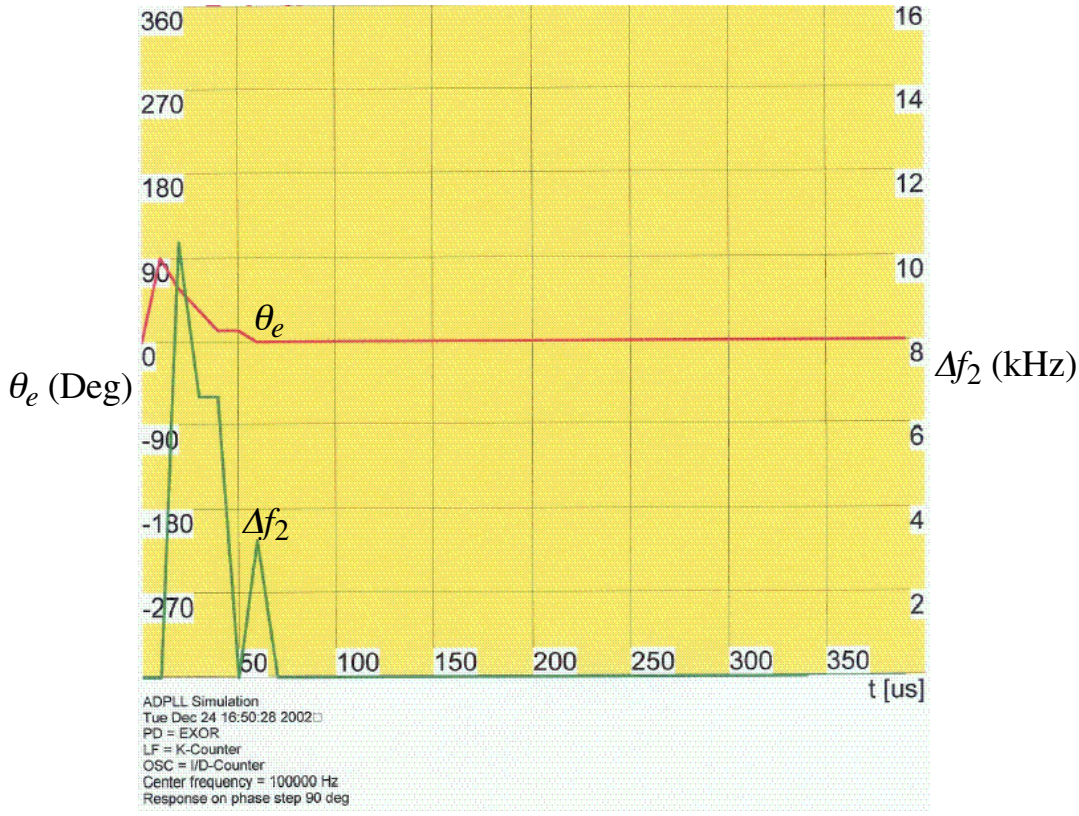
$\Delta f = 13$ kHz $> \Delta f_H$



Loop does not lock

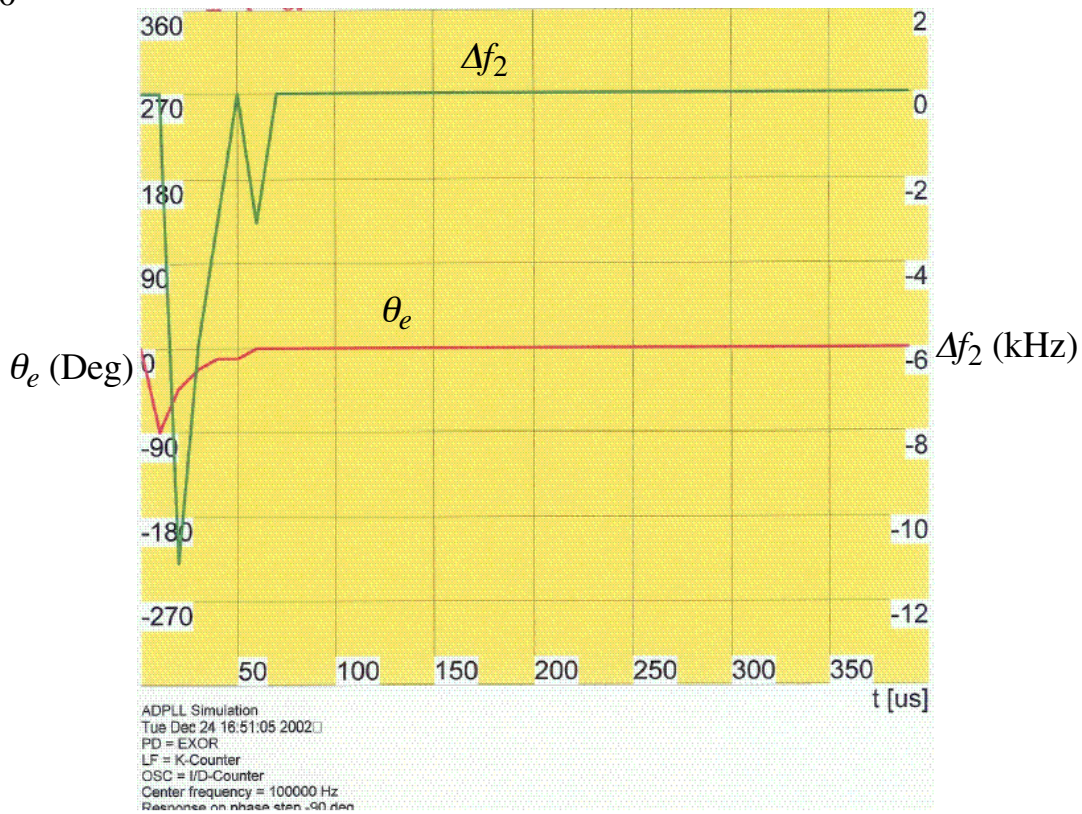
Example 1 – Continued

$\Delta\phi = +90°$



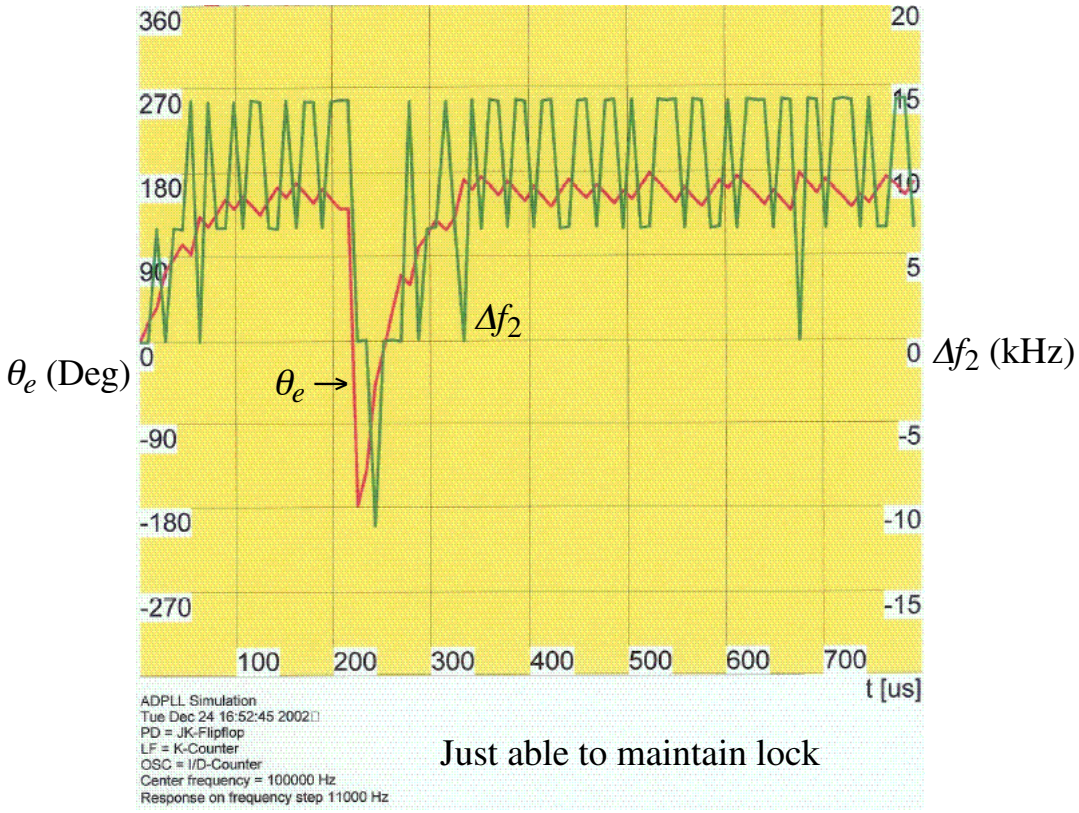
Example 1 – Continued

$\Delta\phi = -90°$



Example 2 – Dynamic Performance of the ADPLL using a JK Flip-flop as the PD

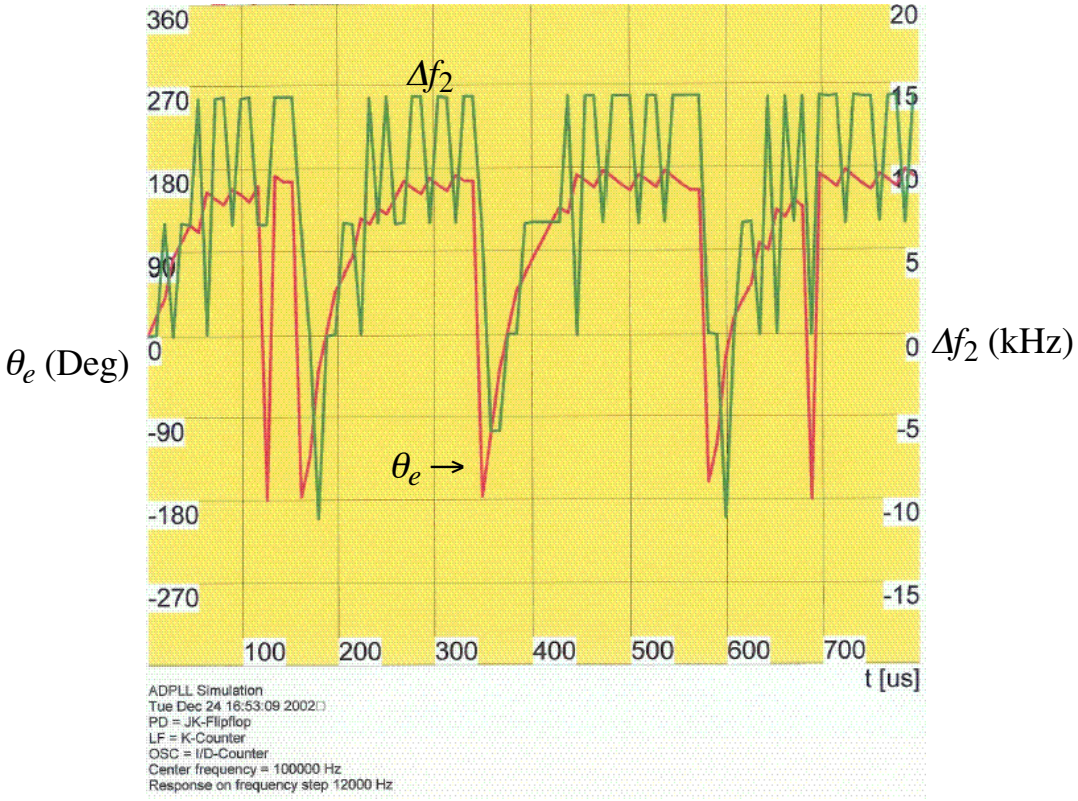
$K = 8,\ M = 16,\ N = 8$ and $\Delta f_H = 12.5$ kHz ($\Delta f = 11$ kHz and $f_o = 100$ kHz)



Just able to maintain lock

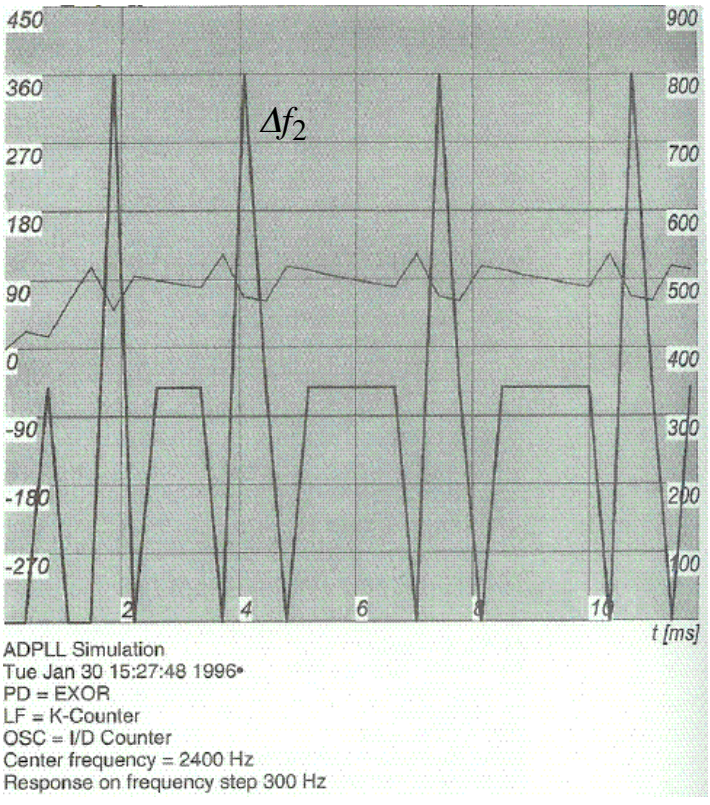
Example 2 – Continued

$\Delta f = 12$ kHz $\rightarrow$ unlocked



Example 3 – FSK Encoder Previously Designed

JK Flip-flop PD, $M = 16$, $K = 8$, and $N = 4$. ($f_o = 2400$Hz and $\Delta f_H = 600$Hz)



SUMMARY

- The ADPLL is implemented entirely of digital circuits
- The digital PDs can have a parallel output or and UP and DOWN output
- Digital VCOs use borrow and carry operations to change the frequency
- This completes our systems perspective of PLLs