Frequency Synthesizer

Frequency Synthesizer (FS)

A frequency synthesizer is a device (an electronic system) that generates a large number of precise frequencies from a single reference frequency.

A frequency synthesizer can replace the expensive array of crystal resonators in a multichannel radio receiver. A single-crystal oscillator provides a reference frequency, and the frequency synthesizer generates the other frequencies. Because they are relatively inexpensive and because they can be easily controlled by digital circuitry, frequency synthesizers are being included in many new communication system designs. Frequency synthesizers are found in many devices, including radio receivers, mobile telephones, radiotelephones, walkie-talkies, satellite receivers, GPS systems, etc.

A frequency synthesizer can combine frequency multiplication, frequency division, and frequency mixing (the frequency mixing process generates sum and difference frequencies) operations to produce the desired output signal.

Direct Frequency Synthesizer

Direct frequency synthesizer is the oldest of the frequency synthesis methods. It synthesizes a specified frequency from one or more reference frequencies from a combination of harmonic generators, band-pass filters, dividers, and frequency mixers.

One method of using a harmonic generator is shown in Figure (1).

![Diagram of Direct Frequency Synthesizer](image)

Figure (1): A direct frequency synthesizer
**Frequency Synthesizer**

The desired frequency is obtained with a filter tuned to the desired output frequency. Highly selective filters are required with this method.

The multiple-oscillator approach is an alternative method. The oscillators are usually easier to realize than the bandpass filters. Figure (2) illustrates a method of generating 99 discrete frequencies from 18 crystal oscillators.

One switch selects one of the nine oscillators that cover the frequency range 1 to 9 kHz in 1-kHz steps, and the other switch covers the frequency range 10 to 90 kHz in 10-kHz steps. The two signals are then combined in a frequency mixer, and the bandpass filter selects the higher of the two mixer output frequencies.

Direct frequency synthesis refers to the generation of new frequencies from one or more reference frequencies by using a combination of multipliers, dividers, bandpass filters, and mixers.

A simple example of direct synthesis is shown in Figure (3).
The new frequency $\frac{2}{3} f_o$ is realized from $f_o$ by using a divide-by-3 circuit, a mixer, and a bandpass filter. In this example $\frac{2}{3} f_o$ has been synthesized by operating directly on $f_o$.

One of the foremost considerations in the design of direct frequency synthesizer is the mixing ratio:

\[ r = \frac{f_1}{f_2} \]

where $f_1$ and $f_2$ are the two input frequencies to the mixer. If the mixing ratio is too large or too small, the two output frequencies will be too close together, and it will be difficult to remove one of the signals with filtering.

**Example 1: If the two mixer input frequencies are 100 and 1MHz, then:**

I. Calculate the mixing ratio.

II. Find the mixer output frequencies.

III. Design a simple direct frequency synthesizer.

**Solution:**

I. 

\[ r = \frac{f_1}{f_2} = \frac{100\text{MHz}}{1\text{MHz}} = 100 \]

II. The mixer output frequencies will be:

$100\text{MHz} + 1\text{MHz} = 101\text{MHz}$ and $100\text{MHz} - 1\text{MHz} = 99\text{MHz}$
III. (H.W.)

The filter requirements can be reduced by using an offset frequency.

Figure (4) illustrates a type of direct synthesis module frequently used in direct frequency synthesizers.

This method is referred to as double-mix-divide. An input frequency $f_i$ is combined with a frequency $f_1$, and the upper frequency $f_i + f_1$ is selected by the bandpass filter. This frequency is then mixed with a switch-selectable frequency $f_2 + f^*$. (In the following $f^*$ refers to any one of 10 switch-selectable frequencies). Frequency $f_2 + f^*$ can be realized with one of the methods illustrated in Figures (1) and (2).

The output of the second mixer consists of the two frequencies $f_i + f_1 + f_2 + f^*$ and $f_i + f_1 - f_2 - f^*$; only the higher frequency appears at the output of the bandpass filter.

If frequencies $f_i, f_1,$ and $f_2$ are selected so that: $10f_i = f_i + f_1 + f_2$, then the frequency at the output of the divide-by-10 module will be:

$$f_o = f_i + \frac{f^*}{10}$$

The double-mix-divide module has increased the input frequency by the switch-selectable frequency increment $f^*/10$. Double-mix-divide modules can be cascaded to form a frequency synthesizer with any degree of resolution. The double-mix-divide
Frequency Synthesizer

The modular approach has the additional advantage that frequencies $f_i$, $f_1$, and $f_2$ can be the same in each module so that all modules can contain identical components.

Considered solely from a theoretical viewpoint, the double-mix-divide module appears unnecessary complicated, since the output frequency $f_i + f^*/10$ could be realized by using one mixer and bandpass filter. The advantages of the approach shown in Figure (4) are practical; it allows better mixing ratios (with relaxed filtering criteria) and allows for the same bandpass filters in each stage.

The output is taken before the last decade divider as this provides a sine wave output. The divider output has a square waveform.

Example 2: Design a direct frequency synthesizer with three digits of resolution, to cover the frequency range of 10 to 19.99 MHz in 10-kHz increments. Specify all frequencies used and the maximum frequency mixing ratio of all mixers.

Solution:

A direct frequency synthesizer with three digits of resolution can be realized using three double-mix-divide modules.

The output is taken before the last decade divider as this provides a sine wave output. The divider output has a square waveform. Therefore, the output of the third module is taken before the decade divider.

It is possible to generate the frequencies between 10 and 19.99 MHz in 10-kHz increments, using the three-module synthesizer, by selecting:

$$f_i = 1\text{MHz} \quad f_1 = 3\text{MHz} \quad f_2 = 6\text{MHz}$$

Since

$$10f_i = f_i + f_1 + f_2$$

The output frequency before the last division by 10 will be:
Frequency Synthesizer

\[ f_o = 10f_i + f_i' + \frac{f_2'}{10} + \frac{f_i'}{100} \]

Since \( f' \) occurs in 1-MHz increments, \( f_i'/100 \) will provide the desired 10-kHz frequency increments. A direct frequency synthesizer with three digits of resolution is shown in Figure (5).

Homework 1: Design a direct frequency synthesizer, to generate 14.86 MHz from 1MHz reference oscillator. Specify all frequencies used and the maximum frequency mixing ratio of all mixers.

Answer: \( f_1' = 6\text{MHz} \quad f_2' = 8\text{MHz} \quad f_3' = 4\text{MHz} \)
The disadvantages associated with direct synthesis are greatly diminished with the frequency synthesis technique (often referred to as indirect synthesis) that employs a phase-locked loop. A simple PLL frequency synthesizer is illustrated in Figure (6).

Figure (6): An indirect frequency synthesizer.

Figure (6) illustrates the basic architecture of the PLL frequency synthesizer. The phase detector generates an output signal that is a function of the difference between the phases of the two input signals. The detector output is filtered (and perhaps amplified), and the dc component of the error signal is applied to the voltage-controlled oscillator. The signal, fed back to the phase detector, is the VCO output frequency divided by N. The VCO control voltage forces the VCO to change frequency in the direction that reduces the difference between the input frequency and the divider output frequency. If the two-phase detector input frequencies are sufficiently close, the PLL feedback mechanism forces the two-phase detector input frequencies to be equal. That is,

\[ f_r = f_d \]
Frequency Synthesizer

And the divider output frequency \( f_d \) is obtained by dividing the voltage-controlled oscillator (VCO) output frequency \( f_o \) by \( N \):

\[
f_d = \frac{f_o}{N}
\]

Therefore, the output frequency \( f_o \) is an integer multiple of the reference frequency, or:

\[
f_o = Nf_r
\]

The PLL with a frequency divider in the loop thus provides a method for obtaining a large number of frequencies from a single reference frequency. If the divide ratio \( N \) is realized by using a programmable divider (integrated circuitry has made the digital programmable divider an inexpensive circuit component), it is possible to easily change the output frequency in increments of \( f_r \). The PLL with a programmable divider provides an easy method for synchronizing a large number of frequencies, all of which are an integer multiple of the reference frequency. Frequency synthesis is a major application of PLLs.

From \( f_o = Nf_r \), we note that the frequency resolution is equal to \( f_r \). That is, the output frequency can be changed in increments as small as \( f_r \); however, this is in conflict with the requirement of a short time interval for changing frequencies. Although an exact expression for the switching time has yet to be derived, a frequently used rule of thumb is that the switching time:

\[
t_s = \frac{25}{f_r}
\]

It takes approximately 25 reference periods to switch frequencies. The frequency resolution is therefore inversely proportional to the switching speed. A contemporary specification for satellite communication systems, which use the frequency hopping is that the reference resolution is equal to 10 Hz and the switching time is less than
Frequency Synthesizer

10\mu s! Since the above rule of thumb predicts a switching time of 2.5s, it is clear that the simple PLL frequency synthesizer cannot meet both specifications.

The choice of reference frequency dominates loop performance.

Effects of Reference Frequency on Loop Performance

The expression for the output frequency \[ f_o = N f_r \] shows that, to obtain fine frequency resolution, the reference frequency must be small. This creates conflicting requirements.

One problem is that to cover a broad frequency range requires a large variation in \( N \). Even if the hardware problems can be overcome, some method will normally be needed to compensate for the variation in loop dynamics that occur for widely varying values of \( N \).

It is shown that, the linearized loop transfer function is:

\[
H(s) = \frac{\Theta(s)}{\Phi(s)} = \frac{f_o(s)}{f_i(s)} = \frac{G F(s)}{s + G F(s)/N}
\]

where \( F(s) \) is the transfer function of the low-pass filter. If \( N \) is to assume a large number of values, say, from 1 to 1000, then there will be a 60-dB variation in the open-loop gain and a correspondingly wide variation in the loop dynamics, unless some method (such as the use of a programmable amplifier) is employed to alter the loop gain for different \( N \) values.

A second problem encountered with a low reference frequency is that the loop bandwidth must be less than or equal to the reference frequency, because the low-pass filter must filter out the reference frequency and its harmonics present at the phase-detector output. Thus, the filter bandwidth must be less than the reference frequency. The loop bandwidth is normally less than the filter bandwidth for adequate stability. Therefore, a low reference frequency results in a frequency synthesizer that will be slow to change frequency.
Frequency Synthesizer

Another problem introduced by a low reference frequency is the effect on noise introduced in the VCO. Figure (7) shows a linearized model of a PLL with the three main sources of noise. Here, $\phi_{N_r}$ is the noise on the reference signal, and $\phi_{N_d}$ is the noise created in the phase detector. The largest phase-detector noise components are at the reference frequency and the harmonics of this frequency. And $\phi_{N_o}$ is the noise introduced by the VCO.

![Figure (7): A PLL synthesizer including three noise sources.](image)

The total noise of the closed-loop system at the VCO output $\theta_N$ is given by (using superposition theorem):

$$\theta_N = \frac{(\phi_{N_r} + \phi_{N_d}) G F(s)/s}{1 + G F(s)/Ns} + \frac{\phi_{N_o}}{1 + G F(s)/Ns}$$

$$= G(s)(\phi_{N_r} + \phi_{N_d}) + G_r(s)\phi_{N_o}$$

Since $F(s)$ is either unity or a low-pass transfer function, $G(s)$ is a low-pass transfer function and $G_r(s)$ is a high-pass transfer function. The PLL functions as a low-pass filter for phase noise arising in the reference signal and phase detector, and it functions as a high-pass filter for phase noise originating in the VCO. Since the VCO noise is a low-frequency noise, the output noise due to $\phi_{N_o}$ is minimized by having the loop bandwidth as wide as possible. At the same time, the loop bandwidth should
Frequency Synthesizer

be less than the reference frequency in order to minimize the effect of $\phi_{Nd}$, which is dominated by spurious frequency components at the reference frequency and its harmonics.

Therefore, the desire to have a low reference frequency $f_r$ in order to obtain fine frequency resolution is offset by the need to have $f_r$ large in order to reduce the loop settling time and to minimize the noise contributed by the VCO.

Variable-Modulus Dividers

Another difficulty with the system illustrated in Figure (6) is that the maximum operating speed of programmable dividers is slower than that required in many communication systems. The upper limit of a programmable divider realized from transistor-transistor logic (TLL) components is approximately 25MHz, and that realized with complementary-symmetry metal-oxide semiconductor (CMOS) logic is about 4MHz. So, for example, if one is to build a $2 \times 10^9$ Hz synthesizer for satellite communications, some other method must be used. There are various ways to overcome this problem. First we discuss the problem of the relatively low operating speed of programmable dividers.

Programmable dividers are slower than fixed-modulus dividers (prescaler). In fact, prescaler are available that operate at gigahertz frequencies. Figure (8) illustrates an indirect synthesizer that contains both a prescaler and a programmable divider in the loop. The prescaler, which can operate frequencies into the gigahertz region, first reduces the output frequency by the factor $P$ before it is applied to the programmable divider. When the loop is in lock:

$$f_r = \frac{f_o}{PN} \quad \text{or} \quad f_o = N(Pf_r)$$

Although the use of prescaler allows the loop to operate with higher output frequencies, the output frequency can be changed only in increments of $Pf_r$. Since the
Frequency Synthesizer

channel spacing is equal to the reference frequency, in order to obtain the same resolution, the reference frequency must be decreased by the prescaler factor P.

![Frequency Synthesizer diagram](image)

**Figure (8): A PLL including a prescaler.**

Another approach, for obtaining good frequency resolution while operating at high output frequencies, uses a method known as **variable-modulus prescaling**. Reconsidering \( f_o = N(f_r) \) , we see that the output frequency resolution could be improved if the value of N were an integer plus a fraction. For example, if \( N = N_0 + AQ/P \) (where A and Q are integers), then the output frequency will be given by:

\[
f_o = P \left( N_0 + \frac{AQ}{P} \right) f_r = PN_0 f_r + AQ f_r
\]

And the resolution can be regained. This equation is not easily implemented, but if \( \pm AP \) is added, the result is:

\[
f_o = [PN_0 + AQ - AP + AP] f_r = [P(N_0 - A) + (P + Q)A] f_r
\]

From this Equation, it is apparent that a dual-modulus counter that divides by \( P + Q \) for A cycles and by P for \( N_0 - A \) cycles could be used to implement the function.

The dual-modulus prescaler system illustrated in Figure (9) has a prescaler that divides by the modulus \( P + Q \) when the modulus control is high and divides by P
**Frequency Synthesizer**

when the modulus control is low. In this particular scheme, the output of the variable-modulus prescaler simultaneously drives the two programmable dividers 1 and 2. The programmable dividers operate at the input clock rate $f_i$ divided by $P$ or $P + Q$. The divide cycle begins with counter 1 preset to $A$, counter 2 preset to $N$, and the modulus control high so that the two-modulus prescaler output frequency is equal to the frequency divided by $P + Q$. The prescaler will divide-by-$P + Q$ until the $A$ counter reaches 0. At this point, the divide-by-$N$ counter is at a value equal to $N\text{(preset)} - A\text{(preset)}$. Next, counter $A$ pulses the prescaler modulus control to low to change to the divide-by-$P$ mode. The prescaler then divides by $P$, $N-A$ times, until the $N$ counter reaches 0.

![Diagram of programmable divider with dual-modulus prescaler]

**Figure (9):** A programmable divider realized with a dual-modulus prescaler.
Frequency Synthesizer

Finally, the divide cycle is restated by reloading the counters with their preset values and resetting the modulus control signal. The number of input cycles in one complete divide cycle is:

\[ D = [P(N_o - A) + (P + Q)A] = AQ + PN_o \]

Note that, \( N \) must be greater than \( A \) for the method to work.

If \( Q = 1 \), then the divide ratio, even though it has a minimum value \( D_{min} = PN_o \), can be implemented in unit steps.

Example 3: Design a phase-locked loop synthesizer, to cover the frequency range from 100 to 109MHz in 1-MHz increments. The 100MHz is too high for a programmable divider, and a 10/11 variable-modulus prescaler should be used. Find the minimum divide ratio.

Solution:

\( P = 10 \) and \( Q = 1 \)

A reference frequency of 1MHz is suitable (a higher frequency would not be).

\((f_o)_{min} = 100\text{MHz} \quad \text{and} \quad (f_o)_{max} = 109\text{MHz}\)

For \((f_o)_{max} = 109\text{MHz}\), the value of \( N \) and \( A \) are found, as follows:

First, let \( A = 0 \), and solve for \(( N_o)_{min}\):

\[ P(N_o)_{min} + AQ = \frac{(f_o)_{max}}{f_r} \]

For \( P = 10 \), \( Q = 1 \), and \( A = 0 \)

\[ 10(N_o)_{min} = \frac{109\text{MHz}}{1\text{MHz}} = 109 \quad \therefore (N_o)_{min} = \frac{109}{10} = 10 \quad \text{(integer)} \]

So \( N_o \) must be at least 10.

Second, for this value of \( N_o \), find \( A_{\text{max}} \) by:
Frequency Synthesizer

\[ A_{\text{max}} = \frac{(f_o)_{\text{max}}}{f_r} - PN_o \quad (Q = 1) \]

\[ A_{\text{max}} = \frac{109\text{MHz}}{1\text{MHz}} - (10)(10) = 109 - 100 = 9 \]

Now \( A \) will vary from 0 to 9.

The minimum divide ratio \( D_{\text{min}} = \frac{(f_o)_{\text{min}}}{f_r} = PN_o = (10)(10) = 100. \)

Thus, the desired division ratio can be obtained by using 10/11 variable-modulus prescaler together with the programmable dividers.

Other variable-modulus division ratios such as 5/6, 8/9, 32/33, 40/41, 64/65, 100/101, and 128/129 are also frequently used.

Homework 2: Could a 100/101 variable-modulus divider be used in Example 3? Explain your answer.

Example 4: Design a phase-locked loop synthesizer, to cover the frequency range from 100 to 100.99MHz in 10-KHz increments. The 100MHz is too high for a programmable divider, and a 100/101 variable-modulus prescaler should be used. Find the minimum divide ratio.

Solution:

\( P = 100 \) and \( Q = 1 \)

A reference frequency of 10-KHz is suitable (a higher frequency would not be).

\( (f_o)_{\text{min}} = 100\text{MHz} \quad \text{and} \quad (f_o)_{\text{max}} = 100.99\text{MHz} \)

For \( (f_o)_{\text{max}} = 100.99\text{MHz} \), the value of \( N \) and \( A \) are found, as follows:

First, let \( A = 0 \), and solve for \( (N_o)_{\text{min}} \):
Frequency Synthesizer

$$P(N_0)_{\text{min}} + AQ = \frac{(f_o)_{\text{max}}}{f_r}$$

For $P = 100$, $Q = 1$, and $A = 0$

$$100(N_0)_{\text{min}} = \frac{100.99\text{MHz}}{10\text{KHz}} = 10099 \quad \therefore (N_0)_{\text{min}} = \frac{10099}{100} = 100(\text{integer})$$

So $N_0$ must be at least 100.

Second, for this value of $N_0$, find $A_{\text{max}}$ by:

$$A_{\text{max}} = \frac{(f_o)_{\text{max}}}{f_r} - PN_0 \quad (Q = 1)$$

$$A_{\text{max}} = \frac{100.99\text{MHz}}{10\text{KHz}} - (100)(100) = 10099 - 10000 = 99$$

Now $A$ will vary from 0 to 99.

The minimum divide ratio $= \frac{(f_o)_{\text{min}}}{f_r} = PN_0 = (100)(100) = 10^4$.

Thus, the desired division ratio can be obtained by using 100/101 variable-modulus prescaler together with the programmable dividers.

**Homework 3:** Could a 10/11 variable-modulus divider be used in Example 4? Explain your answer.

**Down Conversion**

Another approach to circumventing the high-frequency limitation of the programmable dividers is to shift the output frequency down by mixing the output frequency with a local oscillator frequency. Figure (10) illustrates a single down-conversion synthesizer. The low-pass filter following the mixer is used to filter out the higher mixer output frequency $f_o + f_L$. The divider output frequency is:

$$f_d = f_r = \frac{f_o - f_L}{N}$$
So,

\[ f_o = f_L + Nf_r \]

Figure (10): A PLL frequency synthesizer with single down-conversion

The main disadvantages of this method are:

I. The complexity and size are increased.

II. The possibility of spurious components being introduced by the mixer is increased.

III. The phase lag of the filter used in the feedback path can degrade the loop performance.

Methods for Reducing Switching Time and/or Widening the Loop Bandwidth

There are methods available for circumventing the conflict between the need for fine frequency resolution and the need to quickly change frequencies.

A method of reducing the response time is to include a coarse steering signal. When the frequency is changed by altering the divide ratio N, a steering signal can be generated and applied immediately to direct the VCO to the new frequency (See
**Frequency Synthesizer**

Figure (11)). The steering signal can be obtained from a lookup table stored in memory with the D/A converter used to generate the analog steering signal.

![Diagram of Frequency Synthesizer](image)

**Figure (11):** Coarse steering can be used to reduce PLL switching time.

Another frequently used method is to incorporate multiple phase-locked loops in the synthesizer.

**Multiple-Loop Frequency Synthesizer**

One possible method of obtaining fine frequency resolution with a higher reference frequency is illustrated in Figure (12). The output frequency is obtained by dividing the VCO output frequency by \( M \). That is,

\[
f_o = \frac{Nf_r}{M}
\]

And the frequency resolution is:

\[
f = \frac{f_r}{M}
\]
**Frequency Synthesizer**

Hence fine resolution is obtained by making sure that M is sufficiently large. A problem inherent in this technique is that the loop frequency may become too large. The difficulty is illustrated by the following numerical example.

![PLL frequency synthesizer diagram](image)

Figure (12): A PLL frequency synthesizer with a postdivider for increased frequency resolution

**Example 5:** Consider the design of a frequency synthesizer, to cover the frequency range from 10 to 10.1MHz with 1-KHz resolution. The reference frequency is to be 100KHz. To obtain the 1-KHz frequency resolution from the 100KHz reference frequency requires that the VCO output frequency be divided by 100. An output frequency of 10MHz will require that the VCO be operating at 1GHz!

Although adding a post-divider is not a good solution in general, the concept does find practical application in multiple-loop synthesizers. A multiple-loop synthesizer uses one or more loops to obtain the fine frequency resolution and combines the outputs of these loops with that of another loop which generates the higher-frequency components of the desired output frequency. The principles involved can be easily understood by examining one such synthesizer.

**Example 6:** Design a three-loop frequency synthesizer, to cover the frequency range from 35.40 to 40.00MHz in 1-KHz increments. The reference frequency is to be 100KHz.
Frequency Synthesizer

Solution:

Figure (13) shows the three-loop frequency synthesizer discussed in Example (6). The synthesizer consists of three PLLs. PLLs A and B both use the 100-KHz reference frequency. Loop C locks the divided output of loop A \( f_A \) to the difference between the output frequency and the output of loop B \( f_B \). That is,

\[
f_A = f_o - f_B
\]

Or,

\[
f_o = f_A + f_B
\]

Phase-locked loop C serves as a mixer and filter for \( f_A \) and \( f_B \). If \( f_A \) and \( f_B \) are directly combined in a mixer, the sum and difference frequencies will be too close together to be adequately separated with a band-pass filter. The present technique of using a phase-locked loop for frequency mixing does accomplish good separation.

Since the reference frequency of loop A is 100-KHz, its output frequency \( f_a \) can be varied in 1-KHz increments, and:

\[
f_A = \frac{f_a}{100} = N_A \times 10^3
\]

varies in 1-KHz increments. Loop A is used to generate the 1- to 10-KHz increments of output frequency, and loop B the 0.1- and 1-MHz changes in output frequency, but \( f_A \) also serves as the reference frequency for loop C. If \( f_A = 1\)KHz, this will require that loop C be a relatively slow loop and will determine the overall response time of the synthesizer. To reduce the response time of loop C, \( f_A \) is increased by 300-KHz so that:

\[
300\text{KHz} \leq f_A \leq 399\text{KHz}
\]

Therefore,

\[
3 \times 10^7 \leq f_a \leq 3.99 \times 10^7
\]
And $300 \leq N_A \leq 399$

Since $f_B = f_o - f_A$, frequency $f_B$ is reduced by 300-KHz so that:

$35.40 - 0.3 \leq f_B \leq 40 - 0.3$MHz

And

$351 \leq N_B \leq 397$

The response time of the frequency synthesizer is determined by the response times of loops A and B, both of which have a reference frequency of 100-KHz. Hence the overall response time will be approximately $25 \times 10^{-2}$ms, even though 1-KHz frequency increments are obtained.

Figure (13): the three-loop frequency synthesizer discussed in Example 6.
Frequency Synthesizer

Accumulator

The accumulator serves as the phase generator and determines the resolution, frequency, and spurious signal formations.

The two most common accumulators are the binary type and the decimal (BCD).

Binary Accumulators

The binary accumulator is a digital integrator, performing the arithmetic function:

\[ S(n) = S(n - 1) + K \]

Where \( S(n) \) is an N-bit word which is the accumulator output at clock tick \( n \) and \( K \) is the input control.

The accumulator is usually constructed by adders and registers, as shown in Figure (14). The register is a storage device which changes its output only when clocked.

![Accumulator block diagram](image)

**Figure (14): Accumulator block diagram.**
Fractional-N Loop

An alternative method of decreasing loop response time would be possible if $N$ could be made to take on fractional values. The output frequency could then be changed in fractional increments of the reference frequency. Although a digital divider cannot provide a fractional division ratio, ways can be found to effectively accomplish the same task. The technique was originally called digiphase, and a commercial version is referred to as Fractional-N. The most frequently used method to divide the output frequency by $N + 1$ every $M$ cycles and to divide by $N$ the rest of the time. The effective division ratio is then $N + M^{-1}$, and the average output frequency is given by:

$$f_o = (N + M^{-1})f_r$$

This expression shows that, $f_o$ can be varied in fractional increments of the reference frequency by varying $M$. A simplified method for generating the fractional division is shown in Figure (15). The divider divides the input frequency by $N$, and the counter counts the number of cycles of waveform output. Each time the counter reaches a count of $M - 1$, the counter output goes low for one input cycle and one input cycle does not reach the divider. Therefore, the divider requires $N + 1$ input cycles to change state.

The number of output cycles during one complete cycle of the $M$ counter is:

$$f_o = f_d N(M - 1) + f_d (N + 1) = f_d (NM + 1)$$

And the average output frequency per cycle of the $M$ counter is:

$$\langle f_o \rangle_{av} = (N + M^{-1})f_d$$

The average frequency at the divider output is the output frequency divided by $N + M^{-1}$, so a form of fractional division has been realized. The method of implementing fractional-N division shown in Figure (15) will work as long as $M$ is an integer, but normally it will not be. A more general method of implementing fractional division.
Frequency Synthesizer

can be obtained by using a phase accumulator. The phase accumulator approach is illustrated by the following example.

![Diagram](image)

**Figure (15):** A simplified method of implementing fractional division

**Example 7:** Consider the problem of generating 455 KHz by using a fractional-N loop with a 100-KHz reference frequency. The integer part of the division is $N = 4$, and the fractional part is $M^{-1} = 0.55$, or $M = 1.8$. Clearly $M$ is not an integer; the VCO output is to be divided by $5(N + 1)$ every 1.8 cycles, or 55 times every 100 cycles. Although $M$ is not an integer, the fractional can be easily implemented by adding $M^{-1} = 0.55$ to the contents of an accumulator every output cycle. Each time the accumulator overflow (the contents exceed 1); the divider divides by 5 rather than by 4. Only the fractional value of the addition is retained in the phase accumulator.

The phase accumulator realization of a fractional division is illustrated in Figure (16). Fine frequency resolution can be arbitrary obtained by increasing the length of the phase accumulator. The current example used a 100-KHz reference frequency. A resolution of $10^5/10^5 = 1$Hz could be obtained by using a five-stage binary-coded decimal (BCD) accumulator.
The performance of a fractional-N synthesizer will be further illustrated with another numerical example.

Example 8: consider the problem of incrementing the output frequency of 1-MHz synthesizer by 1000Hz, the reference frequency being 10 KHz. Since

\[ f_o = (N + M^{-1})f_r \]

N = 100; if \( f_o \) is to be increased to \( 1.001 \times 10^6 \)Hz, then \( M = 10 \). That is, at every 10 reference cycle (1000 output cycles), the output frequency is divided by 101. The average output frequency is then \( 100.1 \times 10^4 \)Hz, which is the desired frequency. While the reference signal goes through one period, the VCO signal goes through 100.1 cycles and the output of the divider (÷100) goes through 1.001 cycles; its phase relative to the reference frequency advances by \( 0.001 \times 2\pi \) rad each reference cycle. After the 10 reference cycles, the divider reference output leads the reference signal by \( 0.01 \times 2\pi \) rad. At this time one VCO cycle is skipped; the skipping of one VCO cycle delays the divider output by \( 0.01 \times 2\pi \) rad, which is exactly how much the divider output increased in phase.

Although the average output frequency was 1.001MHz in the preceding example, the instantaneous output frequency changes with each reference cycle.
Frequency Synthesizer

because of the increasing phase difference between the divider output and the reference signal. The timing diagram of Figure (17) illustrates this point. If the divider output frequency is slightly faster than the reference frequency, the phase-detector output will consist of pulses of increasing width, and the dc value of these pulses will appear as shown in Figure (18). This voltage will create fluctuations in the output frequency if the frequency is not eliminated before it reaches the VCO.

Figure (17): Timing signals in a PLL frequency synthesizer using fractional division.

Figure (18): Typical waveform of the average value of the phase-detector output of a fractional-N frequency synthesizer.
Frequency Synthesizer

Figure (19) contains a simplified diagram of a fractional-N synthesizer that eliminates the deterministic noise occurring at the phase-detector output by adding a signal equal in magnitude and opposite in sign to the deterministic voltage present at the detector output. The fraction register, adder, and phase register determine how often a pulse is to be removed from the VCO output. The phase register contains the fractional portion of the divisor, and this information is converted to an analog signal in the D/A converter. The analog signal is then used to reduce the phase noise.

![Diagram of a fractional-N frequency synthesizer](image)

Figure (19): A complete fractional-N frequency synthesizer.

One further feature of this analog noise-canceling signal is that it depends on both M and N. If, for example, it is desired to synthesize a frequency of 2.001MHz, then M is again 10, and every 10 reference cycles the output frequency is divided by 201. During each reference cycle the VCO goes through 200.1 cycles. Therefore, the divider output phase relative to the reference
Frequency Synthesizer

frequency advances by $0.001/2\times2\pi$ rad each reference cycle. This phase increment is one-half of that which occurs for an output frequency of 1.001MHz. In general, the amplitude of these steps is inversely proportional to frequency. Therefore, the D/A output amplitude must be adjusted by a programmable gain amplifier, with the gain inversely proportional to N. The analog signal is subtracted from the phase-detector output in order to provide a low-noise VCO control signal.

Direct Digital Frequency Synthesizer

The direct digital frequency synthesizer DDFS is a technology that has been around since the early 1970’s. DDFS is achieved by storing the sine wave values in a lookup table. DDFS consists of a numerically controlled oscillator (NCO), a digital-to-analog converter (DAC or D/A converter), and a low-pass filter. The NCO consists of an adder-register pair (also known as phase accumulator) and a ramp-to-sine wave lookup ROM (a read-only memory). The low-pass filter attenuates the unwanted sampling components and the spurious signals. Figure (20) shows the block diagram of a DDFS.

![Block diagram of direct digital frequency synthesizer (DDFS).](image)

The basic idea is to store $N$ uniformly spaced samples of a sine wave in memory and then to output these samples at a uniform rate to a digital-to-analog converter, where
they are converted to an analog signal. The lowest-output frequency waveform then will contain N distinct points. To generate the lowest frequency, the value 1 is added to the phase accumulator each reference cycle, and the next value from the lookup table is outputted. To output the frequency which is K times as fast as the lowest frequency, the value K is added to the phase accumulator each time and the corresponding value from the lookup table is outputted.

To determine the frequency resolution of a DDFS system, consider a $2^N$-bit phase accumulator and a reference clock $f_{\text{clock}}$. This phase accumulator can address up to $2^N$ different ROM locations (i.e., samples); and if it is incremented by 1 on each clock edge, then all $2^N$ samples are accessed. Assuming that all the samples are unique (usually the case), the frequency resolution is given by:

$$F_{\text{res}} = \frac{f_{\text{clock}}}{2^N}$$

Therefore, the output of the DDFS is related to the phase accumulator input by the following equation:

$$f_{\text{out}} = \frac{K}{2^N} \cdot f_{\text{clock}}$$

Where N is the bit-length of the accumulator and K is the accumulator’s input.

**Example 9:** If a DDFS system uses a 32-bit accumulator and a 10-MHz clock, then find the frequency resolution of this system.

**Solution:**

$$\therefore F_{\text{res}} = \frac{f_{\text{clock}}}{2^N}$$

Then,

$$\therefore F_{\text{res}} = \frac{10 \times 10^6}{2^{32}} = 2.33\text{mHz}$$
Frequency Synthesizer

If P samples are used to represent the waveform at the highest output frequency $f_{\text{max}}$, then $N = \left(\frac{f_{\text{max}}}{f_{\text{min}}}\right)$. P samples are used in the lowest-frequency waveform. The number $N$ is limited by the amount of available memory, and $P$, which must be greater than 2, is determined by the output low-pass filtering requirements. For the period of the highest output frequency,

$$T_{\text{max}} = \frac{1}{f_{\text{max}}} = PT \quad \text{or} \quad f_{\text{max}} = \frac{1}{PT}$$

Where $T$ is the reference clock’s period. Therefore, the highest possible obtainable output frequency is determined by the fastest sampling rate possible.

To complete the DDFS, the memory size and length (number of bits) of each word must be determined. Word length is determined by the system noise requirements. The D/A output samples are those of an exact sinusoid corrupted with deterministic noise due to the truncation caused by the finite length of the digital words. It can be shown that, if an $(n + 1)$-bit word length is used (including 1 bit as the sign bit), the worst-case noise power (relative to the signal) due to the truncation will be approximately:

$$\sigma^2 = (2n)^{-1} \quad \text{or} \quad \sigma^2 = -6n \text{ dB}$$

For each bit added to the word length, the spectral purity improves 6-dB.

For four output samples at the highest frequency, the memory size is determined as:

$$N = 4\frac{f_{\text{max}}}{f_{\text{min}}}$$

Where $N$ is the number of points in the lowest-frequency sinusoid. Clearly $N$ words of memory would be sufficient for storing the data. However, the amount of memory required can usually be marked reduced. First, it is only necessary to store the values for the first quadrant (0 to $90^\circ$) of the sine wave, since the values for the other three
Frequency Synthesizer

Quadrants can be computed directly from these values; so a maximum of \( N/4 = f_{\text{max}}/f_{\text{min}} \) memory points is required. The amount of memory may still be reduced when the spectral purity requirements are not too severe.

Example 10: What word length will be required in a DDFS if the output spectral purity is to be at least 80 dB?

Solution:

Since the noise power is \(-6n\) dB, \( n \) must be at least 14. One additional bit is needed for the sign; therefore, the minimum word length needed is 15 bits for an 80-dB signal-to-noise ratio.

Example 11: design a DDFS to cover the frequency range 0 to 10 KHz with a frequency resolution of at least 0.001Hz. The spectral purity is to be at least 40dB.

Solution:

The use of 8-bit words, including the sign bit, will give a spectral purity of 42dB \((6\times7)\), and this meets the noise specifications. Since

\[
N = 4 \frac{f_{\text{max}}}{f_{\text{min}}} = \frac{4 \times 10^4}{0.001} = 4 \times 10^7 < 2^{26}
\]

It appears at first inspection that a large amount of memory is required. However, only \(2^8 = 256\) different words can be realized using 8-bit words, so 256 memory locations should suffice. The expansion of this apparent contradiction is that although \(4 \times 10^7\) different points are specified, the phase increments are so small that approximately \(2^{26} \div 2^8 = 2^{18}\) increments are needed before a change is registered in the 8-bit word. (A 26-bit word would be required to represent all \(2^{26}\) words).
Frequency Synthesizer

The complete design is illustrated in Figure (21). If four samples are used to represent the maximum frequency of 10 KHz, then a 40 KHz clock is required.

For a resolution of at least 1mHz, a 26-bit phase accumulator is required. However, only 8 bits are needed to address the ROM, and the remainder of the phase accumulator bits is unused.

Figure (21): the direct digital frequency synthesizer discussed in Example 11.

Problems:

1. Design a direct frequency synthesizer to generate $15.8 \times 10^6$Hz from a $1 \times 10^6$Hz reference oscillator.

2. Design a direct frequency synthesizer, using double-mix-and-divide modulus, to cover the frequency spectrum of 25 to 29.999MHz in 1-KHz increments. Specify all frequencies used and the maximum frequency mixing ratio of all mixers.

3. Design a phase-locked loop synthesizer to meet the specifications of problem 2.

   A. What is the reference frequency?
   
   B. What is the range of the divide ratio N?
Frequency Synthesizer

3. The 25-MHz output frequency is too high for a programmable divider, and a variable-modulus divider should be used. Use a 10/11 divider, and determine the initial values of the counters required to synthesize 26.111MHz.

4. Could a 100/101 variable-modulus divider be used in problem 3? Explain your answer.

5. Design a multiple-PLL synthesizer, to cover the frequency range of 35.4 to 40.0MHz to 10-Hz increments. The reference frequency is to be 100KHz. No loop should operate with a reference frequency below 100KHz.

6. Use a fractional-N frequency synthesizer to synthesize a frequency of 2.33KHz using a 1-KHz reference frequency.
   A. What must be the size of the phase accumulator?
   B. Sketch the output of the phase detector (N = 2, M = 3).
   C. Sketch the phase-detector output if the output frequency is to be 4.33KHz (N = 4, M = 3).
   D. What must the phase accumulator size be to realize a frequency resolution of 10Hz?

7. It is desired to design a direct digital frequency synthesizer with a maximum output frequency of 10KHz and a step size of 10Hz.
   A. What must the clock frequency be if four samples are to be used in the highest-frequency waveform?
   B. How many bits must the accumulator contain?
   C. What number must be added to the accumulator to each cycle to generate an output frequency of 100KHz?
   D. What must the accumulator size be if the minimum step size is to be reduced to 1Hz?
   E. What word length is required for a 5-dB signal-to-noise ratio?
8. Design a DDFS to cover the frequency range of 0 to 5KHz in 0.01-Hz increments. The spectral purity is to be at least 50-dB. Specify the accumulator size, memory requirements, sampling rates, and characteristics of the output low-pass filter?

9. Design a frequency synthesizer, to cover the frequency range of 0 to 100.999MHz in 10-Hz increments. The frequency switching time is to be less than 100µs. Discuss different configurations which can be used to meet the specifications.

10. Design a three-loop synthesizer, to cover the frequency range of 198 to 200MHz with a frequency resolution of 10Hz. The loop frequency switching time should be as short as possible.

11. Discuss various method of realizing a frequency synthesizer, to cover the frequency spectrum of 10Hz to 1KHz in 10-Hz increments. The discussion should include the reference frequency used in each case, the synthesizer switching speed, and the advantages and disadvantages of each method.

12. Figure below illustrates another method for covering the 198- to 200-MHz frequency range with a frequency resolution of 10Hz. Select frequency ranges for two direct digital frequency synthesizers, and specify the frequencies of the four oscillators used to mix the frequency up to the specified operating frequency.

13. Design a DDFS to cover the frequency range of 0 to 1MHz with a frequency resolution of at least 0.001Hz. The spectral purity is to be at least 55-dB.

14. Design a three-loop frequency synthesizer, to cover the frequency range from 35.40 to 40.00MHz in 1-KHz increments. The reference frequency is to be 100KHz. Determine $N_A$ and $N_B$ required to obtain an output frequency of 38.912MHz.
Figure (Problem 12): A hybrid direct/direct digital frequency synthesizer.