

## Summary

### Harmonic Frequency Synthesizer

Uses an accumulator technique to generate frequencies that are evenly spaced harmonics of some minimum frequency. Extensive pipelining is employed to permit high clock rates.

### FSK Modulator

A modification of the Harmonic Frequency Synthesizer that automatically switches between two frequencies in accordance with an NRZ input.

## Specifications

### Harmonic Frequency Synthesizer

Maximum Output Frequency	67 MHz
Minimum Output Frequency	1 Hz
Frequency Spacing	1 Hz
Clock Frequency	67 MHz
Number of Bits	26
Number of CLBs	52

### FSK Modulator

Operating Frequencies	10/11 MHz
Jitter	±8 ns
Clock Frequency	64 MHz
Number of CLBs	10

## Xilinx Family

XC3000A/XC3100A  
XC4000/A/D/H

## Demonstrates

Pipelining

## Introduction

Most frequency synthesizers derive their output by using programmable counters to divide the clock frequency. This results in a set of attainable output frequencies that are sub-harmonics of the clock, and are defined by the following equation.

$$f_{OUT} = f_{CLK} / N$$

These frequencies are unevenly spaced, and the spacing becomes especially coarse as the required frequency approaches the clock frequency, where only one half, one third, etc. are available. If more than one exact frequency is required, the clock must be a common multiple of these frequencies.

A better approach is to use an accumulator to generate the frequencies, as shown in Figure 1. This results in a set of harmonic frequencies, defined by the equation:

$$f_{OUT} = N \times f_{CLK} / 2^n$$

Here the attainable frequencies are evenly spaced. If multiple frequencies are required, the clock need only be a binary multiple of a common factor of the frequencies. This requirement is often easier to satisfy. In particular, if

the clock rate is a power of two, all integer frequencies up to the clock rate can be generated.

It must be recognized, however, that these frequencies describe the average rate at which output pulses are generated. Output transitions can only be generated an

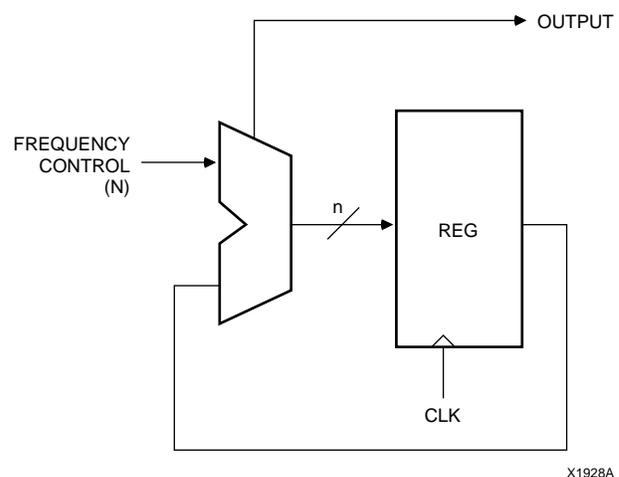


Figure 1. Accumulator-based Frequency Synthesizer

integer number of clock periods apart, and this leads to jitter. As the output frequency approaches the clock rate, this jitter becomes severe.

A potential disadvantage of this scheme is the complexity of the adder and its effect on speed, when compared to the counter approach. However, this can be overcome through the use of pipelining.

**Operating Description**

Each Xilinx XC3000-series and XC4000-series CLB contains two flip-flops. One of these can be used to form the accumulator register, leaving the other to pipeline the carry path. A pipeline flip-flop is inserted between all the bits of the adder. The output skew this creates is not a problem as only the carry-out is of interest.

Matching the pipeline delay at the input is also not an issue if only one frequency is required, as the input never changes. If multiple frequencies are required, the input might simply be changed, but this would cause a phase discontinuity. Where this is unacceptable, a delay equalizer must be added, such that each addition into the accumulator is completed with the same input.

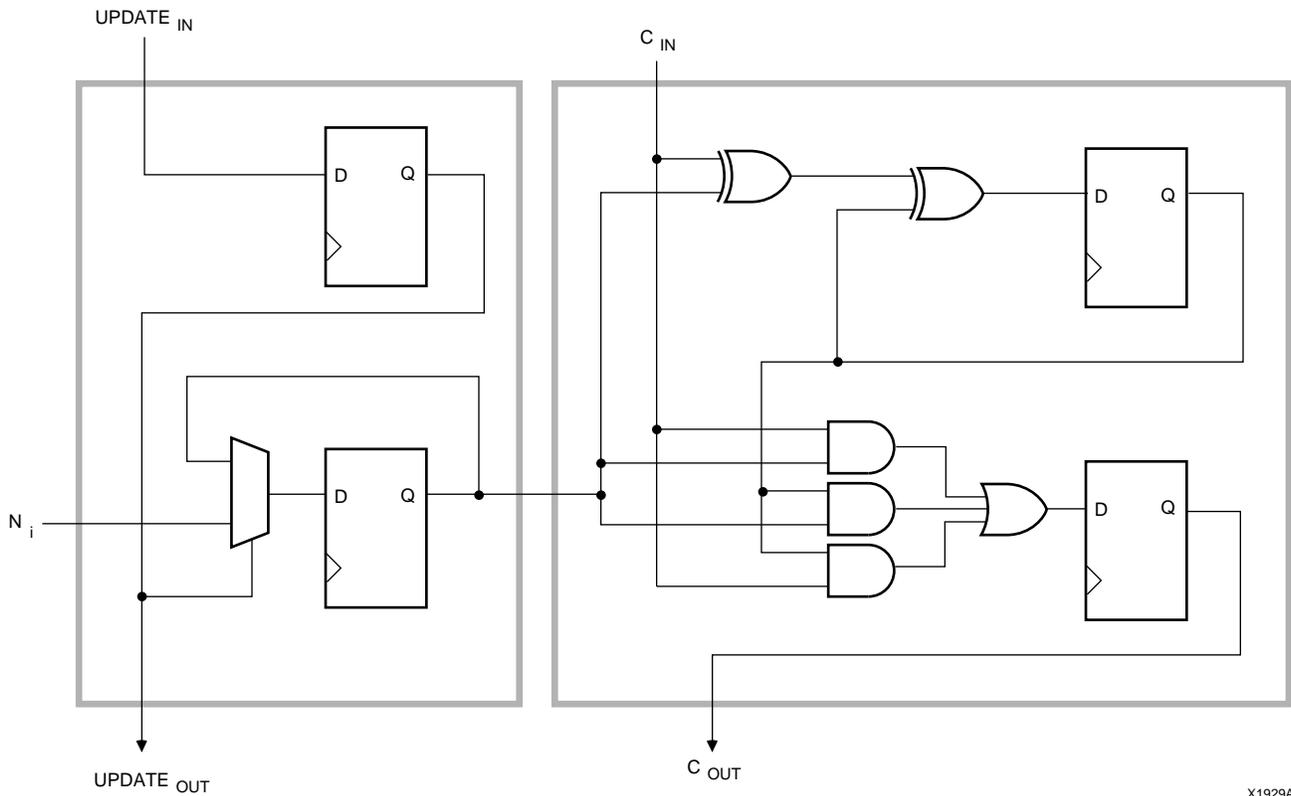
Conceptually, this requires a triangular array of registers, generating a 1-clock delay into the input of the second bit,

a 2-clock delay into the third bit, and so on. However, this can be greatly simplified if the input only changes occasionally.

Figure 2 shows the accumulator cell with its delay equalizer. The accumulator cell is a simple full adder with its output registered and fed back to one of its inputs. A pipeline flip-flop is introduced into the carry path.

The accumulator input that controls the frequency is stored in a register. Individual bits of this holding register are enabled from flip-flops that are connected as a shift register. When the frequency is to be changed, the appropriate number is input to a holding register, and a single one introduced into the shift register. As this one propagates through the shift register, individual bits of the holding register are successively updated. This update occurs in synchronism with an addition propagating through the pipelined adder.

For an n-bit accumulator, the data must be held at the input to the holding register for n clocks after the update pulse. This is the only restriction on how fast the frequency can be changed. Also, it takes n clocks from the update pulse before the frequency change is reflected at the output. At this time, however, the change is instantaneous and phase continuity is maintained.



**Figure 2. Bit-slice of the Frequency Synthesizer**

This synthesizer design uses two CLBs per bit. Exploiting the direct interconnect between CLBs in the XC3000-series devices, either version can be operated at clock frequencies in excess of 90 MHz in a -125 part. If placement and routing do not provide for direct interconnect, the maximum speed is reduced. This is always true when the accumulator is longer than one column of CLBs in the target LCA device.

As an example, a 26-bit frequency synthesizer, clocked at 67.108864 MHz ( $2^{26}$  Hz), generates every integer-valued frequency up to this clock rate. Fifty-two CLBs are required, and the synthesizer fits into any XC3000-series LCA device.

A harmonic frequency synthesizer is useful as an FSK modulator. For FSK modulation, the synthesizer must alternate between two frequencies. This can easily be accommodated by modifying the delay equalizer, as shown in Figure 3.

Two numbers, appropriate to the two frequencies, are applied to the delay equalizer. If the frequencies must be programmable, these numbers can come from inputs or registers. Otherwise, they can be hard-wired at the inputs of the CLBs, or individual function generators can be modified to incorporate them.

NRZ data is applied to the shift register. As this propagates through the shift register, multiplexers at the input to

each bit of the holding register detect changes in the data. When a change is detected, the bit is reloaded from the appropriate number. Again, changes ripple through the holding register in synchronism with the additions. The NRZ data may change every clock, if required.

A typical FSK modulator, as shown in Figure 4, might be required to switch between 10 and 11 MHz. To give a square output, a flip-flop is used to divide the synthesizer output by two. This flip-flop may be the carry pipeline of the final adder, modified to toggle with the carry rather than storing it.

The toggle flip-flop must be enabled at frequencies that are twice the output frequencies. The largest common factor of 20 and 22 MHz is 2 MHz, and the clock frequency must be a binary multiple of this. Higher binary multipliers will result in lower jitter. In this case, 64 MHz is chosen. This is  $2^5$  times 2 MHz, and a 5-bit accumulator must be used. Twenty and 22 MHz are 10 and 11 times 2 MHz, respectively, and these are the numbers that must be accumulated to generate the frequencies (0A Hex and 0B Hex). This FSK modulator may be implemented in only 10 CLBs.

If an analog output is required, either version of the synthesizer may be used to control a counter, as shown in Figure 5. The output of this counter is used to access a look-up table, which provides data to a DAC.

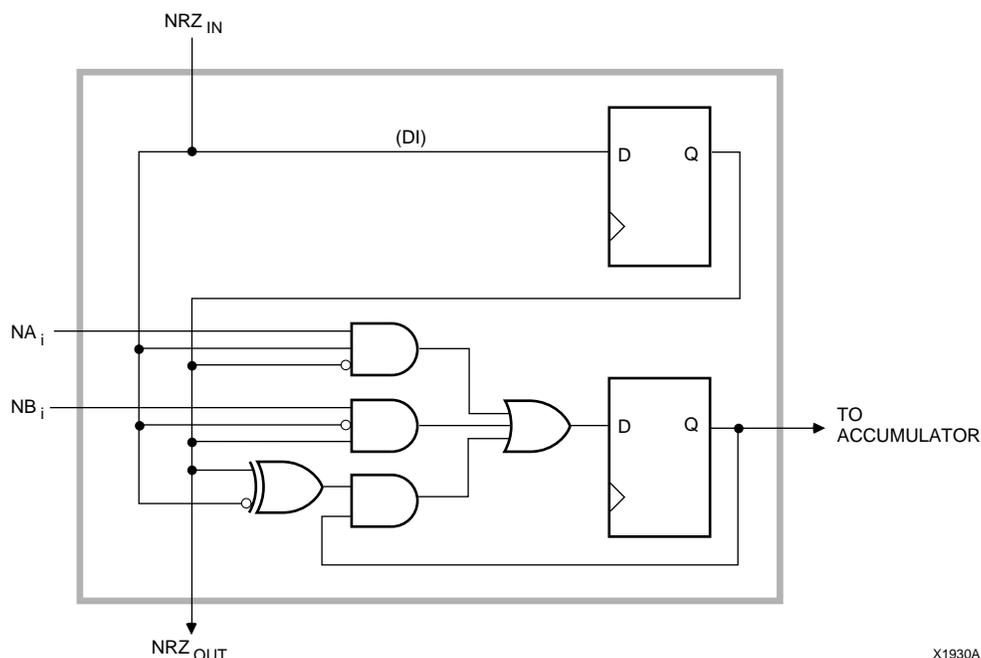


Figure 3. Delay Equalizer for an FSK Modulator

In the XC3000-series, the CLB function generators may be used as ROMs to implement the look-up table internally. The CLBs actually contain RAMs that are written during configuration. As a result, multiple wave-shapes can be supported by re-configuring the LCA device. The

XC4000-series provides user-accessible RAM in the CLB. Wave-shapes, therefore, can be changed on the fly.

An external look-up table may also be used. In particular, a video RAMDAC can be loaded with the wave-shape. This is sequentially addressed at appropriate intervals to generate the waveform with the desired frequency.

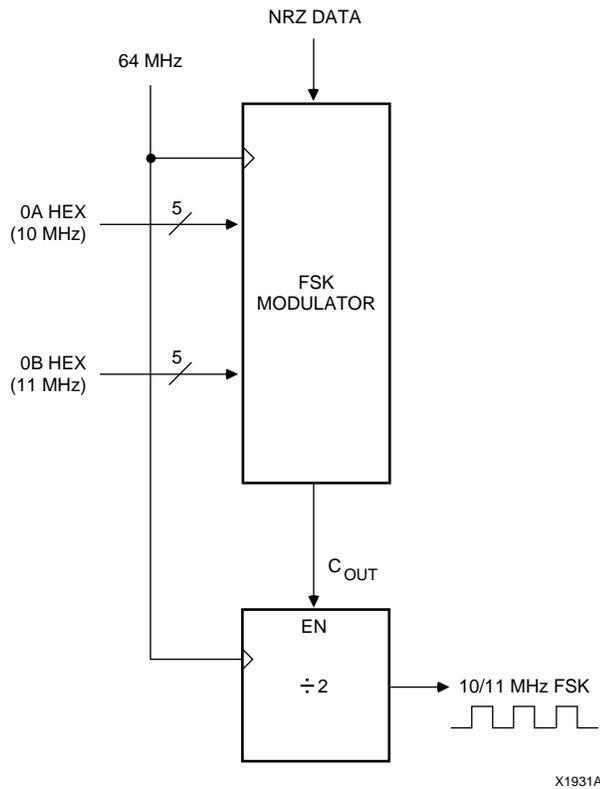


Figure 4. 10/11-MHz FSK Modulator

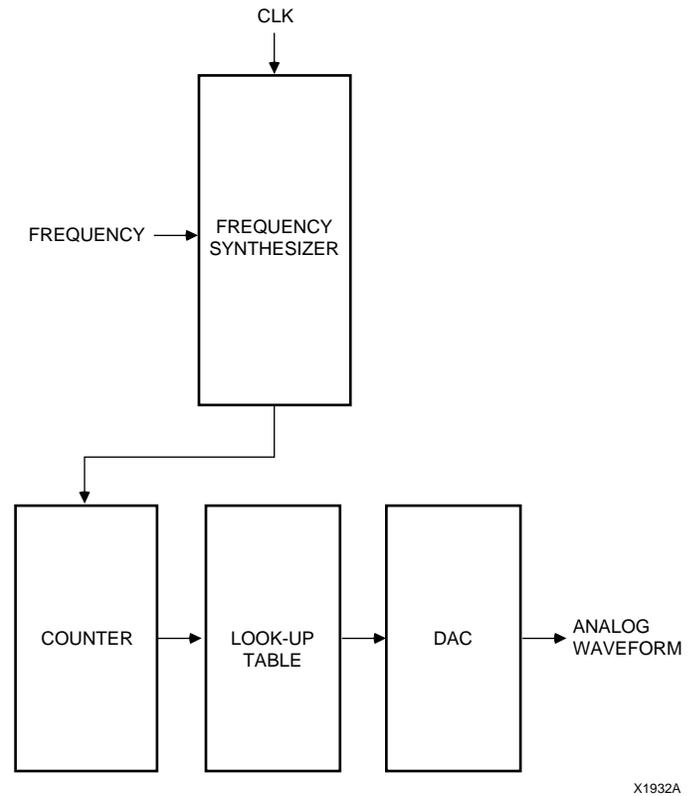


Figure 5. Analog Waveform Generator