**FSK: Implementation Details**

**FSK : Theory**

Consider angle modulated wave

Acos(wc.t+φ(t)) where A,wc are constants and φ(t) is varying.

Phase= wc.t+φ(t)

Instantaneous frequency(wi):wc+ φ’(t)

In FM, φ’(t) is proportional to message signal m(t)

fi= wc + pm(t)

For FSK, m(t) can be 0 or 1

For 0 wi = wc = 2π(fc)

For 1 fi=wc + p=2π(fc + p/2π), where fc is 1MHz

These are the output frequencies Fo0 and Fo1.

For testing of the FSK Modulation, we will do the demodulation of the modulated signal and reconstruct original input digital data bits.

Specifications

1. input digital data rate=100kbps
2. Base Frequency=1MHz
3. Separation between space and marked frequencies=150KHz

We have to transmit Logic 0 as 1MHz and Logic 1 as 1.15MHz

Implementation on FPGA

DDS (Direct Digital synthesizer) IP

This IP consists of phase accumulator and sin and cos lookup table.

INPUT:

1) Clock

2) Phase increment corresponding to output signal to be generated.

OUTPUT:

Sine or cosine values of desired signal

Important Formula derivations:

Phase increment:

Phase increment is some value proportional to desired output frequency.

v0= phase increment corresponding to 0

v1 = phase increment corresponding to 1

v0 = k.(fo0)

v1 =k.(fo1)

fo0= frequency corresponding to signal 0

fo1= frequency corresponding to signal 1

k= proportionality constant which is property of DDS

k:

DDS accumulates phase increment and makes 1 wave till the accumulator fills up to maximum

Let w = Width of accumulator

fi= input clock frequency to DDS i.e., the rate at which phase increment is accumulated

Let n0 = number of clock cycles of DDS clock required to form 1 wave corresponding to frequency of 0 input or no number of samples for 1 wave of fi frequency

no/fi = 1/fo0--------------------------------- equation 1

fo0 =output frequency corresponding to “0”

When phase accumulator reaches its maximum, 1 wave is formed so,

n0.V0 <= 2w -1; ---------------------------equation

Replacing n0 in equation 2 by equation 1 we get,

V0 <= fo0. (2w-1)/fi;

NOTE: Value of V0 will be the greatest integer smaller than the above value. V0 can’t be a floating point number. For example, if V0 <= 3.45, 3 will be used.

Similarly,

V1 = fo1. (2w-1)/fi ;

V1 = phase increment corresponding to “1”

fo1 = DDS output frequency for “1

V0=k.(fo0) ;

Number of Waves in Output Signal :

This is great for troubleshooting.

Data rate = d kbps

fo = output frequency { fo0 for ‘0’, fo1 for ‘1’}

td = time for a data = 1000/d us

nw =number of waves in time td

{nw0 for ‘0’ , nw1 for ‘1’}

nw.(1/fo) =1000/d ----- fo in MHz

That is,

nw0.(1/fo0) =1000/d; {for signal “0”}

nw1.(1/fo1) =1000/d; {for signal “1”}

Summarizing formulas and calculations

For our parameters:

V0 <= fo0 . (2w-1)/fi ;

V1 = fo1 . (2w-1)/fi;

nw0.(1/fo0) =1000/d; {for signal “0”}

nw1.(1/fo1) =1000/d; {for signal “1”}

fo0 = 1 MHz

fo1 = 1.15 MHz

w=40;

d = 100; {data rate 100 kbps}

fi = 100 MHz ;

It is chosen to be 100MHz so that for each wave of 1MHz we have 100 samples given by

no/fi = 1/fo0; as explained earlier.

V0 = 10995116277;

V1 = 12644383720;

nw0=10;

nw1= 11.5;

no0 = 100;

no1 =86;

td = 10 us; {time for 1 data}

Summary:

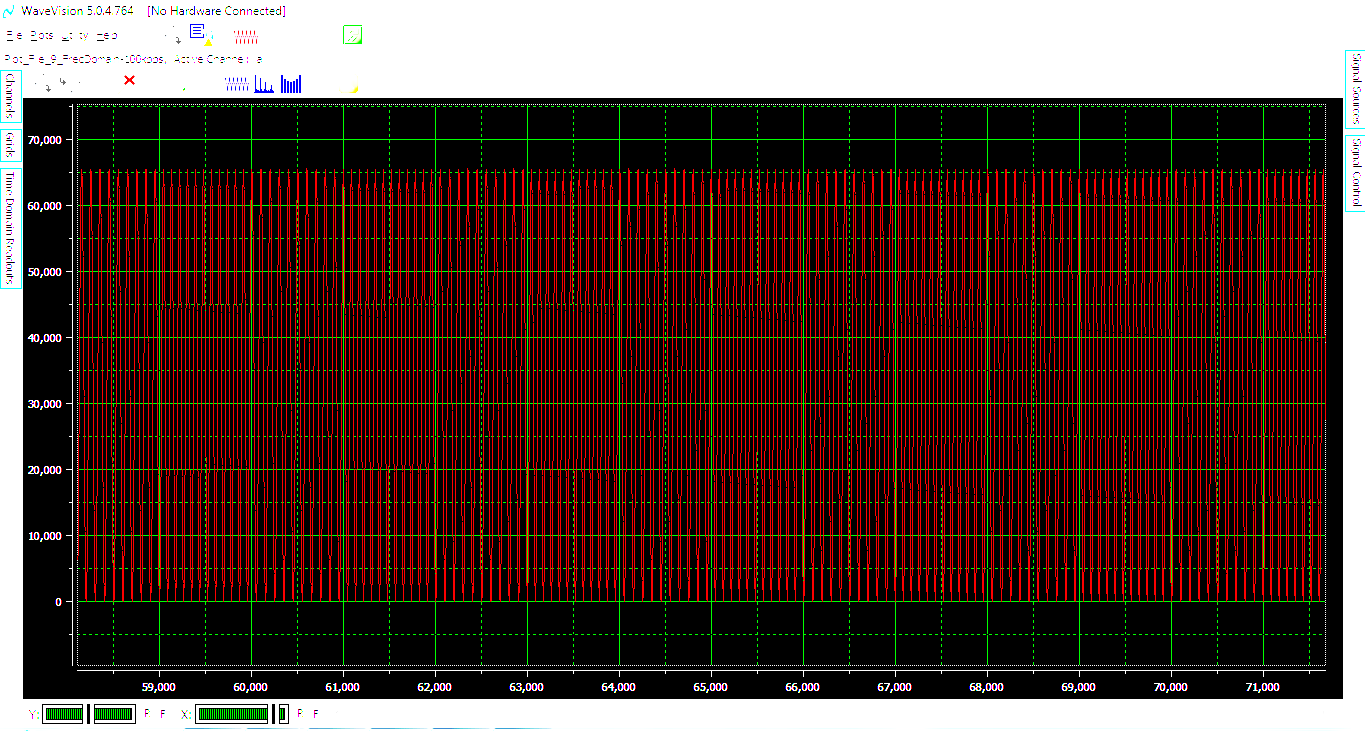
Outputting 100 samples per wave or accumulating V\_0 100 times a wave of frequency 1MHz is generated and 10 such waves are generated for every “0” input in 10 us.

Similarly, outputting 86 samples per wave or accumulating V\_1 86 times a wave of frequency 1.150MHz is generated and 11.5 such waves are generated for every “1” input in 10 us.

1. Phase increment for bits 0 and 1 are respectively 10995116277 and 12644383720.
2. A 2X1 MUX is used to choose between phase increments with the incoming data as the control signal.
3. This phase increment is passed to the DDS at 100MHz corresponding to the incoming value (0 or 1).
4. For one wave to be formed we require phase increment of one bit to be read 100 times.
5. We are using 100MHz because incoming data rate is 100kbps. So time for one bit is 10µs. For 1 wave we need to read incoming data 100 times i.e, every 10/100µs. So frequency is 1/(10/100 µs)=10MHz
6. For obtaining 10 waves corresponding to each data bit, our frequency has to be 100MHz.
7. We implemented the design on Kintex 7 FPGA and exported the ILA data to a Wave plotter.

**RESULTS**

The resultant waveform had sinusoids of frequencies 1.15 MHz and 1 MHz respectively while passing 1 and 0 alternatively. The resulting square wave had a frequency of 50 KHz



The following is the frequency spectrum of the waveform when we passed 0. The peak as seen is 1MHz.



A peak of 1.15 MHz was observed when we passed 1.

