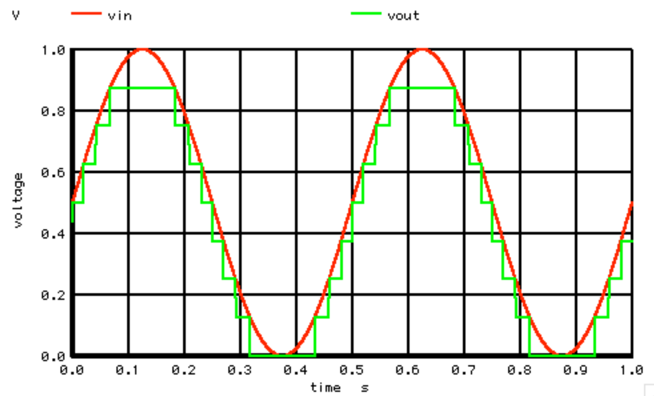
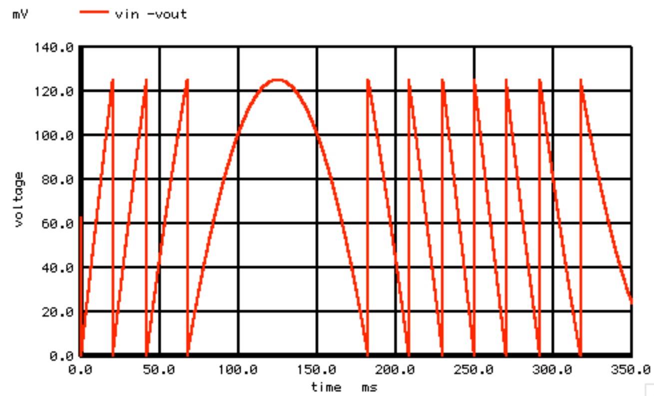
=======Asynchronous_Analog_to_Digital_Conversion========

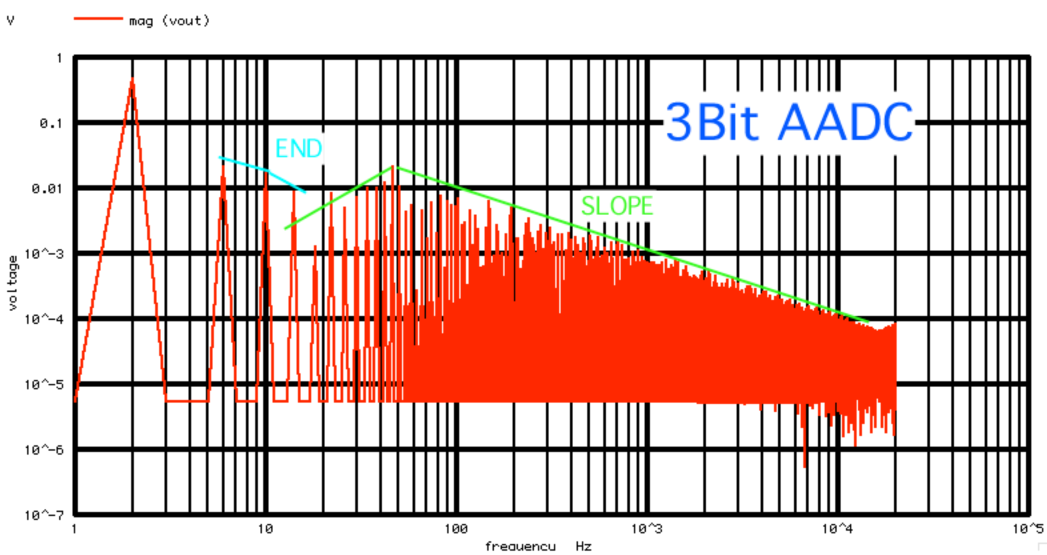




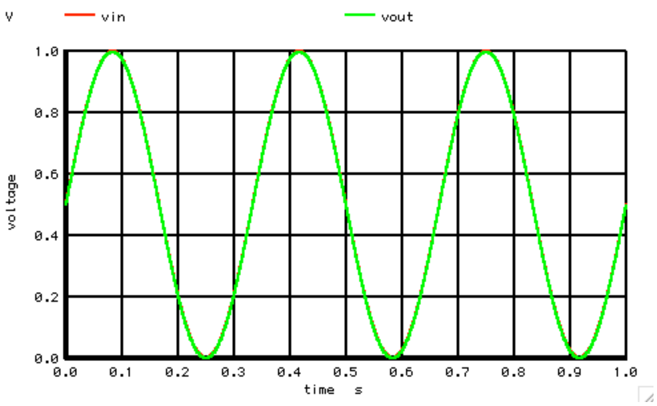
Think of a voltage comparator which has multiple bits of output.
Such a comparator will try and track an moving input signal to an LSB resolution.
It only takes the following six lines of spice code
to create an ideal 3Bit ADC followed by a DAC.

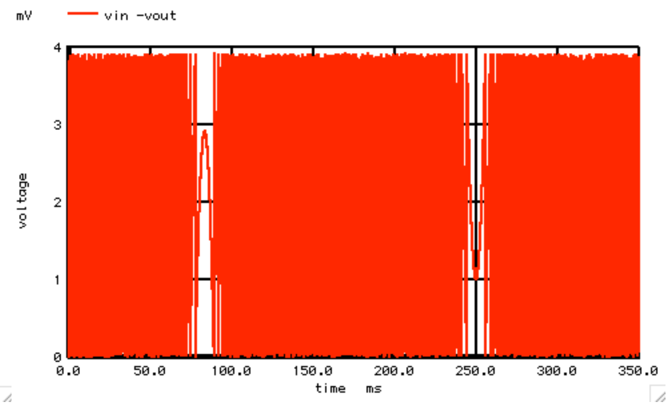
```
BTH0      D0      0       V =     u( V(VIN)  -1/2)
BOFF0     VIN     OFF0    V =     V(D0)/2
BTH1      D1      0       V =     u( V(OFF0) -1/4)
BOFF1     OFF0    OFF1    V =     V(D1)/4
BTH2      D2      0       V =     u( V(OFF1) -1/8)
BDAC      VOUT    0       V =     V(D0)/2+V(D1)/4+V(D2)/8
```

But the data is being captured in a way that is much different than a normal.
A Normal ADC captures the input when it is within a LSB voltage range at a precise time.
This type of sampling detects the input crossing a precise voltage at a precise time.
In other words, it samples only when something happens.
It can be called a Asynchronous Analog to Digital Convertor or AADC for short.

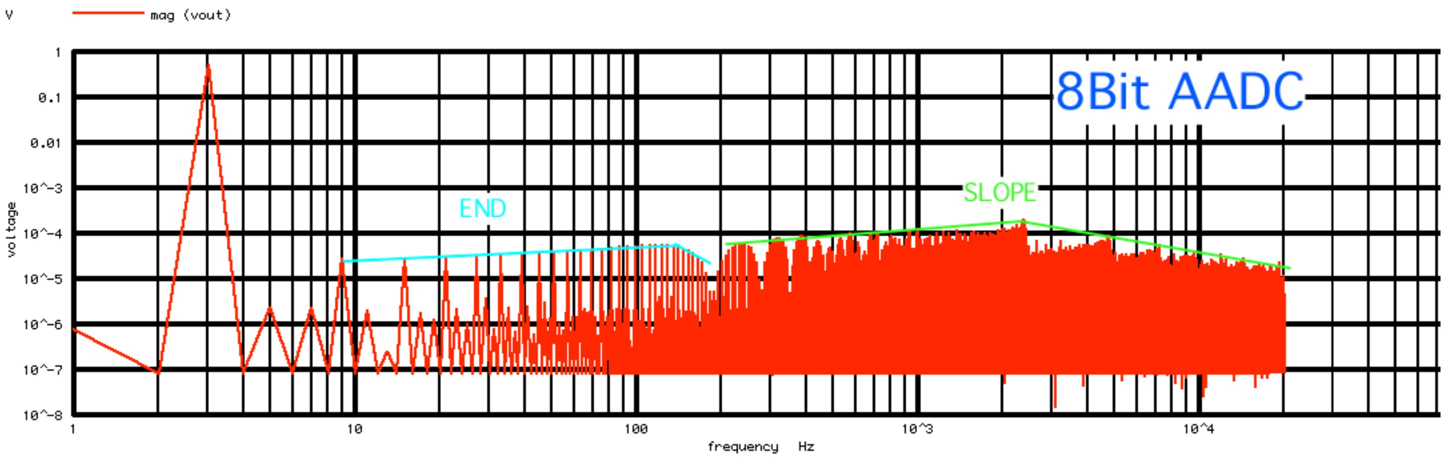


The spectrum of a 3Bit AADC is much different.
A normal ADC has a noise floor defined by the LSB magnitude.
And it displays images of the input signal mirrored around the sample frequency.
Since the AADC has no sample rate, there are no aliasing images.
The spectrum is actually two different types of distortion for the sine-wave example.
There is triangle wave distortion and endpoint distortion.





Adding more stages can simulate a 8Bit ADDC.



At 8Bits, two different types of distortion have undergone some shifts in frequency and magnitude. But still, the LSB resolution of the AADC is not producing a noise floor. It is producing a distortion floor level instead.

========Spice_Code=================================

```
Simple_Asynchronous_ADC_FFT
*
*       VIN
*
*      ____________________   __________   __________   __________   __________   __________   __________
*     _|_      |  TH0      _|_  | TH1      _|_  | TH0      _|_  | TH1      _|_  | TH0      _|_  | TH1      _|_  | TH0      _|_        TH1
*    /   \     | |\      _/   \ | |\     _/   \ | |\     _/   \ | |\     _/   \ | |\     _/   \ | |\     _/   \ | |\     _/   \       |\
*   //\   \    | | \    | \___/ | | \   | \___/ | | \   | \___/ | | \   | \___/ | | \   | \___/ | | \   | \___/ | | \   | \___/       | \
*   \   \//    |_|  >_|   |_____|_|  >_|   |_____|_|  >_|   |_____|_|  >_|   |_____|_|  >_|   |_____|_|  >_|   |_____|_|  >_|   |__________|  >_
*    \___/       | /  |  OFF0      | /  |  OFF1      | /  |  OFF2      | /  |  OFF3      | /  |  OFF4      | /  |  OFF5      | /  |  OFF6      | /  |
*     _|_        |/   |            |/   |            |/   |            |/   |            |/   |            |/   |            |/   |            |/   |
*     ///            _|_              _|_              _|_              _|_              _|_              _|_              _|_              _|_
*                   |D0 |            |D1 |            |D2 |            |D3 |            |D4 |            |D5 |            |D6 |            |D7 |
*                   |___|            |___|            |___|            |___|            |___|            |___|            |___|            |___|
*
*
*         ________
*        __|      \
*        __|       \
*        __|  DAC   \__
*        __|        /
*        __|       /
*        __|______/
*
*=========Create_Signal=================
*V_SIN#    NODE_P NODE_N DC     VALUE  SIN(   V_DC   AC_MAG FREQ   DELAY  FDamp)
VIN        VIN    0      DC     0      SIN(   .5      .499    3              )

BTH0       D0     0      V =    u( V(VIN)  -1/2)
BOFF0      VIN    OFF0   V =    V(D0)/2
BTH1       D1     0      V =    u( V(OFF0) -1/4)
BOFF1      OFF0   OFF1   V =    V(D1)/4
BTH2       D2     0      V =    u( V(OFF1) -1/8)
BOFF2      OFF1   OFF2   V =    V(D2)/8
BTH3       D3     0      V =    u( V(OFF2) -1/16)
BOFF3      OFF2   OFF3   V =    V(D3)/16
BTH4       D4     0      V =    u( V(OFF3) -1/32)
BOFF4      OFF3   OFF4   V =    V(D4)/32
BTH5       D5     0      V =    u( V(OFF4) -1/64)
BOFF5      OFF4   OFF5   V =    V(D5)/64
BTH6       D6     0      V =    u( V(OFF5) -1/128)
BOFF6      OFF5   OFF6   V =    V(D6)/128
BTH7       D7     0      V =    u( V(OFF6) -1/256)
```

```
BOFF7     OFF6   OFF7   V =    V(D7)/254
BTH8      D8     0      V =    u( V(OFF6) -1/512)
BOFF8     OFF7   OFF8   V =    V(D8)/245
BDAC      VOUT   0      V =    V(D0)/2+V(D1)/4+V(D2)/8+V(D3)/16 +V(D4)/32 +V(D5)/64 +V(D6)/128 +V(D7)/256

.control
*TRAN     TSTEP  TSTOP  TSTART TMAX   ?UIC?
tran      .0125m    1      0      .0125m
set       pensize = 2
plot      vin  vout

plot      vin -vout xlimit 0 .35

echo             "===================FFT_and_Plot====================================="
linearize
let              FFT_BandWidth_Hz =  20k
let              FFT_resolution_Hz = 1
echo             "FFT_BandWidth_Hz=  $&FFT_BandWidth_Hz"
echo             "FFT_resolution_Hz= $&FFT_resolution_Hz"
set              specwindow =         "rectangular"
spec             $&FFT_resolution_Hz  $&FFT_BandWidth_Hz   $&FFT_resolution_Hz     v(vout)
*let              freqL = length(frequency)
*let              expect = (1/16)/frequency
plot             mag (vout)    loglog
*plot            mag (vout)  ylog xlimit 400 600
echo             "===================Done============================================="

.endc

.end
```

8.18.11_2.01PM

dsauersanjose@aol.com
Don Sauer