# SAMPLE RATE CONVERTER FOR DIGITAL MODULATOR

Zhuan Ye (Motorola Labs, 1301 E. Algonquin Rd., Schaumburg, IL 60196, USA
Zhuan.ye@motorola.com );
John Grosspietsch (Motorola Labs, Schaumburg, IL 60196, USA.
John.grosspietsch@motorola.com )

## ABSTRACT

A software defined radio (SDR) terminal needs to support several different air interface standards. These standards often define their own unique symbol or chip frequencies. Therefore a sample rater converter (SRC) is an essential part in such SDR terminal. This paper presents a novel architecture for sample rate converter and its implementation techniques for modulator or transmitter applications.

## 1. INTRODUCTION

Today's wireless communication market presents many different standards. This situation has drawn interest to software defined radio (SDR), which can offer seamless, any time and anywhere communication services [1]. An SDR advocates that physical layer signal processing tasks to be implemented using software as much as possible. With the advancement of integration circuit (IC) technology, the future SDR platform may consist of a digital signal processor (DSP) with several accelerators for computing intensive tasks, and the necessary analog to digital (ADC) and digital to analog (DAC) interfaces [1].

The abovementioned ADCs and DACs require high quality, low jitter clock sources to guarantee performances. A traditional approach is to provide a clock with a frequency that is integer multiples of the desired symbol or chip frequency. Different air interfaces define their own unique symbol or chip frequencies. When multiple air interfaces need to be supported, it is much more cost effective to use only one common clock source instead. Therefore it is essential to include sample rate converter (SRC) in the SDR platform. An SRC translates a digital signal sampled at one rate to another sample rate, while the essential information of the original signal should be preserved.

SRCs can take place in both the receiver (ADC) side and transmitter (DAC) side. This paper focuses on the transmitter side, or often referred to as a digital modulator. The SRCs used in digital modulators have been presented in GSM/EDGE/WCDMA modulator [2] and single chip QAM modulator [3]. The SRC presented in this paper allows most of the modulator's interpolation filtering to be implemented using DSP software, which can be decoupled from the final DAC clock. This feature offers more suitability from SDR's point of view.

This paper will be organized as follows. Section 2 provides a brief introduction to SRC. The emphasis of this paper, polynomial based SRC is described in Section 3. In section 4, the architecture and the proposed implementation techniques are described in detail. Finally, a design example of a digital QAM modulator and its measured results are presented in Section 5; these are followed by a few concluding comments.

## 2. SAMPLE RATE CONVERTER

The underlying theory of sample rate converter is the well-studied interpolation filter. Interpolation, as its name suggests, is used to calculate new samples at arbitrary time instants in between existing discrete-time samples [4]. As shown in Figure 1, the original series of samples are denoted as $x[nT_s]$; while the interpolation results are denoted as $x[(n+\mu)T_s]$. The $\mu$ values are often referred to as fractional intervals, representing the sampling time offsets between the new sampling instant and the original sampling instant defined by

$$\mu_k = \frac{(kT_{out} - n_k T_s)}{T_s}, \tag{1}$$

where $T_s$ and $T_{out}$ are the old and new sampling intervals respectively.

The interpolation is essentially a reconstruction problem that can be analyzed using the hybrid analog/digital conceptual model shown in Figure 2. In this model the interpolation outputs are obtained by sampling the analog signal constructed from original discrete-time signal by digital to analog conversion (DAC). Therefore the output of the interpolation is

$$y(kT_{out}) = \sum_{i=-\infty}^{+\infty} x[(n_k - i)T_s] h_a[(i+\mu_k)T_s], \tag{2}$$

where $h_a(t)$ is the impulse response of the analog reconstruction filter. The parameter $n_k$ denotes the basepoint index that identifies the set of original samples (basepoints) to be used for the k$^{th}$ interpolation. This model is only used for analysis purpose since almost all sample rate converters (SRC) are implemented via an all-digital fashion.

One possible implementation for SRC is to use traditional FIR filter to calculate the interpolation filter output x[(n+μ)T$_s$]. As shown in Equation (2), the coefficients for this FIR filter vary with the fractional interval $\mu_k$. Therefore the filter coefficients need to be pre-computed and stored in a memory for each possible $\mu_k$. This technique may impose large memory requirement when the number of $\mu_k$ becomes very large to meet fine timing resolution requirement of the interpolation.

An alternative method, known as the polynomial based interpolation, will be described in the next section. This method doesn't require any memory storage for the filter coefficients, since the coefficients can be computed based on the fractional interval value $\mu_k$.

## 3. POLYNOMIAL BASED SRC

Polynomial based SRC is derived from the classic interpolation problem in numerical analysis. The interpolation filter used for the polynomial based SRC, has an impulse response that can be represented using a polynomial, or piecewise polynomial of $\mu_k$. Therefore, the underlying low pass filter h$_a$(t) in Equation (2) can be expressed in each interval T$_s$ by means of polynomial [5]:

$$h_a(t) = h_a[(n+\mu_k)T_s] = \sum_{l=0}^{N-1} b_l(n)(\mu_k)^l , \qquad (3)$$

where $b_l(n)$ denotes the polynomial coefficients in the n$^{th}$ interval.

A special class of such polynomial is Lagrange interpolation polynomial. Lagrange polynomial interpolation is a classical numerical technique to fit original N samples via a polynomial of degree N-1. Reference [4] described in detail how to construct linear phase interpolation filter by using Lagrange polynomial for a set of basepoints consisting of either even number or odd number of original samples. It is essential to ensure the linear phase property of the reconstruction filter to avoid delay distortion, especially when digital modulator is the target application. The impulse responses of the underlying analog reconstruction filter constructed using Lagrange polynomial using odd number of basepoints (N=3) and even number of basepoints (N=4) are shown in Figure 3. Although for the example consisting of odd number (N=3) of basepoints, the impulse response is not time-continuous; the interpolation output is time-continuous if the interpolation approximation error is neglected [4]. The significance of extending such interpolation to be feasible by using odd number of basepoints is that it allows the flexibility to choose the lowest degree polynomial that can meet the system requirements.

The polynomial based interpolation has some advantages in handling the filter coefficients varying with the fractional interval $\mu_k$. Substitute (3) into (2) and assume that the h$_a$(t) only have non-zero response in the duration of [-M$_1$T$_s$, +M$_2$T$_s$], the interpolation results can be computed from [5]

$$\begin{aligned} y(k) &= \sum_{i=-M_1}^{M_2} x(n-i) \sum_{l=0}^{N-1} b_l(i)(\mu_k)^l \\ &= \sum_{l=0}^{N-1} (\mu_k)^l \sum_{i=-M_1}^{M_2} b_l(i)x(n-i) \qquad , \qquad (4) \\ &= \sum_{l=0}^{N-1} (\mu_k)^l v(l), v(l) = \sum_{i=-M_1}^{M_2} b_l(i)x(n-i) \end{aligned}$$

The polynomial coefficients are determined by the order and type of the polynomial selected for the interpolation, independent of $\mu_k$. Equation (4) also describes a method to compute the interpolation result without explicitly calculating the filter coefficients. Equation (4) can be efficiently evaluated using the nested approach. For example, when N = 3:

$$y(k) = [v(2)\mu_k + v(1)]\mu_k + v(0) \qquad (5)$$

A hardware block diagram implementing Equation (5) is shown in Figure 4. In each column, the value of *v(l)* is evaluated. The multiplications and additions at the bottom of the diagram perform the polynomial calculation using the nested approach. This structure is often referred to as Farrow structure [5]. Farrow structure is only applicable to polynomial based SRC. The following sections will focus on the implementations of polynomial based SRC using Farrow structure.

## 4. IMPLEMENTATIONS OF POLYNOMIAL BASED SRC

In this section, the implementation techniques of polynomial based SRC will be discussed. The SRC mentioned in this section will refer to polynomial based SRC only for simplification purpose. The intent of this paper is to describe the SRC for a digital modulator; therefore the common architecture of a digital modulator is illustrated in Figure 5. A digital modulator usually consists of a square root raise cosine (SRRC) filter for pulse shaping purpose; several stages of FIR interpolation filters to increase the sample rate closer to the final output rate, or the rate the DAC is running at; and an optional sample rate converter that converts sample rate at the output from multistage interpolation filter, which is usually an integer multiple of the input symbol rate ($M \times F_{sym}$), to the arbitrary output sample rate ($F_{DAC}$).

This section is further divided into the following subsections: first the overall architecture of the polynomial based SRC is discussed; in the next two sections, the difference between SRCs using even number of basepoints and odd number of basepoints will be explained in detail.

### 4.1. The architecture of SRC

The major blocks consisting of the SRC can be divided into these portions according to their functionalities: 1) a control block which supplies the basepoint index $n_k$ and the fractional interval $\mu_k$; 2) the computation structure, Farrow structure, to calculate the SRC output based on the fractional interval input.

A numerically controlled oscillator (NCO) is often used to determine $n_k$ and $\mu_k$ for the SRC. The NCO is simply a digital accumulator with an external input referred to as frequency control word (FCW). In the SRC, FCW is determined by the ratio of the input sample rate ($F_{in}$) and the output sample rate ($F_{out}$):

$$FCW = \left[ \frac{F_{in}}{F_{out}} \right]_N , \qquad (6)$$

where $[]_N$ denotes N-bit quantization. The parameter N determines the frequency resolution in the SRC. In this paper, it is assumed that $F_{in}$ is less than $F_{out}$ and therefore an interpolation is performed in the SRC rather than decimation. In a digital modulator, it would be unnecessary to interpolate the signal to a sample rate higher than the final output sample rate. The value of the NCO register represents the sampling phase offset between the original sample instant and the output sample instant. Therefore it will be used to derive the fractional interval $\mu_k$ being used in the Farrow structure.

It can be seen that the value of NCO register will overflow at an effective rate of $F_{in}$. Reference [2] and [3] described a method to use the most significant bit (MSB) of the NCO register value as a clock with a frequency of $F_{in}$ to clock the delay lines in the Farrow structure. Since $F_{in}$ is the output sample rate of the multistage interpolation filter, it should be an integer multiple of the input symbol rate. Therefore this clock can be further divided to clock the different stages in the multistage interpolation filter according to their interpolation ratio with respect to the original symbol rate.

This technique implicitly assumes that the MSB of the NCO register value should toggle when the NCO register value overflows. This toggling only happens when $F_{out}$ is at least twice higher than $F_{in}$. Therefore the DAC has to be running at the sample rate at least twice higher than the output sample rate from the multistage interpolation filters. Two examples are given in Figure 6, the first NCO has its $F_{in}$ set at 12.5MHz, and the second NCO has its $F_{in}$ set at 87.5 MHz. The $F_{out}$ is 100 MHz in both NCOs. It can easily be seen that the MSB of the NCO register value toggles only in the first example, and can therefore be used as the clock of frequency $F_{in}$. In addition, using different clocks for different stages of the interpolation filter chain makes it very difficult for software implementation.

As mentioned before in this section, the NCO register value shall overflow at an equivalent rate of $F_{in}$, therefore this overflow signal can be used to “clock” the delay line for the Farrow structure. This signal can be further divided by M (the aggregate interpolation ratio of the upsampling filters) in frequency, generating a “request for symbol” trigger signal for the interpolation filter chain. A sample buffer is introduced between the interpolation filter chain and the SRC, in order to accommodate the jitter introduced by the NCO [6]. Since the upsampling filters will write M samples to the buffer for every M samples being fetched from the buffer by the SRC, the buffer’s fill level stays at a balanced state without experiencing overrun or underrun. The described SRC architecture is illustrated in Figure 7. This architecture essentially decouples the upsampling operations from the final output sample rate: the upsampling filter block shall provide M samples to the SRC upon every “request for symbol” received from the SRC’s NCO. Therefore the upsampling operations can be implemented by software without providing sample clock for different filter stages. The control logic is divided into two subblocks: the “buffer read control” is used to generate the basepoint index $n_k$; the “fractional interval extraction” is used to calculate the fractional interval $\mu_k$.

In the next 2 subsections, the “buffer read control” and “fractional interval extraction” will be discussed for two cases: one uses even number of basepoints; the other uses odd number of basepoints. The Farrow structure is common for both cases as Section 3 described.

### 4.2. Control for SRC using even number of basepoints

The SRC using even number of basepoints for its interpolation has its basepoint index $n_k$ derived from [6]:

$$n_k = \mathrm{int}[kT_{out} / T_{in}], \qquad (7)$$

where int[z] denotes the largest integer not exceeding z. Also its fractional interval $\mu_k$ is defined as [6]:

$$\mu_k = kT_{out} / T_{in} - n_k , \qquad (8)$$

Since $n_k$ is always less than $kT_{out}/T_{in}$, it can be seen that $0 \le \mu_k < 1$ in this case. When the implementation of the SRC is considered, the NCO register value can be written as:

$$NCO'_k = k\frac{F_{in}}{F_{out}} = k\frac{T_{out}}{T_{in}} , \qquad (9)$$

where NCO'$_k$ represents the NCO register value at the k$^{th}$ output sample instant without modulo operation. Define NCO$_k$ as the NCO register value at the k$^{th}$ output sample instant after modulo operation, or NCO$_k$ equals to NCO'$_k$ modulo 1. Therefore the basepoint index can be derived by:

$$n_k = n_{k-1} + \text{int}[NCO_{k-1} + FCW]. \qquad (10)$$

It can be seen that int[NCO$_{k-1}$+FCW] equals to the carry out signal from the accumulator, and thus denoted as CO$_k$. In Figure 7, this signal is named as "increase n$_k$". Equation (8) can be rewritten as:

$$\begin{aligned}\mu_k &= k\frac{T_{out}}{T_{in}} - n_k \\ &= (k-1)\frac{T_{out}}{T_{in}} + \frac{T_{out}}{T_{in}} - (n_{k-1} + CO_k) . \qquad (11) \\ &= (\mu_{k-1} + FCW) - CO_k = NCO_k\end{aligned}$$

Therefore the fractional interval μ$_k$ is simply the NCO register value NCO$_k$. According to Equation (10) and (11), the "buffer read control " and "fractional interval extraction" can be easily implemented as shown in Figure 8. As this figure suggests, only the L MSBs from the NCO register value is often used as the fractional interval μ$_k$ for the SRC calculation.

### 4.3. Control for SRC using odd number of basepoints

According to reference [4], the SRC using odd number of basepoints for its interpolation can derive their basepoint index n$_k$ from:

$$n_k = round[kT_{out} / T_{in}], \qquad (12)$$

where round[z] performs the common rounding operation on z. The fractional interval for this case can also be derived from Equation (8). Due to the rounding operation, the fractional interval ranges between –0.5 and +0.5 instead.

For this type of SRC, the derivation of basepoint index n$_k$ is more complex than replacing the int[] operation in Equation (10) with round[] operation. This can be explained more clearly with the help of two following examples. Without losing generality, it is assumed that the SRC is performed in the interval of [0.5, 1.5). Assume NCO register value equals to 0.7 at the current output instant; therefore the current basepoint index should equal to 1 by rounding 0.7. In one example, FCW is assumed to be 0.2, which means the output sample rate is 5 times of the input sample rate. At the next output sample instant, the NCO register value equals to 0.9, therefore the next basepoint index n$_k$ should be 1 as well. In the other example, FCW is assumed to be 0.4. Follow the same procedure; the next basepoint index nk should stay at 1 as well. On the contrary, using "revised" Equation (10) for both examples will give us erroneous result: the basepoint index will be incremented to 2 in both examples.

After close investigation of the problem, it can be concluded the increment of the basepoint index used in this type of SRC is dependent on: 1) if the previous NCO register value (NCO$_{k-1}$) is greater than or equal to 0.5; 2) if the current carry out value (CO$_k$) is equal to 1; 3) if the current NCO register value (NCO$_k$) is greater than or equal to 0.5. Table 1 summarizes all the possibilities of these three factors and whether the basepoint index n$_k$ should be incremented. The aforementioned examples (a) and (b) are corresponding to case 3 and case 1 respectively. It is worth mentioning that there are only 6 cases summarized in Table 1 instead of 8, which is the number of all the possibilities for 3 input variables. The reason is that the frequency control word is assumed to be between 0 and 1 for the interpolation purpose. Therefore in case 3, NCO$_k$ can only be greater than 0.5; while in case 4, NCO$_k$ can only be less than 0.5.

| Case # | NCO$_{k-1}$ >0.5 | CO$_k$ | NCO$_k$> 0.5 | Increase n$_k$ |
|---|---|---|---|---|
| 1 | Y | 1 | N | 0 |
| 2 | Y | 1 | Y | 1 |
| 3 | Y | 0 | Y | 0 |
| 4 | N | 1 | N | 1 |
| 5 | N | 0 | Y | 1 |
| 6 | N | 0 | N | 0 |

**Table 1 Buffer control logic for SRC using odd number of basepoints**

Since the value of NCO register ranges between 0 and 1, the MSB of this value indicates whether it is greater or less than 0.5. Thus the logic equation can be written as:

$$\begin{aligned}Inc\_n_k &= !MSB_{k-1} \bullet MSB_k \\ &+ CO_k \bullet (!MSB_{k-1} + MSB_k)\end{aligned} , \qquad (13)$$

where **!** denotes the logic negate operation. Besides the basepoint index n$_k$, another control parameter for SRC is the fractional interval μ$_k$. Based on Equation (8) and (12), it's easy to derive that

$$\mu_k = kT_{out} / T_{in} - round[kT_{out} / T_{in}] . \qquad (14)$$

It can be seen that μ$_k$ is positive when NCO$_k$<0.5, and μ$_k$ is negative when NCO$_k$ ≥0.5. If the normal two's complement numbering system is used, it can be verified that Equation (14) can be realized by prepending the MSB to the extracted L MSBs of the NCO register value.

Figure 9 shows one possible implementation of the control technique described in this section.

## 5. A DIGTAL MODULAOR EXAMPLE USING SRC

The presented SRC was demonstrated using a dual channel QAM modulator application. The modulator application is based on one digital video broadcasting system: ITU J.83 Annex B, which specifies the symbol rate of ~5.057 Msps (million symbol per second) for 64QAM modulation and ~5.36 Msps of 256QAM modulation. The output DAC is chosen to be running at 100 MHz sample rate. The upsampling filter stage interpolates the signal to 16 times the symbol rate, i.e., ~80.91 Msps for 64QAM channel and ~85.76 Msps for 256QAM channel respectively. A polynomial based SRC using 3 basepoints is selected to further convert the sample rate to 100 Msps from the two above mentioned sample rates after the 16× interpolation. This SRC is selected to satisfy both the adjacent channel power rejection (ACPR) and the received error vector magnitude (EVM). Both the upsampling filters and the SRC are implemented using Nallatech-Xilinx's XtremeDSP development board with a two million gate VirtexII FPGA device XC2V2000. The measured EVM performance is shown in Figure 10. Both channel demonstrate greater than 65 dB ACPR and less than 1% EVM.

## 6. CONCLUSION

This paper presented a sample rate converter (SRC) for digital modulator applications. Despite many other possible filter choices, polynomial based SRC is discussed in detail. This is due to the existence of a hardware efficient implementation of polynomial based SRC, namely the Farrow structure. Besides the Farrow structure used for calculating SRC output, the SRC needs to be provided the basepoint index $n_k$ and fractional interval $\mu_k$ for proper control of the sample rate conversion. The architecture presented in this paper decouples the SRC from the previous interpolation filter stages, therefore allowing these interpolations to be implemented using software based approaches. In addition, this paper described in detail how the SRC is controlled when they involve either even number of basepoints or odd number of basepoints. In the end, a QAM modulator example using the proposed SRC is given and the measure performance is presented as well.

## 7. REFERENCES

[1] T. Hentschel, and G. Fettweis, *Sample Rate Conversion for Software Radio*, IEEE Communication Magazine, pp. 142-150, August 2000.

[2] J. Vankka, et. al., *A GSM/EDGE/WCDMA Modulator With On-Chip D/A Converter for Base Stations*, IEEE Tran. On Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 49, No. 10, pp. 645-655, Oct. 2002

[3] K.H. Cho, and H. Samueli, *A Frequency-Agile Single-Chip QAM Modulator with Beamforming Diversity*, IEEE Journal of Solid-State Circuits, Vol. 36, No. 3, pp. 398-407, March 2001.

[4] Z. Ye, *Linear Phase Lagrange Interpolation Filter Using Odd Number of Basepoints*, Proceedings ICASSP'03, pp. VI - 237-240, April 2003

[5] L. Erup, F.M. Gardner, and R.A. Haris, *Interpolation in Digital Modems – PartII: Implementation and performance*, IEEE Trans. On Communications, Vol. 41, No. 6, pp. 998-1008, June 1993

[6] F.M. Gardner, *Interpolation in Digital Modems – PartI: Fundamentals*, IEEE Trans. On Communications, Vol. 41, No. 3, pp. 501-507, March 1993

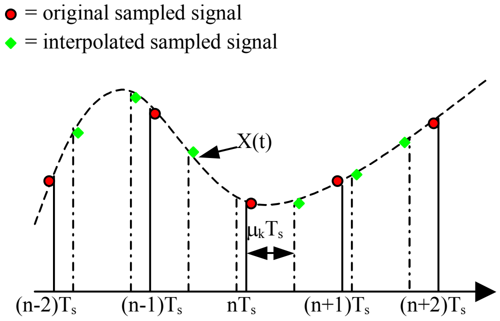


**Figure 1 Interpolation in time domain**

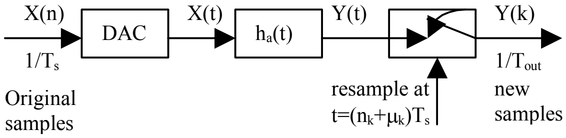


**Figure 2 Analysis model for SRC**

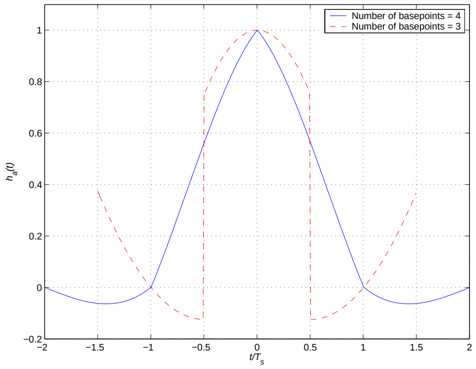


**Figure 3 Impulse response examples of Lagrange polynomial interpolation filter (N=4 and N=3)**

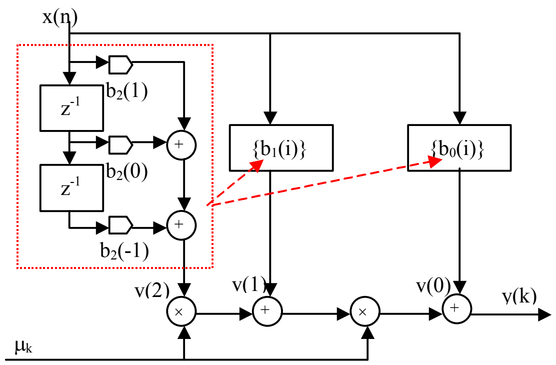


Figure 4 Farrow structure for Equation (5)

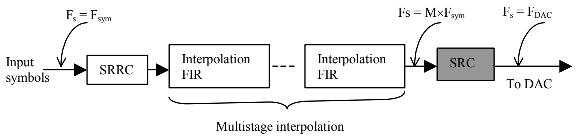


Figure 5 Digital modulator architecture

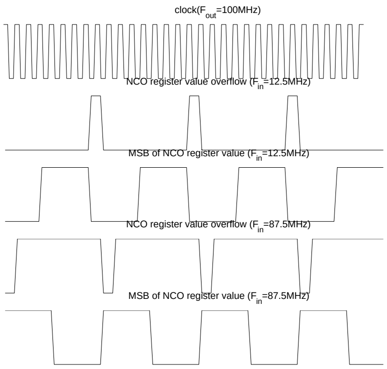


Figure 6 NCO examples with different FCW

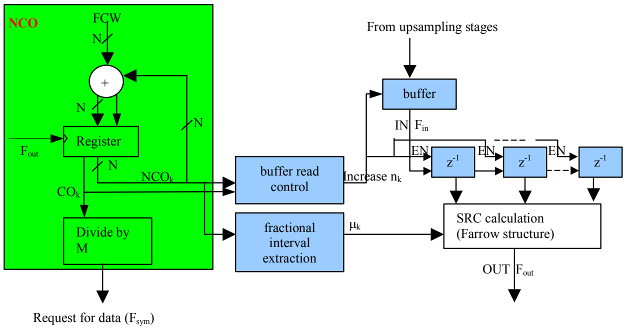


Figure 7 Sample rate converter architecture

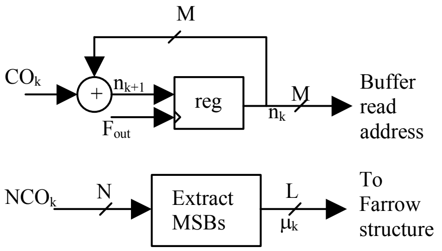


Figure 8 Control for SRC (using even number of basepoints)

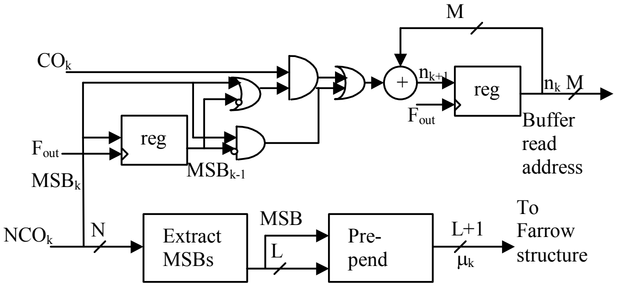
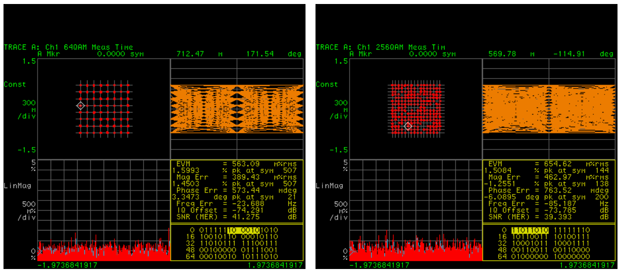


Figure 9 Control for SRC (using odd number of basepoints)



```
TRACE A: Ch1 64QAM Meas Time
  A Mkr          0.0000 sym        712.47   m      171.54   deg
```

```
 EVM         =  563.09    m%rms
1.5993       %  pk at sym   507
 Mag Err     =  389.43    m%rms
1.4503       %  pk at sym   507
 Phase Err   =  573.44    mdeg
3.3473   deg pk at sym    21
 Freq Err    =  -23.688    Hz
 IQ Offset   =  -74.291    dB
 SNR (MER)   =  41.275     dB
```

```
 0  01111110  00101010
16  10010110  00010110
32  11010111  11100111
48  00100000  01111001
64  00010010  10111010
```

```
TRACE A: Ch1 256QAM Meas Tim
  A Mkr          0.0000 sym        569.78   m     -114.91   deg
```

```
 EVM         =  654.62    m%rms
1.5084       %  pk at sym   144
 Mag Err     =  462.97    m%rms
-1.2551      %  pk at sym   138
 Phase Err   =  763.52    mdeg
-6.0895  deg pk at sym   200
 Freq Err    =  -85.187    Hz
 IQ Offset   =  -73.785    dB
 SNR (MER)   =  39.393     dB
```

```
 0  11011010  11111110
16  10110011  10100111
32  10001011  10001111
48  00110011  00110000
64  01000000  10000000
```

Figure 10 EVM measurement from digital modulator with SRC (left-64QAM, right-256QAM)