



Hideo Okawara's Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 53 $\pi/4$ DQPSK Modulation

*ADVANTEST Corporation
November 2013*

Preface to the Series

ADC and DAC are the most typical mixed signal devices. In mixed signal testing, an analog stimulus signal is generated by an arbitrary waveform generator (AWG) which employs a D/A converter inside, and an analog signal is measured by a digitizer or a sampler which employs an A/D converter inside. The stimulus signal is created using mathematical methods, and the measured signal is processed with mathematical methods, extracting various parameters. It is based on digital signal processing (DSP) so that our test methodologies are often called DSP-based testing. Test/application engineers in the mixed signal field should have a thorough knowledge about DSP-based testing. FFT (Fast Fourier Transform) is the most powerful tool here. This lecture will deliver a series of fundamental knowledge on DSP-based testing, especially FFT and its related topics. It will help them comprehend what the DSP-based testing is and its assorted techniques.

Editor's Note

For other articles in this series, please visit the Advantest website at:
<https://www.advantest.com/US/ADV006722>

Preface

Digital modulation/demodulation is today's standard in communication systems. There are two categories in digital modulation. One is a single carrier modulation and the other is multi-carrier modulation. Although the single carrier modulation is already out-of-date, there are many interesting techniques used in it, so the knowledge of such classic standard would be useful for daily testing jobs. One of the traditional modulation methods " $\pi/4$ shifted DQPSK (Differential Quadrature Phase Shift Keying)" is discussed in this article.

$\pi/4$ Shifted DQPSK

The $\pi/4$ Shifted DQPSK is one of the QPSK modulation types. Old portable telephone systems of PDC (Personal Digital Cellular) and PHS (Personal Handy-phone System) employ this modulation. Figure 1 illustrates the octagonal constellation points of this modulation, which contains two square constellations overlaid with the axis shift of $\pi/4$ radian. For convenience of identification, each point is numbered from #0 at the coordinate of (1, 0) to #7 at $(\sqrt{2}, -\sqrt{2})$ here. "D" in DQPSK stands for "differential phase", which means that transition of constellation points conveys information. A data stream is sliced every two bits and dealt with a unit of di-bit that contains "00", "01", "11" and "10". Figure 2 illustrates how the current location #7 jumps to the next point according to the coming di-bit. There are four routes available. Based on the data "00", "01", "11" and "10", the current location (red) moves to the next location rotated by +1, +3, +5 or +7 times $\pi/4$ radian respectively. For example, if the coming data is "01", which means +3 units of shift or rotation, the current location #7 will move to the location #2. ($7+3-8=2$)

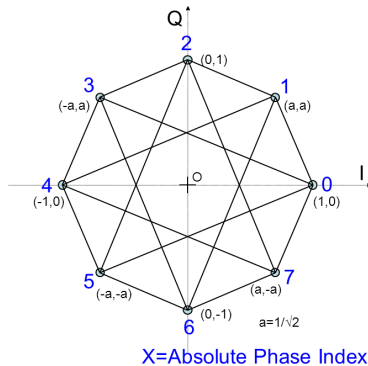


Figure 1: Constellation Points

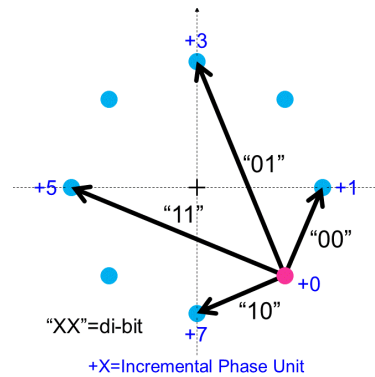


Figure 2: Transition Routes

Figure 3 illustrates examples of the phase transition routes. The distance from the origin to the locus represents the amplitude of a modulated signal. Figure 3 also shows that the locus never goes through the origin, meaning the amplitude of the modulated signal does not change so radically. Therefore, power amplifiers employed in this modulation require less linearity than a regular QPSK, whose locus goes through the origin. This is an advantage of the $\pi/4$ Shifted DQPSK.

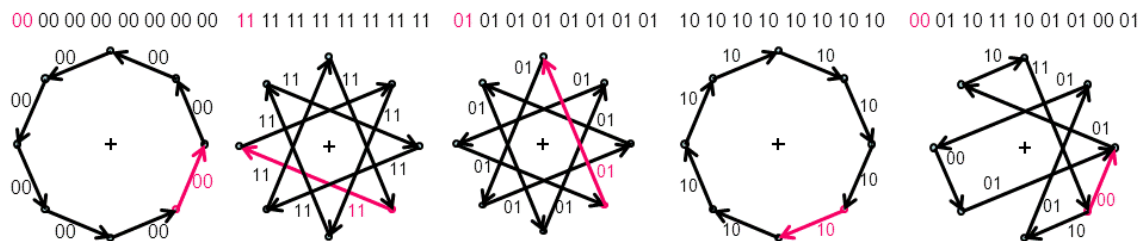


Figure 3: Example Transition Routes (red=start)

Modulation Scheme

In order to demonstrate the modulation with using the MB-AV8 in the V93000 system, the example condition is settled as follows;

- Data PRBS 512 bits
- Di-bit 256 symbols (= 512 / 2)
- Symbol Rate 192 k baud (k symbols/sec)
- Roll-off Filter Roll-off Factor $\alpha=0.5$, Raised-cosine $\sqrt{}$ Nyquist
- IF Carrier 1.536 MHz (= 8 x 192 k baud)

The carrier frequency of the IF (Intermediate Frequency) is set as a convenient value for the resource. It is not the case of real cellular phone systems. The HF AWG in the MB-AV8 generates the modulated signal. The resource condition is as follows:-

- AWG Sampling Rate 49.152 Msps (= 32 x 1.536 MHz)
- AWG Waveform Length 65536 words (= 256 words x 256 symbols)

Figure 4 illustrates the whole scheme of the waveform generation. The 512 bits of PRBS data is generated for test purpose. Firstly, it is sliced every 2 bits, which is called "di-bit." Each di-bit selects one of the four transition routes according to the rule depicted in Figure 2, so the 256 constellation points would be defined. This is a differential phase shift modulation so that the starting location of phase is not important. Each one of the symbol points has a set of x and y coordinates which is depicted in Figure 1. After the mapping of di-bits, the (x, y) data is split into the I/Q trains of impulses. The symbol rate is 192 k baud. So the impulses are aligned at the rate of 192 kHz. Eventually a modulated IF waveform is loaded in the AWG so that the I/Q data of impulse trains are over-sampled at the pace of the AWG that is 49.152 Msps. The I/Q impulse trains are rolled off by the root raised-cosine filter in order to limit the bandwidth of the final modulated signal. The frequency limited I/Q waveforms are multiplied by the cosine and sine IF waveforms respectively, and then they are summed together making a single waveform, which is appropriately scaled to the full-scale range of the AWG.

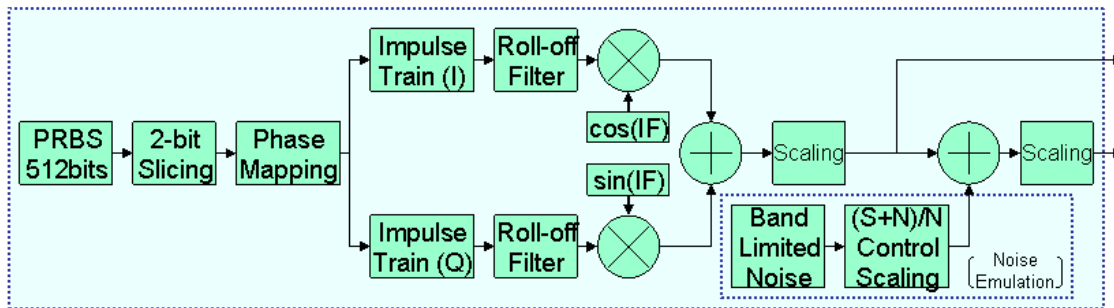


Figure 4: Modulated Waveform Generation Scheme

Test signals should emulate the mission mode condition for giving stress to the device under test (DUT). So, there are additional blocks included in Figure 4 in the right bottom corner. This part is provided to make the signal look more realistic adding a band limited noise.[1] It is not essential for the modulation processing. Actually, not only noise but also jitter may be injected in the signal.[2]

Waveform Generation Process

Let us now take a look at the processing of the waveform generation according to the schematic in Figure 4. The test signal is the 512-bit PRBS, which can be generated by the 9-bit Linear Feedback Shift Register (LFSR). The 9-bit LFSR generates $2^9-1=511$ bits of data, containing 256 "1" and 255 "0". In order to make it convenient for di-bit structure, the size of the input bit stream should be an even number, so one more "0" is added to the end of the stream.

Figure 5 depicts the top and tail parts of the 512-bit PRBS, 256 di-bit and its absolute phase index trend which is interpreted from the di-bit stream. The initial phase is supposed to be located at #0, and the following indexes of the 256 di-bits go as illustrated. The phase index ends as #4 which is different from the starting location #0. So if this phase index locus is continuously looped, the data conveyed in the stream would be slightly altered from the second loop. The transition from #4 to #5 is +1, so the di-bit is "00". So the stating data "11 11 11 11" would be changed as "00 11 11 11" from the second loop. However, the embedded digital data is not essential in this experiment, because the modulated waveform itself and how dense the constellation points concentrate are evaluated.

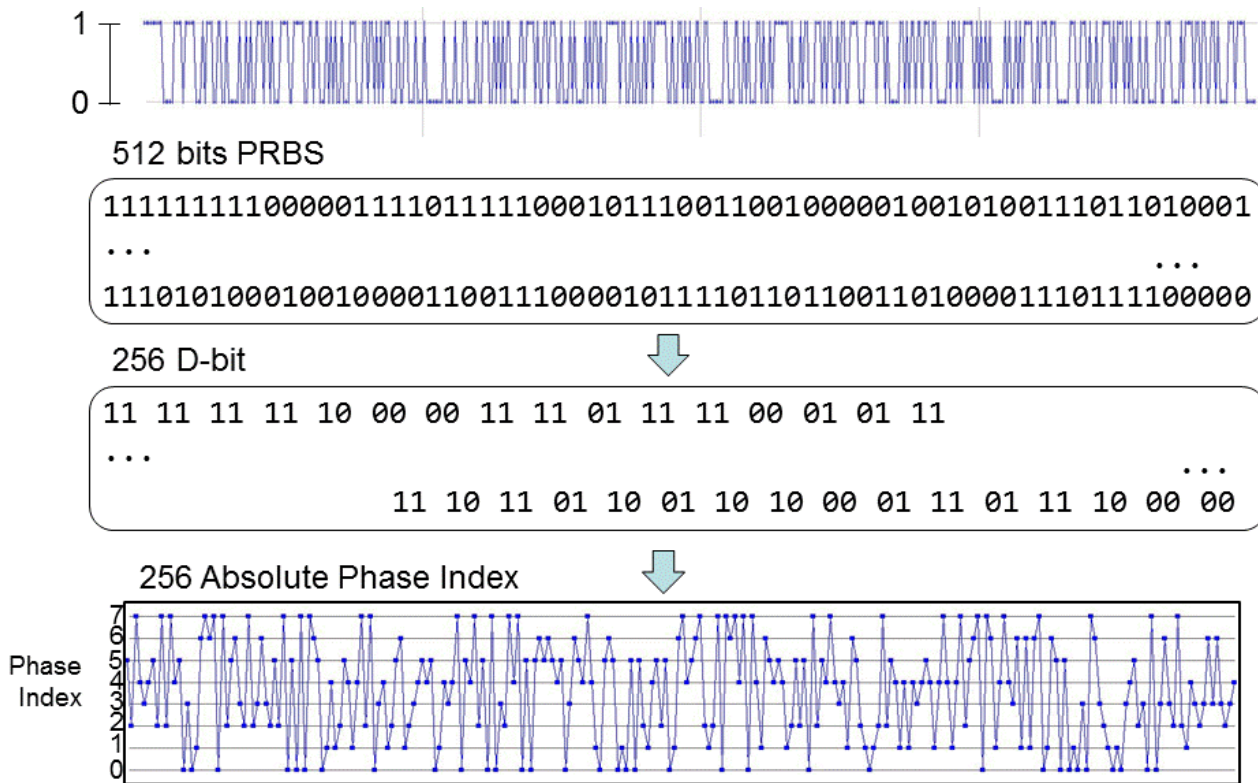


Figure 5: 512-bit PRBS → 256 Di-bit → Phase Index

The phase index in Figure 5 means the (x, y) address of the constellation points, which can be separated to the I-signal (x array) and the Q-signal (y array). Then the I/Q-signals are transformed to the impulse waveforms as illustrated in Figure 6. The time step is at 49.152 Msps. The amplitude of impulses is either 1.0 or $1/\sqrt{2}$. The impulses exist at the rate of 192 kHz. The space from one impulse to the next is filled up with 0 in the I/Q-signal arrays.

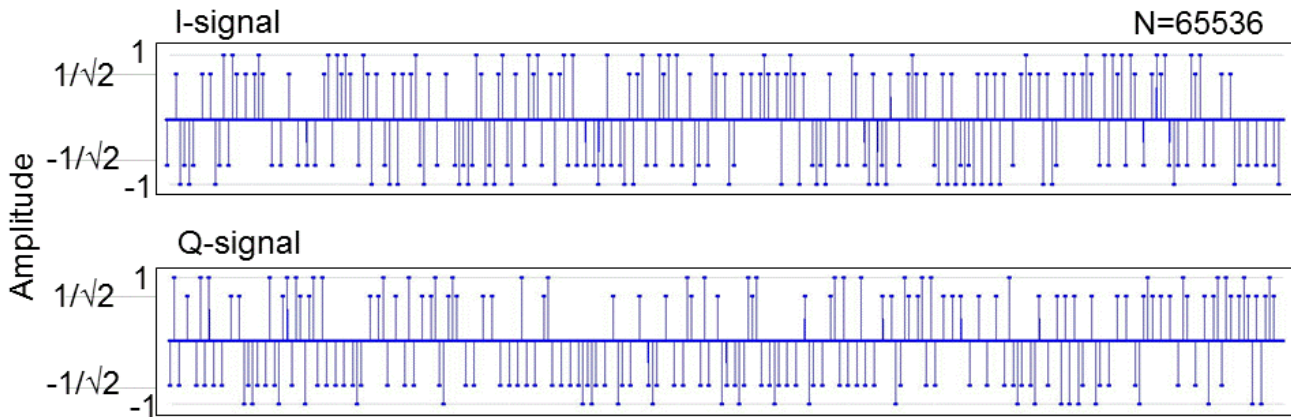


Figure 6: I/Q Impulse Trains

The impulse trains are filtered by the raised cosine "root" Nyquist filter. Figure 7 illustrates the raised cosine Nyquist characteristics with the roll-off factor $\alpha = 0.5$. In the system, this frequency response is square-rooted and evenly compromised in the modulation side and the demodulation side. While the signal goes through a modulator and a demodulator, the combined filter response completes the raised cosine characteristics because $\sqrt{a} \cdot \sqrt{a} = a$.

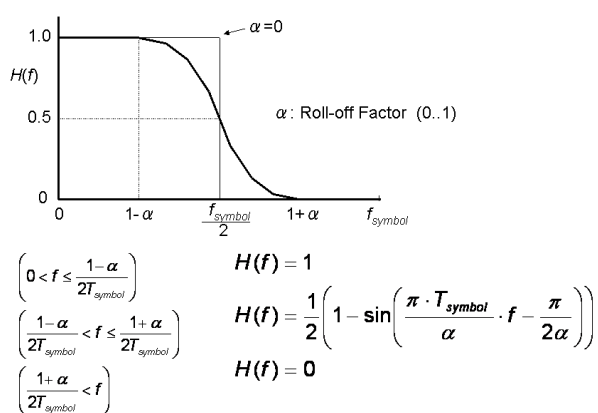


Figure 7: Raised Cosine Characteristics

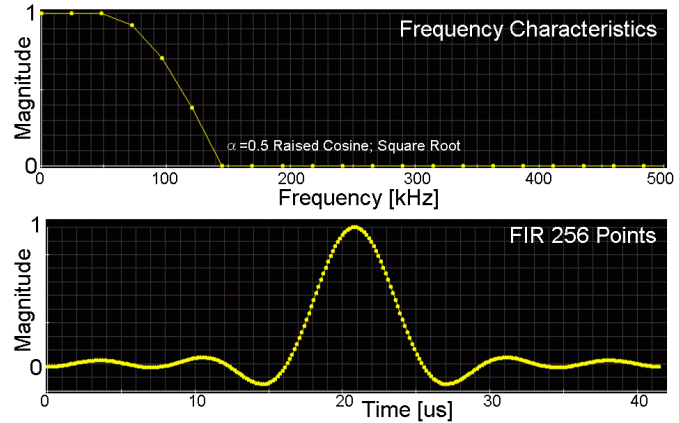


Figure 8: Square Root Response and FIR

The roll-off response of the raised cosine in Figure 7 is square rooted and transformed to the finite impulse response (FIR) as illustrated in Figure 8. Firstly, the root Nyquist response is designed in the frequency domain, and transformed to the time domain waveform by using the inverse FFT (IFFT). For more information about FIR generation, look at the archived article.[3]

By convoluting the FIR curve to the impulse trains in Figure 6, the rolled-off I/Q-signal waveforms are generated as Figure 9 shows. Figure 10 displays the I/Q-signals in the 2-dimensional coordinate. This is the I-Q combined signal locus. You can see the locus does not cross over the origin. As you may notice, the eight constellation points at this stage are not well converged. This is because the roll-off characteristics is square rooted or halved so that the raised-cosine response is not yet completed.

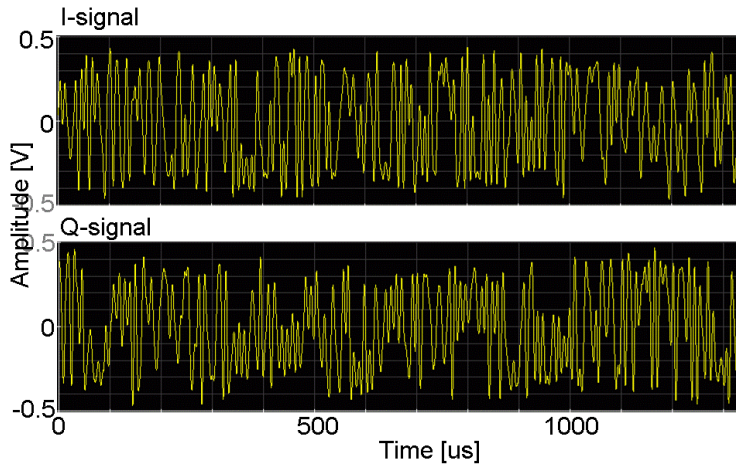


Figure 9: Rolled-off I/Q Signals

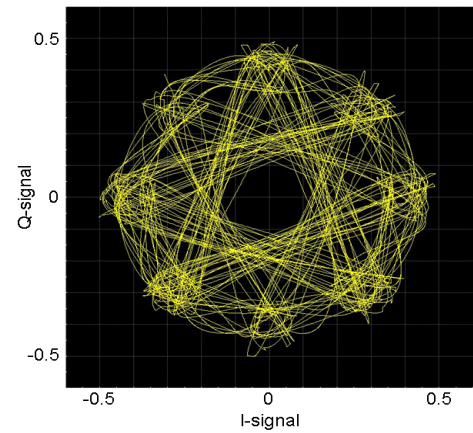


Figure 10: I-Q Signal Locus

The rolled-off signals are multiplied to the orthogonal IF carriers respectively. The carriers are expressed $\cos(IF)$ and $\sin(IF)$ in Figure 4. The 2048 cycles of cosine and sine waveforms are generated and multiplied to the I/Q-signals respectively, and then they are summed up together making a single IF waveform, which is shown in Figure 11. The amplitude is scaled to ± 0.5 V which is ready for downloading to the AWG. As you can see the envelope of the combined waveform does not cross over the zero level, meaning the amplitude change is limited, so the amplifier in the real device requires less linearity. Figure 12 shows the FFT spectrum of the modulated signal. The carrier 1.536 MHz is modulated and the occupied band-width is approximately 300 kHz.

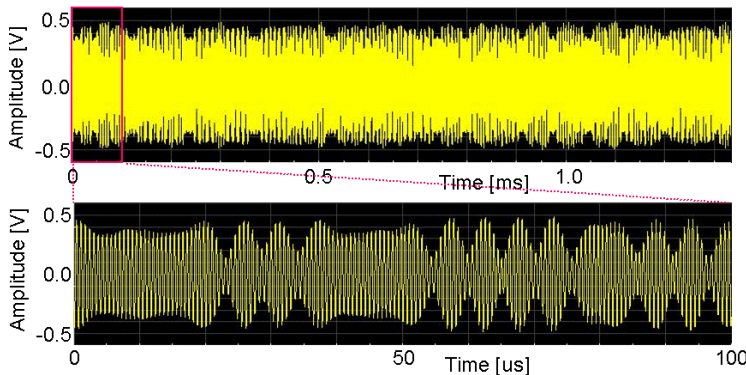


Figure 11: IF Modulated Signal

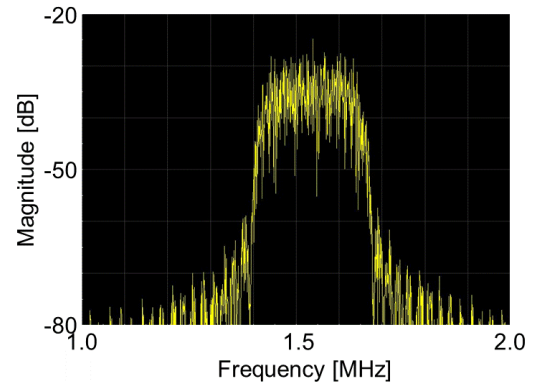


Figure 12: Mod. Signal Spectrum

As previously mentioned, the block diagram in Figure 4 includes blocks for adding a band-limited noise. If you need to use a S/N controlled modulated signal, you can add band-limited noise to the waveform shown in Figure 11 by using the method described in the archived article.[1]

Reference

- [1] DSP-Based Testing Fundamentals 32 "Noise Emulation"
- [2] DSP-Based Testing Fundamentals 30 "Jitter Injection"
- [3] DSP-Based Testing Fundamentals 14 "FIR Filter"

Example Source Code

The subroutine:

```
200: // Subroutine for PRBS Data Array
201
202: int iPRBS9(int iFlag) // 9-bit LFSR; PRBS Generation
203: {
204:     static int iSR;
205:
206:     if (iFlag) iSR=(((iSR&16)^(iSR&1)<<4)<<5|iSR)>>1;
207:     else      iSR=511; // Seed "11111111"
208:
209:     return (iSR&1);
210: }
211:
```

The main routine:

```
11:
12: INT     i,j,k,m;
13: INT     Nbits=512; // Even Number
14: INT     Nsymbols=Nbits/2; // Symbol Size
15: ARRAY_I iBit; // Data Container
16:
17: iBit.resize(Nbits);
18: for (i=0;i<(Nbits-1);i++) iBit[i]=iPRBS9(i);
19: iBit[Nbits-1]=0;
20:
21:
22:
23: // Symbol Phase Allocation
24: INT     iPabs;
25: ARRAY_I iPhase;
26: iPhase.resize(Nsymbols); // Symbol Index Space
27:
28: j=0; // iPhase[]: (0,1,2,3,4,5,6,7)
29: iPabs=0; // Initialize Absolute Phase Index
30: for (i=0;i<Nbits;i=i+2) {
31:     if ((iBit[i]==0)&&(iBit[i+1]==0)) iPabs=iPabs+1;
32:     else if ((iBit[i]==0)&&(iBit[i+1]==1)) iPabs=iPabs+3;
33:     else if ((iBit[i]==1)&&(iBit[i+1]==1)) iPabs=iPabs+5;
34:     else if ((iBit[i]==1)&&(iBit[i+1]==0)) iPabs=iPabs+7;
35:     if (iPabs>7) iPabs=iPabs-8; // Absolute Phase Index Adjustment
36:
37:     iPhase[j]=iPabs; j++; // Decided Absolute Phase Index
38: }
39:
```

```

40:
41: DOUBLE   dP4=M_PI/4.0;
42: ARRAY_D  dX,dY;
43:
44: dX.resize(8); dY.resize(8);           // Constellation Address (8 Points)
45: for (i=0;i<8;i++) {                  // Address (dX,dY)
46:     dX[i]=cos(dP4*i);
47:     dY[i]=sin(dP4*i);
48: }
49:
50:
51: DOUBLE   dFsymbol=192.0 kHz;          // Symbol Rate           (192 kHz)
52: DOUBLE   dFif=dFsymbol*8;            // IF Carrier Frequency  (1.536 MHz)
53: DOUBLE   dFawg=dFif*32;              // AWG Sampling Rate     (49.152 MHz)
54: INT      Nawg=Nsymbols*8*32;         // AWG Waveform Size     (8*32=256)
55:
56: ARRAY_D  dI,dQ;                       // I-Q Impulse Train
57: dI.resize(Nawg); dQ.resize(Nawg);
58: m=0;
59: for (i=0;i<Nsymbols;i++) {
60:     k=iPhase[i];                       // Phase Location
61:     dI[m]=dX[k];                       // Impulse I
62:     dQ[m]=dY[k];                       // Impulse Q
63:     m++;
64:     for (j=0;j<255;j++) {              // Zero Insertion (255 "0")
65:         dI[m]=0.0;
66:         dQ[m]=0.0;
67:         m++;
68:     }
69: }
70:
71: INT      Nfir=2048;                    // FIR Length
72: DOUBLE   dFrsln=dFawg/Nfir;           // Frequency Resolution
73:
74: DOUBLE   dTsymbol=1.0/dFsymbol;       // Symbol Period
75: DOUBLE   dRolloff=0.5;                // Roll-off Rate
76: DOUBLE   dF1=(1.0-dRolloff)/(2*dTsymbol); // Roll-off Start Frequency
77: INT      N1=(INT)(dF1/dFrsln);        // Roll-off Start Index
78: DOUBLE   dF2=(1.0+dRolloff)/(2*dTsymbol); // Roll-off Stop Frequency
79: INT      N2=(INT)(dF2/dFrsln);        // Roll-off Stop Index
80: DOUBLE   dR=M_PI/dRolloff*dTsymbol;
81: DOUBLE   dC=M_PI/(2.0*dRolloff);
82: DOUBLE   dFreq,dU;
83: ARRAY_D  dH;
84:
85: dH.resize(Nfir/2);                    // Roll-off Container
86: for (i=0;i<=N1;i++) dH[i]=1.0;       // Pass-band
87: for (i=N1+1;i<=N2;i++) {            // Roll-off-band
88:     dFreq=i*dFrsln;
89:     dU=0.5*(1.0-sin(dR*dFreq-dC));    // Raised cosine
90:     dH[i]=sqrt(dU);                  // Square Root (For TX)
91: }
92: for (i=N2+1;i<(Nfir/2);i++) dH[i]=0.0; // Rejection-band
93:

```



```

100:
101:  ARRAY_D      dFIR;           // FIR Container
102:  ARRAY_COMPLEX CH,CFIR;      // FIR Response Spectrum
103:  DOUBLE       dP;
104:
105:  CH.resize(Nfir);
106:  CH[0]=COMPLEX(dH[0],0.0);
107:  for (i=1;i<(Nfir/2);i++) {
108:      dP= -i*M_PI;
109:      CH[i]=COMPLEX(dH[i]*cos(dP),dH[i]*sin(dP)); // Polar to Rectangular
110:      CH[Nfir-i]=COMPLEX(CH[i].real(),-CH[i].imag()); // Complex Conjugate
111:  }
112:  CH[Nfir/2]=COMPLEX(0.0,0.0);
113:
114:  DSP_IFFT(CH,CFIR);
115:  dFIR=CFIR.getReal(); // FIR Result
116:
117:  DOUBLE dMin,dMax;
118:  INT    iMin,iMax;
119:  DSP_ABS_MINMAX(dFIR,&dMin,&dMax,&iMin,&iMax); // Normalize FIR
120:  DSP_MUL_SCL(1.0/fabs(dMax),dFIR,dFIR);
121:
122:
123:
124:
125:
126:
127:
128:
129:
130:
131:  ARRAY_D dIx,dQx;
132:  DSP_CONVOL(dFIR,dI,dIx,ON); // Roll-off Filtering
133:  DSP_CONVOL(dFIR,dQ,dQx,ON); // Roll-off Filtering
134:
135:  DOUBLE dMax1,dMax2;
136:  DSP_ABS_MINMAX(dIx,&dMin,&dMax1,&iMin,&iMax);
137:  DSP_ABS_MINMAX(dQx,&dMin,&dMax2,&iMin,&iMax);
138:  if (dMax1>dMax2) dMax=dMax1; else dMax=dMax2; // For Baseband I/Q Generation
139:  DSP_MUL_SCL(0.5/fabs(dMax),dIx,dIx); // I signal: Scaling for AWG
140:  DSP_MUL_SCL(0.5/fabs(dMax),dQx,dQx); // Q signal: Scaling for AWG
141:
142:
143:
144:
145:
146:
147:
148:
149:
150:
151:  DOUBLE dT;
152:  DOUBLE dNcycles=(Nawg/dFawg)/(1.0/dFif); // IF Carrier Cycles
153:
154:  ARRAY_D dCOS,dSIN,dwave,dwVxCOS,dwVxSIN;
155:
156:  dCOS.resize(Nawg); dSIN.resize(Nawg);
157:  dP=2.0*M_PI*dNcycles/Nawg;
158:  for (i=0;i<Nawg;i++) {
159:      dT=dP*i;
160:      dCOS[i]= cos(dT); // IF Carrier cos()
161:      dSIN[i]=-sin(dT); // IF Carrier sin()
162:  }
163:
164:  dwave.resize(Nawg);
165:  DSP_MUL_VEC(dIx,dCOS,dwVxCOS); // (I signal) x cos()
166:  DSP_MUL_VEC(dQx,dSIN,dwVxSIN); // (Q signal) x sin()
167:  DSP_ADD_VEC(dwVxCOS,dwVxSIN,dwave); // (Isig*cos)+(Qsig*sin)
168:
169:  DSP_ABS_MINMAX(dwave,&dMin,&dMax,&iMin,&iMax); // IF Modulated Signal
170:  DSP_MUL_SCL(0.5/fabs(dMax),dwave,dwave); // Scaling for AWG
171:

```