Keywords: CDMA, verilog, waveform, transmit

APPLICATION NOTE 918

CDMA Reverse-Link Waveform Generator FPGA for Production Transmit Path Tests

By: Bob Kelly, Senior Applications Engineer
May 01, 2002

Abstract: Maxim has designed an easy-to-build CDMA baseband-modulation generator for circuit evaluation of our various products which design into the cellular handset transmit path. This design combines a high-density programmable logic device, a crystal oscillator, and a pair of matched low-pass filters to provide the desired quadrature output waveforms. The digital circuitry incorporated into a complex programmable logic device (CPLD) is based on the IS95 standard. A 4.9152MHz crystal oscillator drives a CY37256 CPLD to produce the 1.2288MHz digital outputs, which provides an accurate I/Q (in-phase and quadrature) bit-stream to the output low-pass filters. The performance was confirmed by measuring ACPR on the MAX2361 transmitter IC and comparing the results with the ACPR observed by using an Agilent E4433B arbitrary waveform generator as the reference signal source. The lab-measured ACPR was found to agree within 0.5dB.

This application note presents the digital portion of the CDMA reverse link waveform generator, and describes some of the design issues and how they were handled.

Additional Information
- Wireless Product Line Page
- Quick View Data Sheet for the MAX2361
- Applications Technical Support

Introduction

Maxim has designed an easy-to-build CDMA baseband-modulation generator for circuit evaluation of our various products which design into the cellular handset transmit path. This design combines a high-density programmable logic device, a crystal oscillator, and a pair of matched low-pass filters to provide the desired quadrature output waveforms. The digital circuitry incorporated into a complex programmable logic device (CPLD) is based on the IS95 standard. A 4.9152MHz crystal oscillator drives a CY37256 CPLD to produce the 1.2288MHz digital outputs, which provides an accurate I/Q (in-phase and quadrature) bit-stream to the output low-pass filters. The performance was confirmed by measuring ACPR on the MAX2361 transmitter IC and comparing the results with the ACPR observed by using an Agilent E4433B arbitrary waveform generator as the reference signal source. The lab-measured ACPR was found to agree within 0.5dB.

Click here for an overview of the wireless components used in a typical radio transceiver.

This application note presents the digital portion of the CDMA reverse link waveform generator, and describes some of the design issues and how they were handled.

Digital System Description

The Textbook CDMA Generator

Figure 1 shows the block diagram for a CDMA reverse channel generator. The CDMA generator is composed of the following items:

- The digital data source. In a cell phone, which is coded speech data.
- Encoding and interleaving functions.
- A Walsh code generator.
- A 42-bit long PN (pseudonoise) generator of maximal length, refered to here as the "long code".
- 3 modulo-two mixers or exclusive OR gates.
- Two "short code", 15-bit PN maximal length shift registers.
- A one-half chip delay, equal to (813.8ns / 2) or 406.9ns.
- A pair of matched finite impulse response (FIR) low-pass filters.
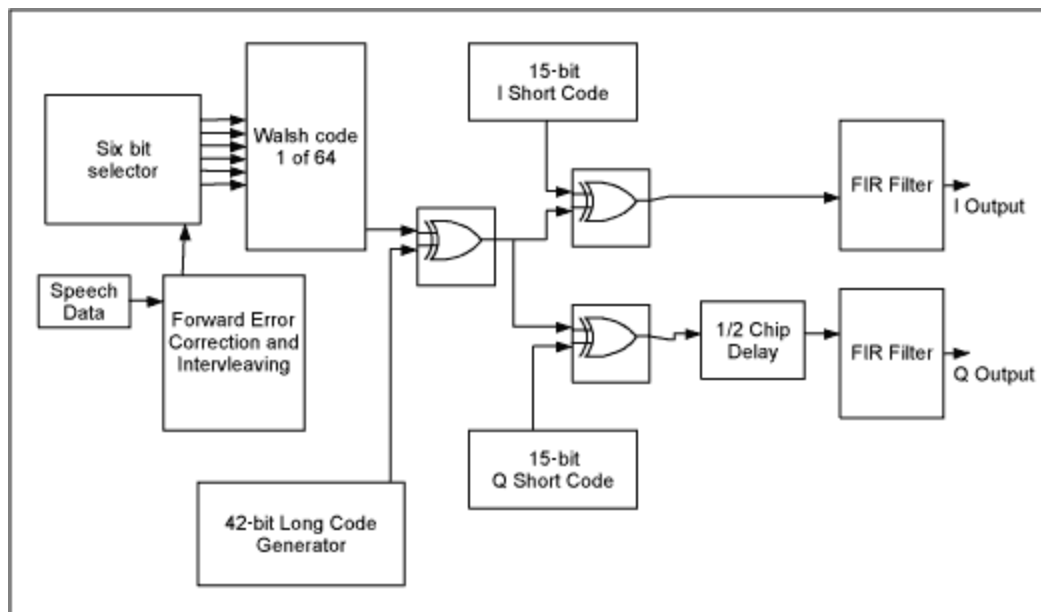

Figure 1. Text book CDMA reverse link generator.

Shortcuts Taken in This Effort

For practical measurement purposes, some coding simplification was possible. The blocks associated with forward error correction (FEC) and the interleaving were eliminated, as their contribution to the spectrum was found to be minimal. To simulate the CELP codec data source, a 7-bit maximal PN generator was used.

The 42-bit long code was implemented as a 31-bit long PN, maximal length shift register. This step was taken to preserve registers in the CPLD and to make design verification faster. A 42-bit PN generator shifting at 1.2288MHz will take ~3.6 million seconds to repeat the cycle. (A simple calculation here will

reveal that your test bench must remain undisturbed for 41 days!) One approach to speed up the verification of a 42-bit PN generator is to run the clock faster. Even if the clock ran at 20MHz, it would still take 2.5 days for the cycle to repeat. A 31-bit PN code, running at 20MHz will repeat in less than 2 minutes which is a more reasonable test time.

The digital (FIR) low-pass filters at the output are replaced by simple inductor and capacitor-based passive filters. This application noteglosses over the complexity and importance of these filters, but for those interested the design was implemented as a 7th-order elliptical low pass at 600kHz, with a phase equalizer section. A key performance point is set for -45dBc at 740kHz, and -65dBc at 881kHz and beyond. The bandwidth of the transmitted spectrum, the rate of roll-off, and the resultant ACPR observed are directly linked to the quality of these filters.

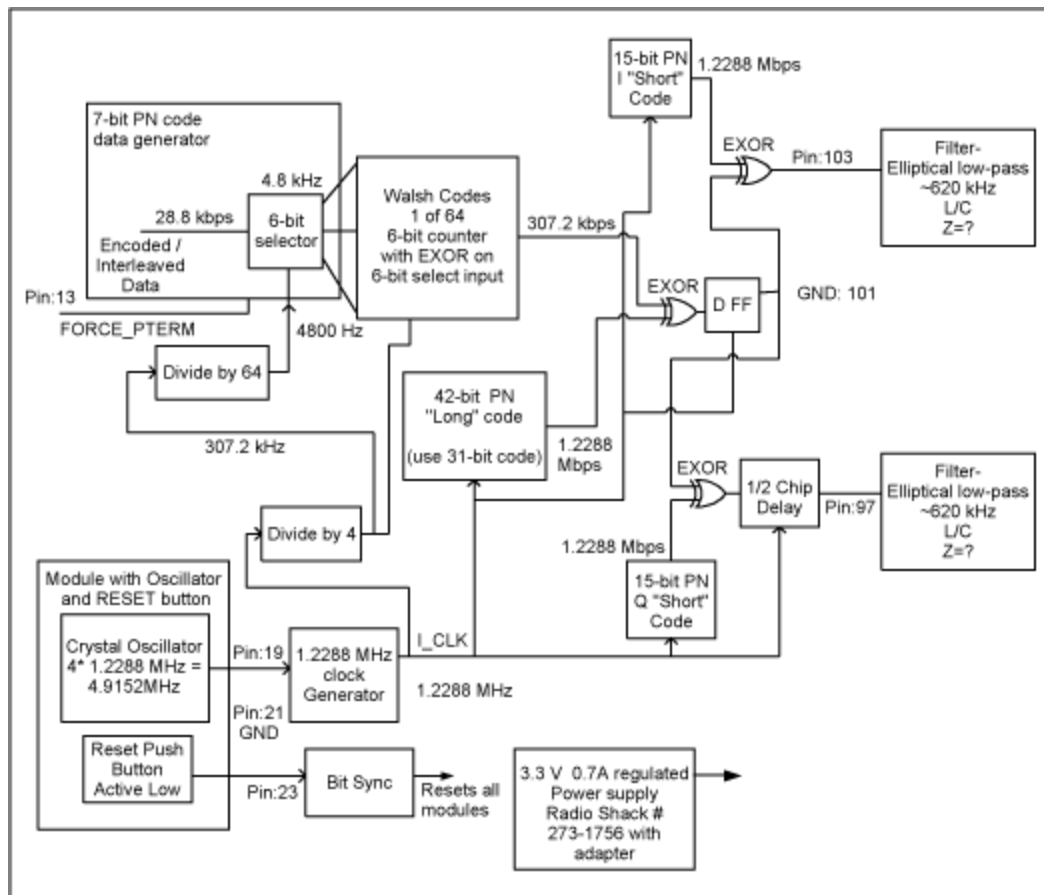**Figure 2** shows the abbreviated block diagram implemented in this effort.

Figure 2. Implemented CDMA reverse link generator.

Design Details

This application note will not attempt to present every detail of the design and implementation. Instead, a few key modules will be used to illustrate design techniques and solutions. It is not the intent to convert the reader into a Verilog expert.

Walsh Code Generator

The Walsh Code generator is usually described in literature in using matrix notation.

$$W_{2n} = \begin{bmatrix} W_n & W_n \\ W_n & \overline{W_n} \end{bmatrix}$$ (Eq. 1)

The construction of the Walsh vectors assumes a starting seed of $W_1 = 0$. The lower right-hand region of the Walsh matrix, shown in Equation 1 by the W_n with a bar on top, implies a bit-wise logic inversion of each entry in the matrix. Each row in the Walsh matrix can be generated with some exclusive or gates and a six-bit counter. Until this observation is made, the Walsh matrix may seem a daunting module to generate with Verilog code and fit in a CPLD. The listing of the Verilog code is included here as a sample only.

```
module walsh( clk, resetn, select, wout);
//Walsh code generator. Selects one out of N = 64.


input clk, resetn;
input [5:0]select; // vector to select which walsh code is generated
output wout;
reg [5:0] cntval;
// intermediate terms to keep output exor size small.
reg [5:0] p ;
reg t01, t23, t45; // these registers are used to pipeline the EXOR
section
reg s0, s1; // more pipeline registers for EXOR
always ? (negedge resetn or posedge clk)
begin
  if(!resetn) // Is it time to reset??
  begin
    cntval <= 0; // initialize the counter register
  end
  else
  begin
    cntval <= cntval + 1; //Warp does an efficient job implementing this.
  end
end
always ? (negedge resetn or posedge clk)
begin
  if(!resetn) // Is it time to reset??
  begin
    p[5:0] <= 0; // initialize all registers associated with this section.
```

```
    t01 <= 0;
    t23 <= 0;
    t45 <= 0;
    s0 <= 0;
    s1 <= 0;
   end
   else
   begin
    p <= cntval & select ;
    t01 <= p[0] ^ p[1] ; // the ^ symbol is the exclusive OR operation.
    t23 <= p[2] ^ p[3] ;
    t45 <= p[4] ^ p[5] ;
    s0 <= t01 ^ t23;
    s1 <= t45;
   end
 end
 assign wout = s0 ^ s1 ; // a last bit of async. Logic to generate the final
 output
 endmodule
```

In the Verilog code above, note the precautions taken to structure the operations and use synchronous logic as much as possible. The use of synchronous techniques helps the design in two regards:

1. Metastability conditions are minimized.
2. The logic maps efficiently into the CPLD architecture.

To perform digital design using modern hardware description languages, the target hardware must always be kept in mind. Since most CPLDs have a regular structure consisting of LOGIC GATES -> REGISTER, it makes for efficient mapping into the CPLD if the Verilog code follows this same structure. In this way, the engineer can guide the synthesis engine. If this is ignored and many levels of asynchronous logic are implied in the Verilog code, the CPLD fitter will expand the logic and quickly exceed the capacity of your target CPLD. This is especially true with exclusive OR functions. The Walsh code generator described by this Verilog code pipelines the exclusive OR functions using intermediate registers, giving a very compact implementation.

The Walsh generator Verilog code also illustrates several key points for reliable design. In the beginning of each "always ?" section a test is made for a reset condition, and then all the associated registers are preset or cleared as needed. Always starting in a known state is a proven design technique for reliable operation. Also illustrated is the separate section to implement the 6-bit counter, and another section is used to create the exclusive OR logic to select the correct Walsh vector to output. Lastly, note the use of comments, shown in Verilog by the // symbols.

PN Generators

The CDMA generator employs four PN generators. There are two basic methods to implement maximal

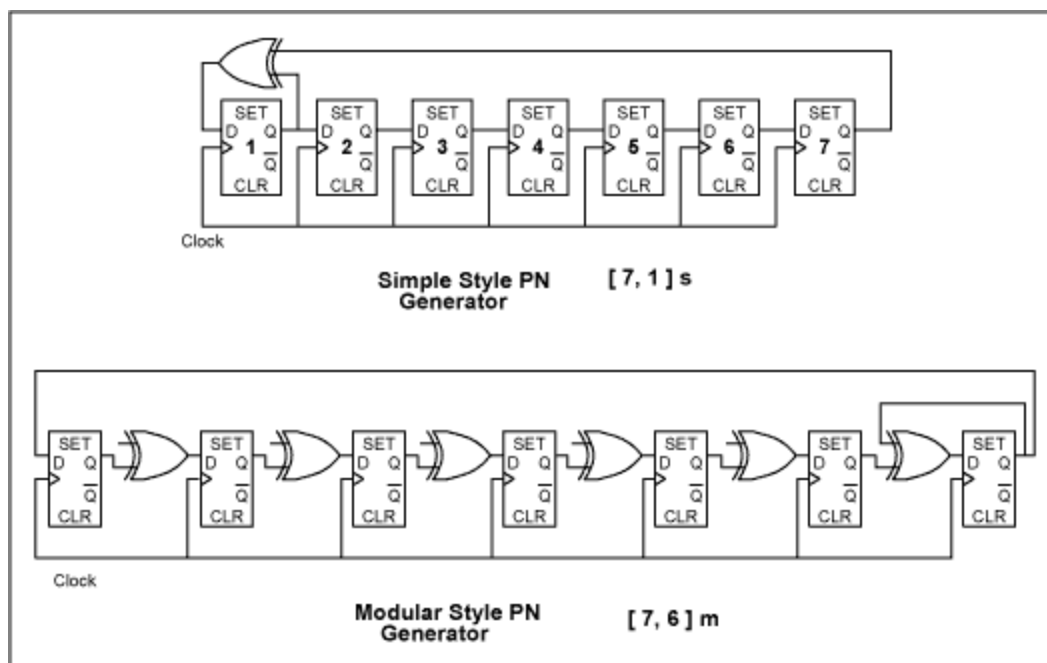length code shift registers: a simple feedback style, and the modular style. (See **Figure 3**)

Figure 3. Examples of simple and modular PN generators.

The simple PN generator is acceptable for use in situations where the code rate (chip) is slow compared to the logic employed and the number of feedback taps is small. As the number of taps grows, the simple approach starts to show limitations, as the delay through multiple levels of asynchronous logic continue to add in series and limit the maximum clock speed that can be used. The simple PN generator is used for simulating random data in this design, being clocked at 4.8kHz.

The modular PN generator uses more logic gates, as the EXOR operation is performed in parallel at each stage of the linear register. The penalty is not too severe in CPLD implementations, as the Verilog code is written to use the EXOR construct when needed, and a simple D type flip-flop elsewhere.

The CDMA-specific polynomials used in this design are:

Short Code I:

$$I(X) := X^{15} + X^{13} + X^9 + X^8 + X^7 + X^5 + 1$$

Short Code Q:

$$Q(X) := X^{15} + X^{12} + X^{11} + X^{10} + X^6 + X^5 + X^4 + X^3 + 1$$

Long Code:

$$LC(X) := X^{42} + X^{35} + X^{33} + X^{31} + X^{27} + X^{26} + X^{25} + X^{22} + X^{21} + X^{19} + X^{18} + X^{17} + X^{16} + X^{10} + X^7 + X^6 + X^5 + X^3 + X^2 + X^1 + 1$$

The Verilog code to implement the short I code follows:

```verilog
module i_code_s( clock, resetn, i_code_out);
// Generate the 15-bit PN code using the polynomial
// x^15 + x^13 + x^9 + x^8 + x^7 + x^5 + 1

input clock, resetn;
output i_code_out;
reg [15:1]pi;
always ?(posedge clock or negedge resetn)
  if(!resetn)
    begin
    pi <= 15';b111111111111111; // init the shift register with ones
    end
  else
    begin // here starts the modular shift register
    pi[1] <= pi[15];
    pi[5:2] <= pi[4:1] ;
    pi[6] <= pi[5] ^ pi[15];
    pi[7] <= pi[6];
    pi[8] <= pi[7] ^ pi[15];
    pi[9] <= pi[8] ^ pi[15];
    pi[10] <= pi[9] ^ pi[15];
    pi[13:11] <= pi[12:10];
    pi[14] <= pi[13] ^ pi[15];
    pi[15] <= pi[14];
    end
assign i_code_out = pi[15];
endmodule
```

This implementation only uses 5 EXOR constructs, whereas a fully modular implementation would need 14 EXOR constructs. This shows how Verilog can be applied to only synthesize needed logic.

Two other items should be pointed out at this time. Maximal length PN generators do not allow all zeros in the shift register, as nothing would be output except a constant low state. A rugged design should include logic to detect the all-zeros condition and insert a "1" if needed.

The second item of note is that maximal length PN generators have almost perfect balance between the numbers of ones and zeros in the full sequence. The number of ones will be one greater than the number of zeros. This produces a small DC offset that may disrupt the operation of a mixer or modulator in a radio. Standard CDMA inserts an extra zero state into the sequence to force DC balance and keep the modulators operating properly.
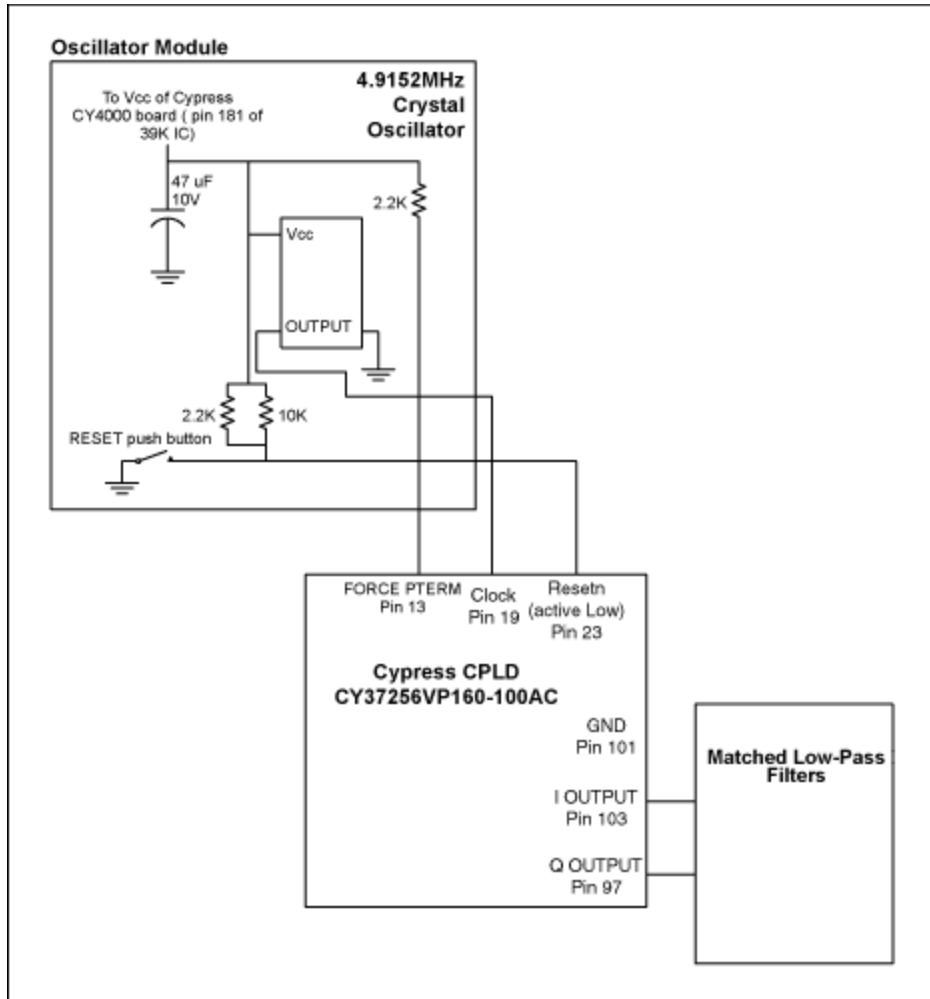
Figure 4. CDMA generator schematic.

Not shown in Figure 4 is the detail of the in-circuit programming support. This small amount of extra circuitry amounts to a 10-pin header that allows a special cable to connect to a personal computer';s parallel printer port. The Windows®-based PC runs special software from Cypress Semiconductor which downloads the bit configuration file into the CY37256. This is a powerful approach to digital system implementation. It allowed rapid identification of errors in the code, and made is simple to re-program the CPLD in circuit and quickly resume testing.

Results

The entire CDMA reverse link generator, including the output low-pass filters, was used to test ACPR on the MAX2361 in a CDMA application. The same test was then carried out using an Agilent E4433B as the baseband I/Q CDMA signal source. The results for ACPR agreed between the two different sources very well; within ±0.5dB. The importance of the output filters on the ACPR result can not be stressed enough, and should be a subject of another application note.

References and Resources

1. R.C. Dixon, *Spread Spectrum Systems*. New York: John Wiley & Sons, 1976
2. David P. Whipple, "North American Cellular CDMA", *Hewlett-Packard Journal*, December 1993, pp. 90-97
3. Ken Coffman, *Real World FPGA Design with Verilog.* Upper Saddle River: Prentice Hall PTR, 1999, ISBN 0-13-099851-6
4. Samir Palnitkar, *Verilog HDL, A Guide to Digital Design and Synthesis* Sunsoft Press/Prentice Hall, 1996, ISBN 0-13-451675-3
5. Special thanks to Lane Hauck at Cypress Semiconductor for his advice and guidance in learning Verilog and the nuances of CPLD design.
6. Special thanks for Dave Devries of Maxim Integrated Products for his collaboration and certain key insights during this project.

Related Parts	
MAX2361	Complete Dual-Band Quadrature Transmitters

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 918: http://www.maximintegrated.com/an918
APPLICATION NOTE 918, AN918, AN 918, APP918, Appnote918, Appnote 918
Copyright © by Maxim Integrated Products
Additional Legal Notices: http://www.maximintegrated.com/legal