Optimize the MAX2309 PLL Divider Values in Dual-Mode Cellular Phones

Describes method to set the PLL divider values in dual mode (analog AMPS and digital TDMA) cellular phones. IS-136 dual mode phone frequency plan is outlined, using a first IF of 119.64MHz, and an AMPS IF of 455kHz. Tradeoffs are needed between PLL phase detector frequency, loop bandwidth and spurious outputs. Brute force solution must switch loop filter components to control stability. A 50Hz error in AMPS mode allows PLL to use same filter components.

Additional Information: Wireless Product Line Page
Quick View Data Sheet for the MAX2309
Applications Technical Support

Introduction

The original cellular telephone system in the USA is based on analog frequency modulation (FM) technology and is called advanced mobile phone system (AMPS). Many telephones still attempt to support the old standard along side the modern digital systems. This is not a tremendous technical challenge when the baseband processor and RF (radio frequency) transceiver are designed for this support. Some modern baseband processors have opted to abandon AMPS support, forcing a telephone handset design engineer to "graft on" the needed circuitry for an analog implementation of AMPS. This application note presents a typical frequency plan for a cellular telephone handset that supports both time division multiple access (TDMA) IS-136 digital mode, and analog AMPS mode. A method is given to select phase lock loop (PLL) divider values to maintain fast response time in the 2nd local oscillator (LO). Using a temperature compensated crystal oscillator (TCXO) of 19.44 MHz, a first IF (intermediate frequency) of 119.64 MHz, and an AMPS IF of 455kHz, it is possible to operate the 2nd LO phase detector at ~40 kHz for both digital and analog modes. A Mathcad worksheet is presented which is easily modified to support other 1st IFs and TCXOs.

Figure 1 shows a partial block diagram for a typical cell phone which supports both IS-136 TDMA and AMPS operation. The front end is the MAX2338 and the required RF filtering. The 1st LO drives the RF mixer, and is controlled by the phase locked loop, referenced to the TCXO. The output of the MAX2338 is the IF of 119.64 MHz, and is conveyed through a differential IF filter to the IF processing circuits. The digital IF is accomplished in the MAX2309 RXIF. This IC

contains the quadrature demodulator which mixes the IF down to baseband I and Q outputs. The MAX2309 also contains the PLL and voltage controlled oscillator (VCO) needed to implement the 2nd LO for the radio. The PLL is programmed through a three-wire serial bus familiar to most engineers. The feedback and reference dividers are actually pairs of registers. This allows the divider values for digital mode operation and analog (FM-AMPS) operation to be loaded once at system initialization, then it is simple to select the correct pair of registers to change PLL modes.

In the system under question, the baseband processor does not have the proper support to allow the AMPS FM signal to be digitally demodulated. A small FM IF section is added, using common 455kHz IF filters which are low-cost and readily available. The needed LO to mix the 1st IF of 119.64MHz down to 455kHz is borrowed from the MAX2309. This is possible because when the phone is in AMPS mode, the I/Q outputs are not needed for digital processing.

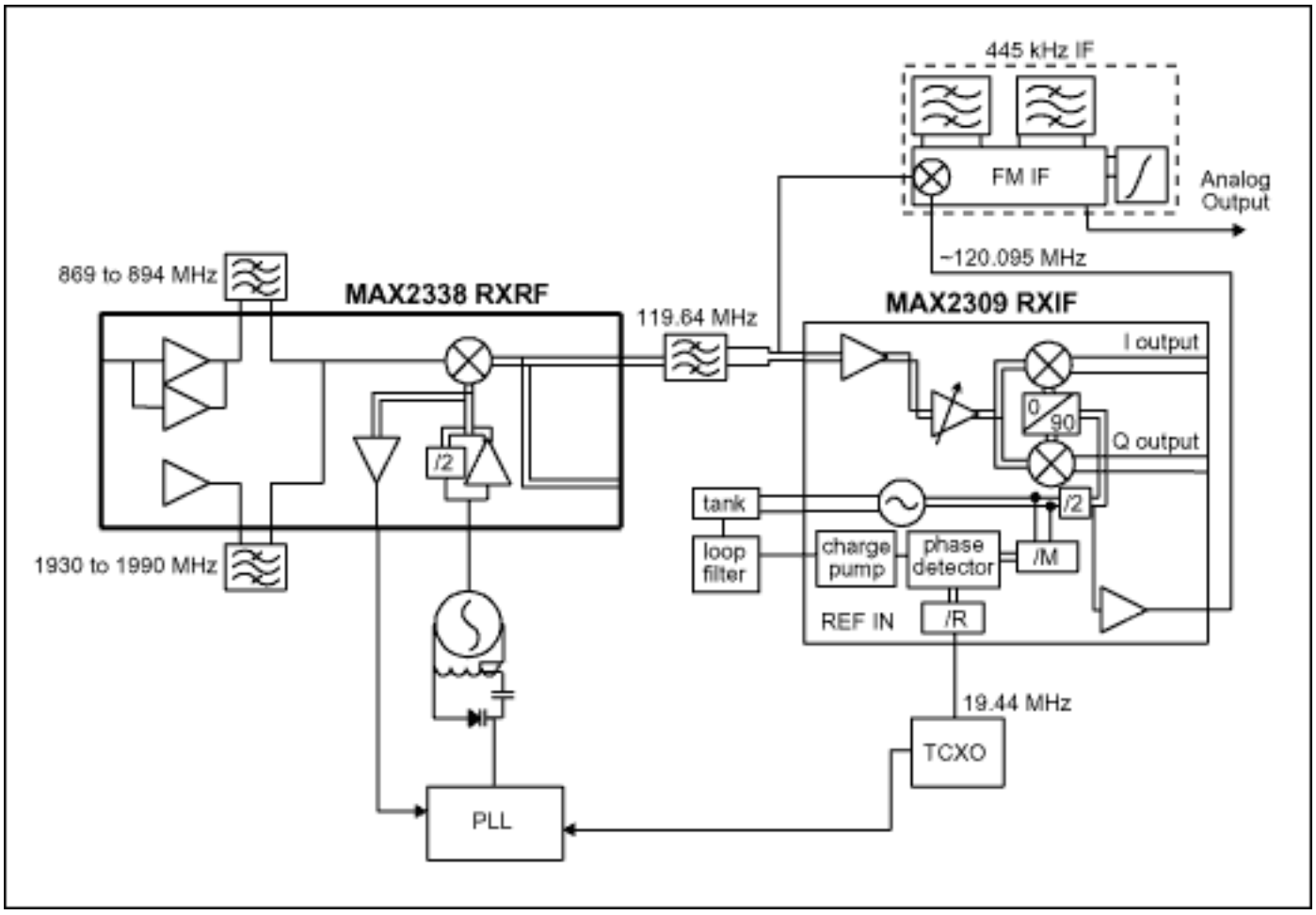


Figure 1. A dual mode receiver for TDMA with analog AMPS support.

The Second LO for Digital Mode

The 2nd LO to allow TDMA digital operation needs to be 119.64MHz. This exactly mixes the IF to baseband to allow digital to analog conversion. The digital signal processor (DSP) then takes

over and completes the needed error detection and correction, etc. The VCO in the MAX2309 that produces the 2nd LO needs to run a twice the required frequency, since there is an internal divide by 2. This divide by two improves performance by:

1. Moving the VCO away from the 1st IF.
2. More accurately obtaining 50% duty cycle.
3. Simplifies the 90 degree circuitry in the quadrature demodulator.

To produce the 2nd LO used for TDMA operation, the internal registers need to be programmed with values that will run the VCO at 239.28MHz. Several solution sets exist for this:

- R = 486, M = 5982 Fref (phase detector operation rate) = 19.44MHz/486 = 40kHz
- R = 243, M = 2991 Fref = 80kHz

Both solutions presented above for the 2nd LO digital mode appear to be acceptable, as the phase detector is at a high enough rate. This allows a given loop filter to reduce the comparison spurious products to an acceptably low level, while still having a wide enough PLL bandwidth for fast response and good noise suppression.

The Second LO for AMPS

The 2nd LO for the AMPS IF can be a source of added complexity if care is not taken. The straight-forward approach to generate the 2nd LO with traditional integer values is as follows:

1. Note the required LO. 120.095MHz in the example given here.
2. Adjust the LO frequency to account for the divide by 2 function. VCO = 240.19MHz
3. Note the position of the last digit in the VCO frequency. The VCO is specified with the last digit in the 10kHz location. This is a candidate for the phase detector comparison rate, Fref = 10kHz. There may be other solutions possible, but usually visual inspection is not sufficient to identify these other rates.
4. Confirm that the timebase, TCXO, is an integer multiple of Fref. 19.44MHz/ 10kHz = 1944
5. Calculate the value for R, the reference divider. 1944 from step #4.
6. Calculate the value for M, the feedback divider. 240.19MHz / 10kHz = 24,019.

It would appear that the design is done, but consider the issue of switching between the AMPS mode and the TDMA mode. The phase detector rate will change from 10kHz to 40kHz. The loop filter will likely need to change time constants when changing mode, and this will add some extra circuitry. Not to mention the lock time and noise character will be different between AMPS and TDMA modes. It would be an improved design if the comparison frequency were close enough between the two modes that the filter components could serve either mode, avoiding the need for switched components. It is a noble cause, but non-trivial most of the time. We need to find other integer values for R and M which will produce the desired output frequency. A few minutes

with a calculator will usually convince an engineer that finding these other integer divide values might consume many hours!

The Computer to the Rescue!

The hardware to generate the 2nd LO is configured to support integer values only. (Until we have fractional PLLs, we must find another way to solve this problem.) That implies that the frequencies we can generate are "quantized", and a small change in the divider setting may jump the output frequency over the desired value. A computer can easily search for the desired register settings, but there is no guarantee it will find a solution. We must bend the rules at this point in order to make progress.

Radio engineers have traditionally sought to hit LO frequencies exactly. In an AMPs receiver, do we really need to exactly place the 2nd LO on frequency? With a little thought one can arrive at the conclusion that a small amount of error in the 2nd LO should be tolerated gracefully, due to the operation of the discriminator and the bandwidth of the 2nd IF filters. The 2nd LO has to be close enough to exact to get the signal through the IF filters and into the demodulator. The fact that the frequency is shifted up or down a small amount should pose no serious problems. That is what we shall do to assist in finding other solutions to the 2nd LO PLL settings. The Mathcad worksheet included in the Appendix was used to search for solutions to this problem, using the constraint that the 2nd LO can tolerate 25Hz of error.

Results

The first portion of the Mathcad worksheet (Appendix) computes the number of iterations needed to do a brute force, exhaustive search over all R and M values. 45.79 million attempts will be needed. I attempted to do this using Microsoft's Excel spreadsheet, and it couldn't even handle a sheet large enough to test the idea. The problem was then scaled back to a subset, and the spreadsheet still was working after 10 minutes!

There is an old adage that the best optimizer is the "one between your ears"[1]. It is clear that one needn't exhaustively search. For each value of R, there is only one (possibly two?) M value that makes sense to test. The Mathcad worksheet follows this method. The reference divider values, R, are allowed to range from a minimum value to a maximum value. For each R the appropriate M value is calculated and converted to an integer. The output of the PLL is calculated using R and M. This output is compared to the target value, 240.19MHz. An error is computed and compared to the allowed error budget of 50Hz. A vector of R-values is built composed of only the R-values that meet the error budget. This vector is graphed to allow a visual inspection of the results. The exact R-values were then found by rapid trial and error, using the graph as a guide for where to search for the correct R-value. Six R-values were quickly found that work. (We only are interested in ones that produce phase detector rates near 40kHz or 80kHz.)

For R = 422, M = 5214 the output frequency is: 19.44MHz ▪ (5214/422) = 240.189952607 Error

~ 47Hz

The phase detector will operate at ~ 46kHz, which is close enough to the 40kHz rate when in digital mode to allow the same loop filter to be used for both modes.

Appendix

PLL divider value algorithm to find feedback and reference divider in a frequency synthesizer such that the highest phase dectector reference frequency is used, giving the widest possible loop bandwidth.

Define values to be used in this effort. Assume 1st IF = 119.64 MHz, use high side injection at 2nd LO.

Ftcxo:= $19.44 \cdot 10^6$;The PLL reference oscillator
Ftarget:= $240.19 \cdot 10^6$;The desired or target output frequency (high-side injection @2nd mixer)

Estimate how many iterations would be needed with a brute force search...

The highest value for R will set the PLL phase detector reference to 10kHz, allowing simple integer value for M to exactly set the output frequency to the desired value. The lowest value of R to be considered is one that would run the phase detector at the maximum allowed rate. For this effort, assume the detector can run ~ 1MHz maximum.

StepSize:= $10 \cdot 10^3$

Rmin:= Ftcxo / $1 \cdot 10^6$

Rmin = 19.44 ;This must be an integer, since the actual hardware will be simple logic. Convert to integer.

Rmin:= Rmin - mod(Rmin, 1) ;Convert the real value to integer

Rmin = 19

Rmax:= Ftcxo / StepSize ;Estimate the largest value to be considered for the reference divide value

Rmax:= Rmax - mod(Rmax, 1)

Rmax = 1944

Estimate the Range of feedback divider values to be considered

Mmin:= Ftarget • Rmin / Ftcxo

Mmin:= Mmin - mod(Mmin, 1)

Mmin = 234

Mmax:= Ftarget • Rmax / Ftcxo

Mmax:= Mmax - mod(Mmax, 1)

Mmax = 24019

For each value of R, try the whole range of M values. Estimate the number of attempts this method entails...

MaxAttempts:= (Rmax - Rmin) • (Mmax - Mmin)

MaxAttempts = $4.579 \cdot 10^7$

Clearly, a more efficient method must be found. Excel running a on PC with large memory resoureces will not be able to handle this exhaustive search. After some thought, it is understood that only one or two values of M need to be tried for each value of R. This greatly reduces the amount of computation to locate other solutions.

For a given R value, the required M can be calculated exactly. The M value will usually b non-integer. This method attempts to locate integer values of M that will produce the desired output frequency with a specified error. It is straight forward to compute each M . bound can be placed on the output frequency based on the R value, the TCXO frequency and the allowed error component.

ε:= 50 ;Allowed frequency error

R:= Rmin, Rmin +1...Rmax ;establish a range of R values to search

Estimate the feedback divider value to test.

Me (R) = Ftarget • R / Ftcxo

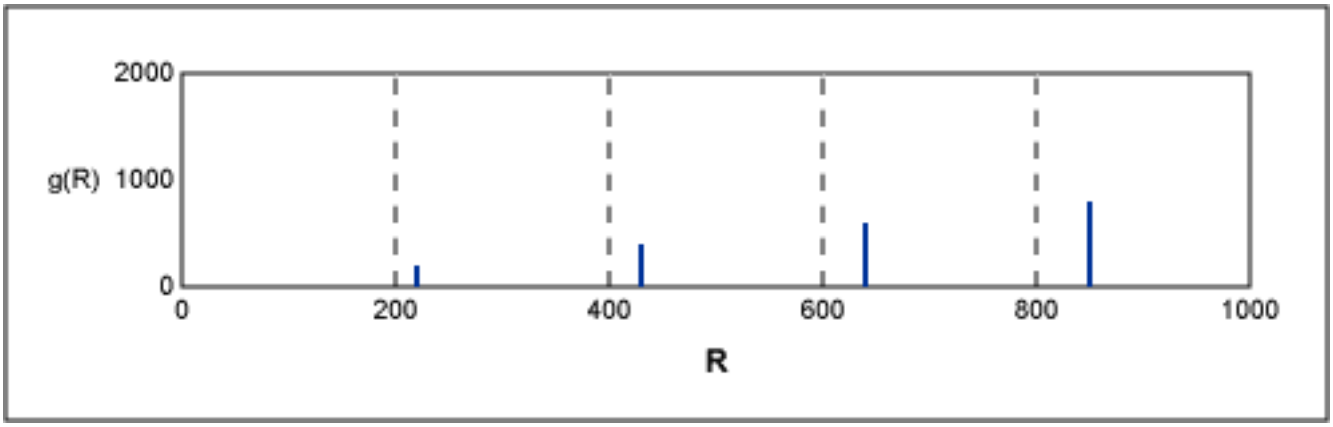
Calculate the integer portion of Me

intMe (R):= Me (R) - mod(Me(R), 1)

We need to test each output frequency for existence within the error bounds.

Fout(R):= intMe(R) • Ftcxo / R

g(R):= if [$(fout(R) - ftarget)^2 < \varepsilon^2$, R, 0]



Locate the exact R values that produce acceptable error values.Use the graph to guide the guesses.

Phase Detector Comparison	Frequency
g(211) = 211	Ftcxo / 211 = 92132.701
g(422) = 422	Ftcxo / 422 = 46066.351
g(633) = 633	Ftcxo / 633 = 30710.9
g(844) = 844	Ftcxo / 844 = 23033.175
g(1055) = 1055	Ftcxo / 1055 = 18426.54
g(126) = 1266	Ftcxo / 1266 = 15355.45

References

1 Coffman, Ken *Real World FPGA Design with Verilog*, Prentice Hall PTR, Upper Saddle River, NJ, 2000, pg. 71.

June 2002

More Information

MAX2309: QuickView -- Full (PDF) Data Sheet -- Free Samples

MAX2338: QuickView -- Full (PDF) Data Sheet -- Free Samples