

# Reconfigurable Firmware-Defined Radios Synthesized from Standard Digital Logic Cells

Muhammad Faisal, Youngmin Park, David D. Wentzloff  
Dept. of Electrical Engineering and Computer Science  
University of Michigan, Ann Arbor, MI, USA

## ABSTRACT

This paper presents recent work on reconfigurable all-digital radio architectures. We leverage the flexibility and scalability of synthesized digital cells to construct reconfigurable radio architectures that consume significantly less power than a software defined radio implementing similar architectures. We present two prototypes of such architectures that can receive and demodulate FM and FRS band signals. Moreover, a radio architecture based on a reconfigurable all-digital phase-locked loop for coherent demodulation is presented.

**Keywords:** Synthesizable, Reconfigurable Radios, All-digital Receivers, Firmware Defined Radios

## 1. INTRODUCTION

The idea of having a reconfigurable radio that can host a variety of communication standards in a single platform dates back to 1980s; however, radio reconfigurability was first officially introduced in the context of software-defined radios in 1995 when the term software radio (SR) was coined [1]. The idea of software radios was to use the same hardware to transmit and receive different communication standards, and the radios can be reconfigured through reprogramming a digital signal processor. This provides the most flexibility in reconfiguring the radio; however, it typically comes at the expense of high power consumption and degraded communication performance. Recently, various commercial software-defined radios have appeared [2]-[4], some of which the front-end analog hardware alone consumes over 10W and they further require an interface to a PC to implement the DSP.

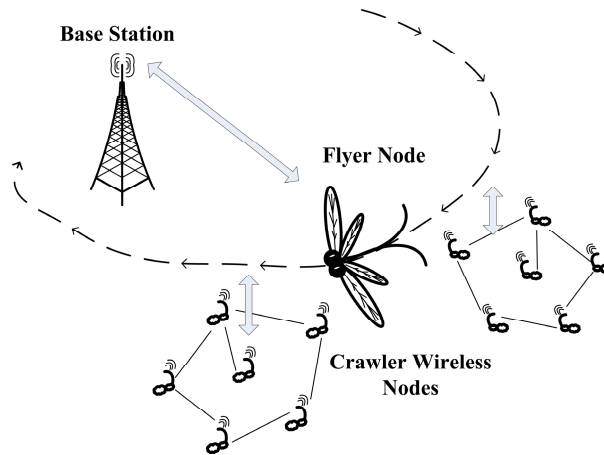


Figure 1: Wireless communication application scenario targeted by the MAST project.

One application scenario for reconfigurable radios is a battle field where situational awareness in harsh environments is crucial. In this application, a wireless node is required to maintain communication in the presence of large RF interference and through complex channels such as collapsed buildings or out of a cave. Figure 1 illustrates one such scenario being targeted by the Micro Autonomous Systems and Technology (MAST) project [5]. In this application, battery-operated autonomous flyers travel over collections of crawling nodes forming local networks, and it

is the flyer's objective to upload information from the local networks and reliably relay that information over long ranges to a base station. A few design criteria that can be inferred from the MAST application are: low power consumption to reduce the size and weight of the batteries carried by the autonomous nodes, reconfigurability of the radios to avoid interfering RF signals and possibly to avoid detection, and finally the ability to change frequencies and modulation formats for robustness against harsh wireless channels.

### 1.1 Software-Defined Radio

Figure 2(a) illustrates the architecture of a traditional software-defined radio (SDR) receiver. Typically, a SDR consists of an RF front-end that can be tuned to receive multiple frequency bands. The front-end is immediately followed by a high performance analog-to-digital converter (ADC) that digitizes the received signal. The digitized received signal is then demodulated through digital signal processing. Often the DSP is an instruction-based processor that can perform very complicated computations for DSP algorithms, at the expense of high power consumption due to the typical sample rate of the ADC. This architecture is extremely flexible, and can be reconfigured through software programming of the DSP and configuration of the RF front-end. Current commercial SDRs occupy a large area and their total power consumption is on the order of  $\sim 100\text{W}$ . This hinders their deployment in applications that also require portability.

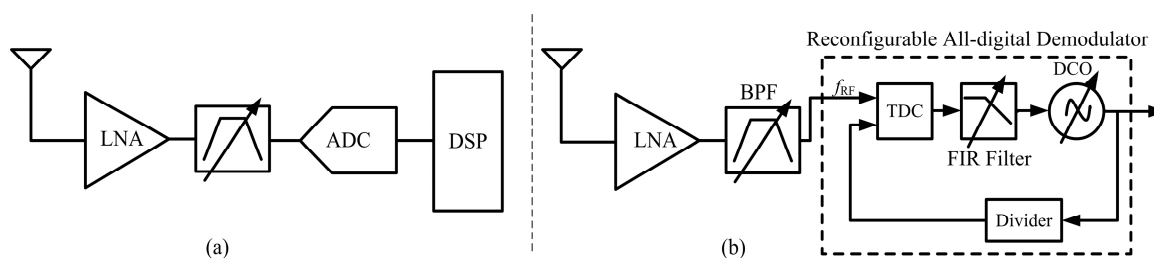


Figure 2: (a) Traditional software-defined radio architecture, and (b) an example firmware-defined radio architecture that directly inputs a phase-modulated RF signal to an all-digital demodulator without an ADC.

### 1.2 Firmware-Defined Radios

Recent advances in CMOS processes have made it practical to realize circuits with high timing precision, which makes all-digital implementations of traditionally analog functions feasible. The main contributions of this paper are the implementation and demonstration of firmware-defined radio (FDR) receiver architectures as shown in Figure 2(b). By inspecting the software-defined radio block diagram, we can conclude that large amounts of power and area are consumed by the ADC and DSP blocks. An alternative implementation of a demodulator that does not require explicit high-resolution analog-to-digital conversion of the RF signal could significantly reduce the power and area requirements of a reconfigurable radio. Assuming we have a very selective reconfigurable RF front-end, we can demodulate the received signal using a reconfigurable all-digital phase-locked loop (ADPLL) that can work with various modulation formats as well as frequency bands. An all-digital PLL (ADPLL) can be used to demodulate PSK and FSK modulated signals. A *synthesizable* implementation of the ADPLL makes it possible to reconfigure and optimize it for different frequency bands and modulation formats, e.g. by reconfiguring an FPGA or similar device implementing the ADPLL and FDR demodulator. In a 65nm CMOS process, a frequency band up to a center frequency of 6GHz can be practically targeted, without the need for a Nyquist-sampled ADC. The main advantage of the FDR is moderate flexibility at a fraction of the area and power of SDRs. The drawback is an FDR does not have the same capability or flexibility of the SDR.

## 2. FIRMWARE-DEFINED RECEIVER ARCHITECTURES

FDR receivers primarily take advantage of the precise timing capabilities of deep-submicron CMOS processes. For example, single gate inverter delays may be on the order of picoseconds in a 32nm CMOS process. FDR receivers derive their strength from this underlying advantage of fast CMOS circuits; therefore, the modulation formats that they are capable of supporting must play to this strength. For this reason, FDRs are amenable to phase and frequency modulation formats. With these modulation formats, information is encoded in the phase (or timing) of the wireless signal, while the amplitude remains constant. For example, QPSK, higher-order PSK, and FM, are commonly used constant-amplitude modulation that meets the requirements for FDR implementation. Amplitude information, on the other hand, is much

more difficult to extract using only logic gates, because logic signals can only take on one of two amplitudes:  $b=\{0,1\}$ . Therefore, amplitude modulated signals are not compatible with FDRs, such as higher order QAM and OFDM because it produces signals that do not have constant amplitude. While the later amplitude modulation formats are popular in high-performance wireless communication links, the limitations of using only timing-based modulation in FDRs may be outweighed by the advantages of reconfigurability and low power operation. In this section, we outline all-digital receiver architectures for performing timing-based demodulation that cover a range of implementation complexity and performance. We begin with the most basic, counter-based architecture, then introduce a digitally-controlled, tunable delay element, and finally present an ADPLL-based architecture capable of resolving fine timing differences.

## 2.1 Counter-Based Frequency-to-Digital Converter

The first architecture is a frequency-to-digital converter (FDC) that directly measures and digitizes the frequency of an input signal. This architecture can be used in an FM demodulator by sampling the frequency of the FM signal at a designed symbol rate, and then de-mapping each symbol frequency to the corresponding received data bit. A block diagram of the FDC is shown in Figure 3. Also included in Figure 3 is an illustration of the de-mapping of binary FSK modulated data to the corresponding received bits (also known as bit slicing). All functional blocks beyond the limiter are standard digital logic cells, and therefore this entire FDC can be implemented in a reconfigurable digital device such as an FPGA.

The FDC uses a coarse/fine procedure to measure the frequency as follows. The incoming FM signal is passed through a limiter to convert the signal to a digital clock with amplitude of  $\{0,1\}$ . The cycles of the FM signal are then counted over the duration of one symbol period, where a symbol period (close to the data rate) is much longer than the period of the FM signal (close to RF frequency). This count of FM cycles over a fixed period of time provides a coarse measurement of the FM frequency. However, there may be some fine residual time between the last rising edge of the FM signal and the rising edge of the symbol clock. This residual time is quantized with an accuracy of a high-frequency fine clock by counting the number of fine clock cycles over the duration of the residual time. The fine clock frequency should be much greater than the FM frequency, and it ultimately defines the timing resolution of the FDC. For an FM modulated signal, we can calculate the effective number of bits (ENOB) of resolution of the coarse frequency measurement as  $ENOB = \log_2(2f_{dev}/f_{symbol})$  where  $f_{dev}$  is the frequency deviation of the FM signal and  $f_{symbol}$  is the symbol sample rate. The fine residual time measurement further increases the resolution by an additional  $\log_2(f_{fine}/f_{FM})$  where  $f_{fine}$  is the high-frequency clock and  $f_{FM}$  is the average frequency of the FM signal.

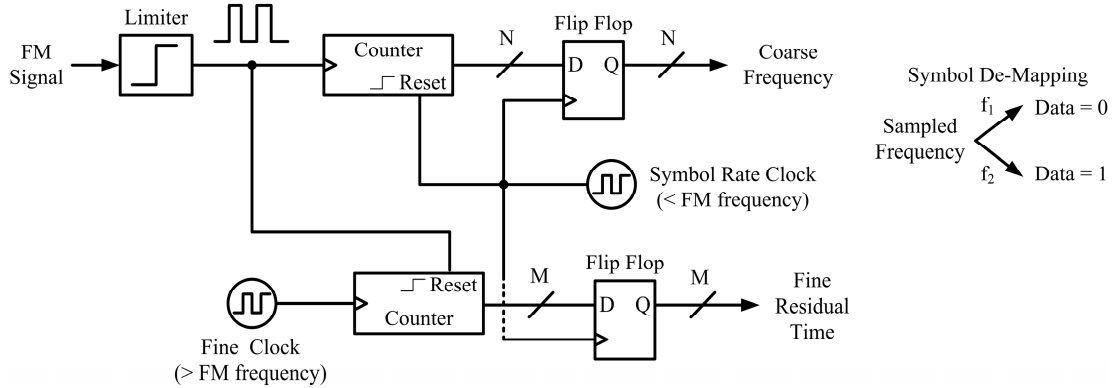


Figure 3: A counter-based frequency-to-digital converter (FDC) that measures a coarse frequency by counting FM cycles over a fixed time duration equal to the symbol period, and a fine residual time with resolution equal to the fine-clock period.

## 2.2 Digitally-Controlled Delay Element

The counter-based FDC has the advantage of using very few gates and an all-digital implementation; however, a major disadvantage of this architecture is that the fine resolution is determined by the maximum speed of a high-frequency clock. For each CMOS process, there is a practical limit on how fast standard cell counters may be clocked, which will limit this resolution. Furthermore, as the frequency of this oscillator increases, so does the power consumption of the circuit. In the following section we will describe an architecture that can provide much finer time

resolution than the maximum clock frequency, and do so without the added power consumption. But first, in this section we will describe the basic digitally-controlled delay element which will form the building block for the following and future firmware-defined radio architectures.

A schematic of the digitally-controlled delay element is shown in Figure 4, along with a three-stage digitally-controlled ring oscillator using the delay element as the basic building block [6]. The delay element consists of a number  $N$  tri-state buffers all connected in parallel as shown in the figure. Tri-state buffers are components that are available in all standard digital cell libraries, which can output one of three states: 1) drive high, 2) drive low, 3) drive nothing (high-impedance output). When all tri-state buffers in a delay cell are turned on, a signal incident on the input of the delay cell will propagate through the cell with the shortest possible delay. As individual tri-state buffers are disabled, the propagation delay through the cell will increase with a resolution that is much finer than a gate delay (1ps is typical for a 65nm CMOS process, and 32 tri-state buffers per delay cell). These digitally controlled delay cells can be used to create delay lines and ring oscillators, which can further be assembled into DLLs, PLLs, and demodulators. By using only digital gates available in standard cell libraries, the entire designs may be synthesized and implemented in a digital device, such as an FPGA or a partially reprogrammable ASIC. The main advantages of this delay cell are its compact size, ability to be synthesized, and its fine timing resolution. One disadvantage is the calibration required to measure the relative impact of each tri-state buffer in the cell. Due to mismatch in the wiring and layout of the delay cells, the impact of each buffer is unique. However, we have developed an all-digital calibration technique for these delay cells, which is explained in detail in [6].

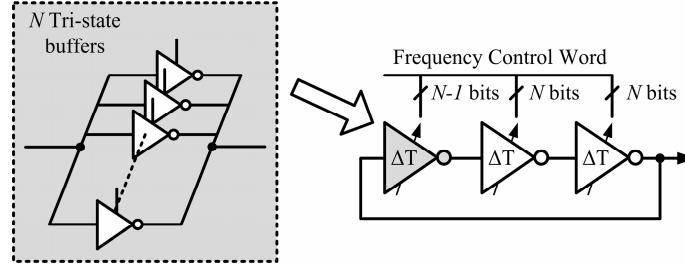


Figure 4: A tunable delay cell formed out of parallel connected tri-state buffers and a three-stage digitally-controlled ring oscillator using copies of the delay cell [6].

### 2.3 All-Digital Time-to-Digital Converter and PLL

Finer accuracy timing measurements are required in order to achieve a higher performance FDR receiver. In this section, we describe an architecture for an all-digital time-to-digital converter (TDC) implemented in a 65nm CMOS process that can achieve 1ps resolution. A block diagram of the TDC is shown in Figure 5. The goal of the TDC is to measure the time difference between the rising edges of the *Start* and *Stop* signals. When the *Start* is asserted, the slow DCO begins to oscillate with a period of  $T_s$ , and the number of oscillations is counted by the coarse counter. After an input delay of  $T_{input}$ , the *Stop* signal is asserted which triggers the faster DCO to oscillate with a period of  $T_f$ . At this time, the coarse counter is disabled, and the output of the counter represents a coarse measurement of the time between *Start* and *Stop* rising-edges ( $T_{coarse}$ ). To improve the measurement accuracy, the residue of the input delay ( $T_{fine}$ ) is measured by the Vernier structure. Because  $T_f$  is slightly smaller than  $T_s$ , the time difference between rising edges of the two oscillations is reduced every cycle by the difference in periods ( $T_s - T_f$ ), and the edge of the fast DCO eventually catches up with the slow DCO. By counting the number of cycles it takes for the fast DCO to catch up with the slow DCO,  $T_{fine}$  is measured. Then, the overall measurement of  $T_{input}$  can be determined as  $T_{input} = T_{coarse} + T_{fine} = N_s \times T_s + N_f \times (T_s - T_f)$  where  $N_s$  and  $N_f$  are the number of cycles of the slow and fast oscillations, respectively, and  $(T_s - T_f)$  is programmed to be much smaller than  $T_s$ . As shown in Figure 5, the TDC operates in two-steps; a coarse step and a fine step. The coarse step resolution is the period of the slow DCO, and the fine step resolution is the difference between the periods of the two DCOs. Note that the fine resolution does not depend on the absolute frequencies of the DCOs, but only their *difference* in periods.

Unlike conventional Vernier delay lines, the cyclic Vernier TDC adopts the ring structure to extend input range, limited only by the counter size that easily scales according to application. Also, the linearity of the TDC improves by

utilizing the periods of the DCOs, which is repetitive and consistent over operation, while the delay in Vernier delay lines is more susceptible to variation and mismatch. The drawback of the cyclic Vernier TDC is a large latency of the fine step measurement. It takes one period of the fast DCO cycle ( $T_f$ ) to resolve a time difference of one fine step resolution ( $T_s - T_f$ ). To reduce the conversion time, we adopt the two-step operation. The coarse step covers a large input time difference without any latency, and the residue of the coarse step is measured by the fine step, which is less than one cycle of slow DCO.

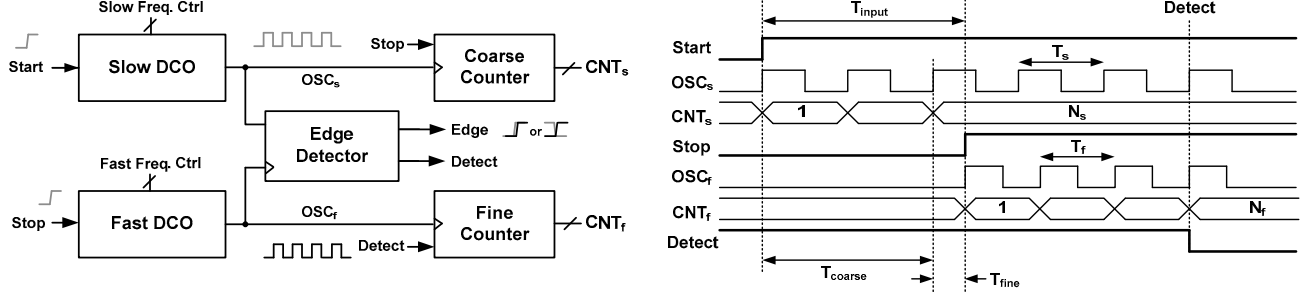


Figure 5: Block diagram of the time-to-digital converter and the corresponding timing diagram.

Figure 6 shows the die photo of the fabricated TDC, and measurement results with the two-step operation. The slow DCO is tuned to have a period of 220ps by turning off the appropriate tri-state buffers. Then, the period of the fast DCO is calibrated to have a slightly lower period than the slow DCO. Figure 6 shows two different fine step resolutions, 1ps and 5ps. According to the target performance, the coarse and fine step resolutions are digitally reconfigured, providing flexibility in the TDC. The power dissipation in the Vernier structure depends on  $T_{input}$ , and the sampling frequency of the TDC. The DCOs, which are the most power hungry blocks in the TDC, oscillate only when  $T_{input}$  is measured. Therefore, the TDC operation is duty-cycled, and power dissipation is proportional to the sampling frequency. The measured power consumption during coarse and fine step measurements is as follows.  $P_{coarse} = 1.4mW \times T_{coarse} \times F_s$  and  $P_{fine} = 2.6mW \times T_s / (T_s - T_f) \times T_{fine} \times F_s$  where  $P_{coarse}$  and  $P_{fine}$  are the powers during coarse and fine measurement, and  $F_s$  is the sample frequency.

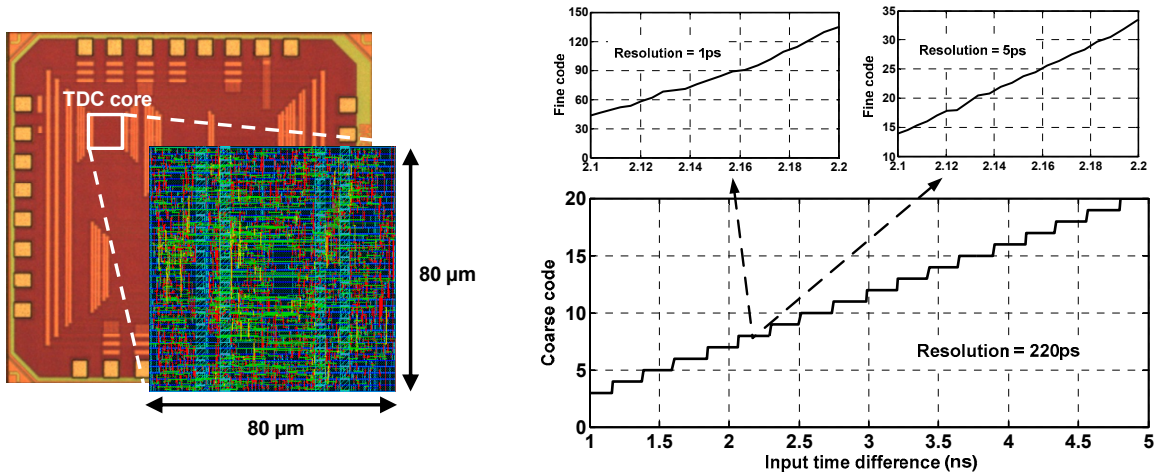


Figure 6: A die photograph of the fabricated all-digital TDC and the measured coarse and fine results [7].

An all-digital phase locked loop replaces conventional analog blocks in a PLL with an all-digital architecture. Figure 7 shows the general block diagram of an ADPLL. In the ADPLL, a TDC is used to compare the phase error between the reference clock ( $F_{ref}$ ) and the divided clock ( $F_{div}$ ) from the digitally controlled oscillator (DCO). Then, the digitized phase error is filtered by a digital loop filter (DLF), and utilized to control the frequency of the DCO. The TDC and DLF replace a charge pump and large passive components of conventional analog PLLs, reducing the power dissipation and area of the ADPLL. Also, the digital interface between the blocks enhances testability and

programmability of the ADPLL. In this architecture, the DLF and the divider are required to satisfy only timing constraints; thus, they can be implemented with digital logic circuits, following a digital design flow.

Because a PLL can accurately track the phase and frequency of an input signal, it has several uses in both FM and PM signal (constant-amplitude) demodulators. Furthermore, because the entire ADPLL can be implemented using digital logic gates, it can be synthesized onto an FPGA or partially programmable ASIC. This would allow the hardware to be reprogrammed to implement ADPLLs that are optimally designed for a target frequency band and modulation format. Additionally, because ADPLLs can track signals up to 6GHz in a 65nm CMOS process, it is practical to implement the demodulator directly at the RF frequency provided adequate channel-select filtering can be accomplished prior to the demodulator. The delay cell, TDC, and ADPLL comprise the base blocks with which more advanced FDR receivers can be constructed.

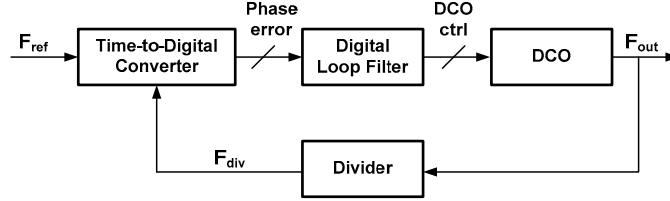


Figure 7: All-digital PLL using a time-to-digital converter and digitally controlled oscillator (DCO).

### 3. FM RADIO AND FRS RADIO PROTOTYPE

Figure 8 shows the block diagram of a prototype FDR capable of demodulating FM radio and FRS radio by reconfiguring the RF front-end and demodulator in the FPGA. The RF front-end was developed with commercial mini-circuits components. For the FM radio, a local Ann Arbor radio station at 107.1MHz is selected using a bandpass filter, and mixed down to an intermediate frequency of 10.7MHz in order to keep the signal at a reasonable frequency for processing in the FPGA. The down converted FM signal is further filtered with a high-Q bandpass filter, and then converted to a digital clock using a hysteretic comparator as a limiter. The output of the comparator is then sent to an FPGA where a counter-based demodulator demodulates the FM signal to extract the transmitted audio information. For debugging and testing purposes, the sampled output of the demodulator was then converted to an analog audio signal and played into a PC speaker. We were able to successfully demodulate the 107.1MHz broadcast using this architecture.

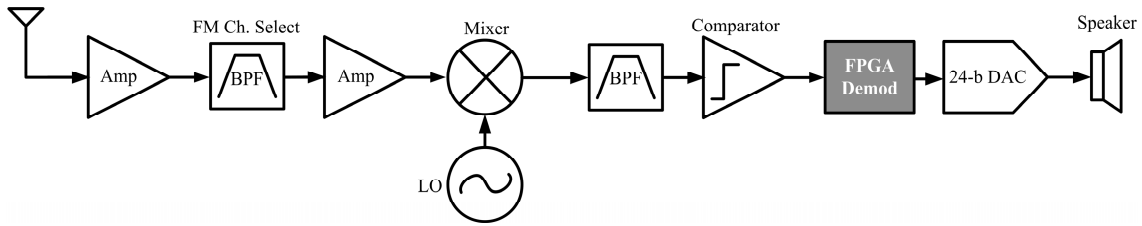


Figure 8: FM radio and FRS radio prototype receiver block diagram.

The block diagram of the demodulator implemented on the FPGA is shown in Figure 9, which is similar to the coarse/fine counter based FDC described in Section 2.1. Time resolution of the FM demodulator is limited by the FPGA fine clock, which is set to 100MHz for the prototype. The received FM signal is first synchronized to the system clock. The synchronized FM clock cycles are then counted by a coarse counter. The system clock is divided down to generate an audio sampling clock of 8kHz. The values of the free running fine counter are latched into separate registers when either a rising edge of the audio sampling clock or the rising edge of FM input arrive. The latched values are then subtracted in order to measure the residual time period remaining after the coarse frequency measurement. The residual count is then combined with the coarse count as a floating point number to make a full DAC word.

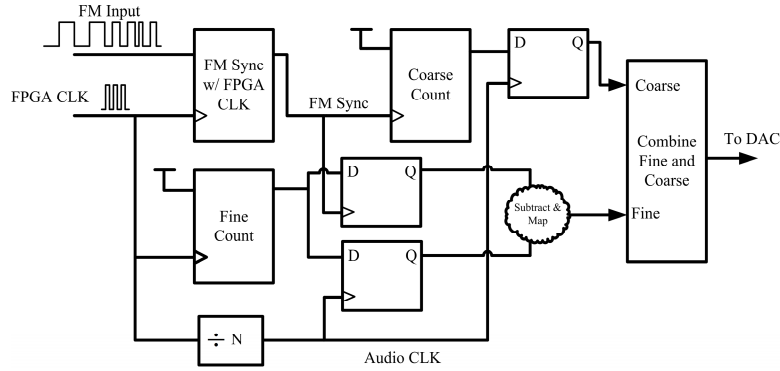


Figure 9: FM demodulator implemented on the FPGA.

To demonstrate the reconfigurability of the FDR, we also implemented a family radio service (FRS) receiver by reprogramming the demodulator and modifying the RF front-end. FRS radios operate in the 462-467MHz band. The block diagram of the entire receiver is similar to the FM radio (Figure 8). The differences between the FM and FRS radio signals are that the center frequencies are different, and that the FRS channel is 25kHz wide as compared to 50kHz for FM radio. Both radios use FM modulation, therefore similar demodulators are used with some modifications to optimize the performance. As a proof of concept, the front-end was only configured to receive one channel of FM (107.1MHz) and FRS (462.6375MHz) radios. The respective LO frequencies for down-conversion were tuned to 96.4MHz and 473.33MHz. Moreover, while the audio sample rate for the FM radio was 8kHz, 3kHz was used for the FRS radio in order to maintain moderate resolution with the smaller bandwidth of FRS radio signals. By modifying the RF front-end to select the FRS channel, and reprogramming the FPGA, FRS audio was then successfully received and played through the DAC output to the speakers. A picture of the prototype is shown in Figure 10.

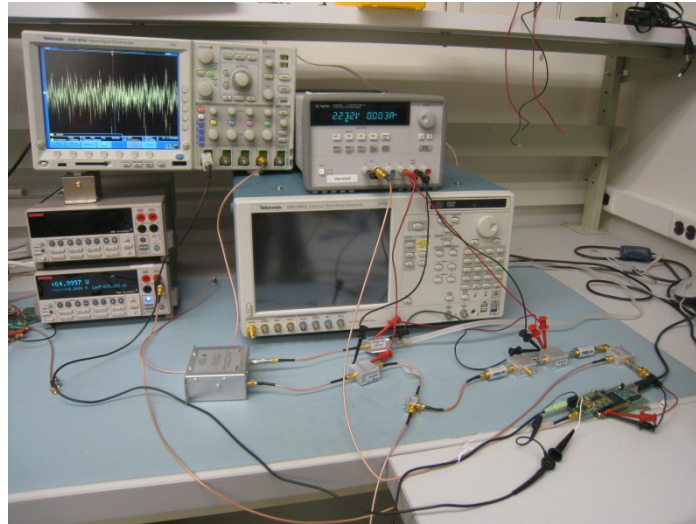


Figure 10: A counter-based frequency-to-digital converter (FDC) that measures a coarse frequency by counting FM cycles over a fixed time duration equal to the symbol period, and a fine residual time with resolution equal to the fine-clock period.

While using an FPGA provided a flexible digital platform to demonstrate the FDR demodulators, it imposed operating frequency limits which required the down-conversion of the RF signals to an intermediate frequency of 10.7MHz. By using a TDC and ADPLL architecture in a custom, partially programmable ASIC, similar functionality could be accomplished directly at the RF frequencies without down-conversion.

#### 4. CONCLUSIONS

There are many applications that demand reconfigurability in a wireless communication link. Software-defined radios provide the most flexibility; however, with the highest cost in power consumption. For battery-operated applications, the typical power consumed by SDRs is not practical. In this paper we describe the firmware-defined radio as a practical alternative to SDRs. FDRs can achieve moderate flexibility due to their all-digital implementation, and reduce power consumption by eliminating high-speed ADCs and complex DSP, provided there is sufficient filtering of the received signal in the RF front-end. Given the recent advancements in switchable and tunable MEMS filters, and the timing performance of deep-submicron CMOS technology, all-digital, reconfigurable FDRs are an attractive alternative to more traditional SDRs and multi-mode transceivers commercially available today.

#### REFERENCES

- [1] Mitola, J., "The software radio architecture," IEEE Communications Magazine, 33(5), 26-38 (1995).
- [2] "Wideband Receivers – IC-R9500-02", Universal Radio Inc  
<http://www.universal-radio.com/catalog/widerxvr/0095.html> Retrieved: 3/24/2011
- [3] "WR-G315e", WinRadio  
<http://www.winradio.com/home/g315e.htm> Retrieved: 3/15/2011
- [4] "The FlexRadio FLEX-5000A™ Software Defined Radio", Flex Radio Systems,  
[http://www.flex-radio.com/Products.aspx?topic=F5Ka\\_details](http://www.flex-radio.com/Products.aspx?topic=F5Ka_details) Retrieved 3/15/2011
- [5] "Micro Autonomous Systems and Technology (MAST)", Army Research Labs,  
<http://www.arl.army.mil/www/default.cfm?page=332>, Retrieved 3/15/2011
- [6] Park, Y., Wentzloff, D. D., "An All-Digital 8pJ/pulse IR-UWB Transmitter Synthesized from a Standard Cell Library," IEEE Journal on Solid-State Circuits, (2011).
- [7] Park, Y., Wentzloff, D. D., "A Cyclic Vernier TDC Synthesized from a Standard Cell Library in 65nm CMOS," IEEE Transaction on Circuits and Systems, (2011).