A Baseband Processor for Impulse Ultra-Wideband Communications

Raul Blazquez, *Member, IEEE*, Puneet P. Newaskar, *Member, IEEE*, Fred S. Lee, *Member, IEEE*, and Anantha P. Chandrakasan, *Fellow, IEEE*

***Abstract*—This paper presents a baseband processor architecture for pulsed ultra-wideband signals. It consists of an analog-to-digital converter (ADC), a clock generation system, and a digital back-end. The clock generation system provides different phases of a 300-MHz clock using four differential inverter stages. The specification of the jitter standard deviation is 100 ps. The Flash interleaved ADC provides four bit samples at 1.2 Gsps. The back-end uses parallelization to process these samples and to reduce the signal acquisition time to 65 μs. The entire synchronization algorithm is implemented in the digital domain, without feeding any signals back to the clock control. The baseband processor and ADC were implemented on the same 0.18-μm CMOS die at 1.8 V as part of a complete baseband transceiver. A wireless data rate of 193 kb/s is demonstrated.**

***Index Terms*—Analog-to-digital conversion, radio receivers, synchronization, transceivers, ultra-wideband technology.**

I. Introduction

ULTRA-WIDEBAND (UWB) radio communication, though a widely used technology in the millimeter-wave radar community, is recently being re-visited by the integrated circuits community as a viable high-speed, last-meter wireless link technology. In 2002, the Federal Communication Commission (FCC) authorized the use of UWB signals for communication purposes [1] in the band from 3.1 to 10.6 GHz with the following constraints: a spectral mask that sets a maximum average equivalent radiated isotropic power spectral density of -41.3 dBm/MHz and a minimum bandwidth of 500 MHz. IEEE working group 802.15.3a is addressing a standard for Wireless Personal Area Network Communications (WPAN) with high data rates at short distances using UWB signals.

UWB signals promise large data-rates, low probability of interception, and the capability of estimating distances between the transceivers with a precision as good as a few centimeters, depending on their bandwidth. UWB transceivers are dominated by a digital baseband that would perform most of its required functionality.

Manuscript received December 15, 2004; revised March 1, 2005. This work was sponsored by Hewlett-Packard under the HP/MIT Alliance and by the National Science Foundation (NSF) under Grant ANI-0335256. Any views expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

R. Blazquez, F. S. Lee, and A. P. Chandrakasan are with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: rbf@mit.edu; fslee@mit.edu; anantha@mtl.mit.edu).

P. P. Newaskar is with Silicon Laboratories Inc., Austin, TX 78735 USA (e-mail: pnewaskar@alum.mit.edu).

Digital Object Identifier 10.1109/JSSC.2005.852157

In this paper, the architecture, implementation, and measurements of a baseband processor for pulsed ultra-wideband signals are presented. Although originally designed and tested for baseband pulses over a wireless link, this architecture may be applied to an FCC-compliant transceiver by adding functionality to the RF front-end for up/down-conversion within the 3.1–10.6-GHz band. This architecture was implemented using a TSMC 0.18-μm CMOS process at 1.8 V.

II. UWB Signals

A pulsed ultra-wideband signal uses a sequence of very short pulses in which the information is encoded either in the sign, resulting in binary phase shift keying (BPSK) or pulse amplitude modulation (PAM) [2], or the delay, resulting in pulse position modulation (PPM) [3]. BPSK was chosen for the transceiver implemented here because, for binary signals, it has a 3-dB signal-to-noise ratio (SNR) advantage over PPM (considered as an orthogonal modulation [4]). This work focuses on a receiver for pulsed UWB signals, using 0–300-MHz baseband pulses. This architecture is easily scaled to larger bandwidths. In this implementation, each bit of information is represented with a sequence of 31 pulses with a width equal to $T_p = 3.3$ ns, and every two consecutive pulses are separated by $T_f = N_f \cdot T_p$ with $N_f = 50$, resulting in a very low duty cycle ($\sim$2%) [5] and a bit duration of $D_{\text{bit}} = 1550 \cdot T_p$. The information is encoded on the sign of the pulses, that also depends on the corresponding bit of a Gold code sequence c_i of length $N_c = 31$. Channelization is implemented by assigning a different Gold code to each user.

Suppose the bitstream is denoted by a sequence of binary symbols b_j (with values $+1$ for bit 1 or -1 for bit 0) for $j = -\infty, \ldots, \infty$. Let A denote the amplitude of each pulse $p(t)$. Then, the transmitted signal is

$$s_{\text{BPSK}}(t) = A \sum_{j=-\infty}^{\infty} \sum_{i=0}^{N_c-1} b_j c_i p\left(t - jN_cT_f - iT_f\right). \quad (1)$$

The data packet is comprised of a preamble and the payload. During the whole packet, the same Gold code is used. The preamble is composed of a sequence of pulses whose signs follow several repetitions of the Gold code, plus a final sequence of 31 pulses in which the Gold code is reversed. This last sequence represents a bit 0 (as opposed to the previous repetitions that represent a sequence of bits 1) indicates the end of the preamble and the beginning of the payload.

The time to achieve packet synchronization is a critical specification of any high-data-rate wireless system. The length of the preamble must be long enough to guarantee a high probability of achieving signal acquisition.

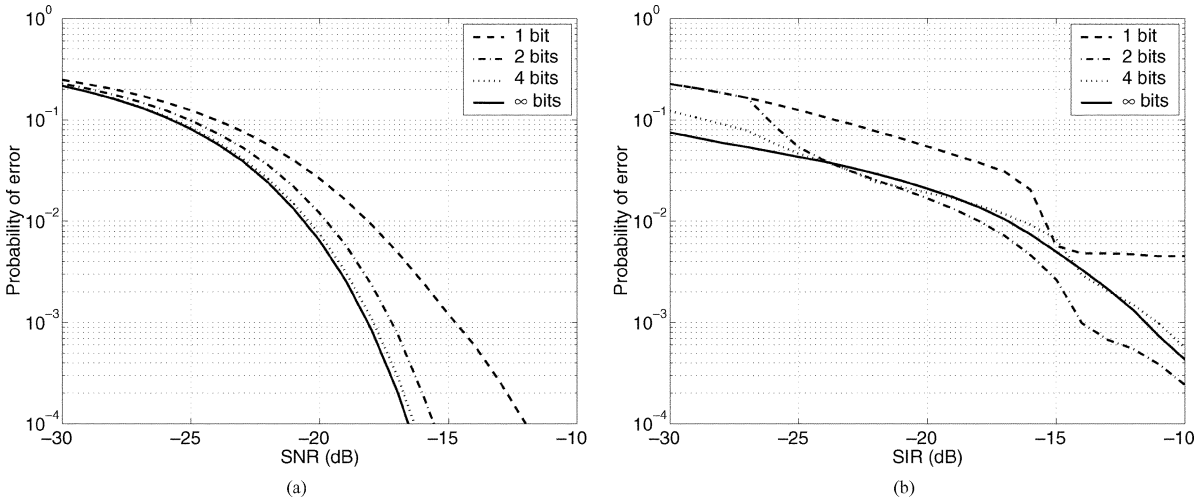

Fig. 1. BER as a function of (a) SNR or (b) SIR for different ADCs. (a) Noise-limited environment. (b) Interference-limited environment.

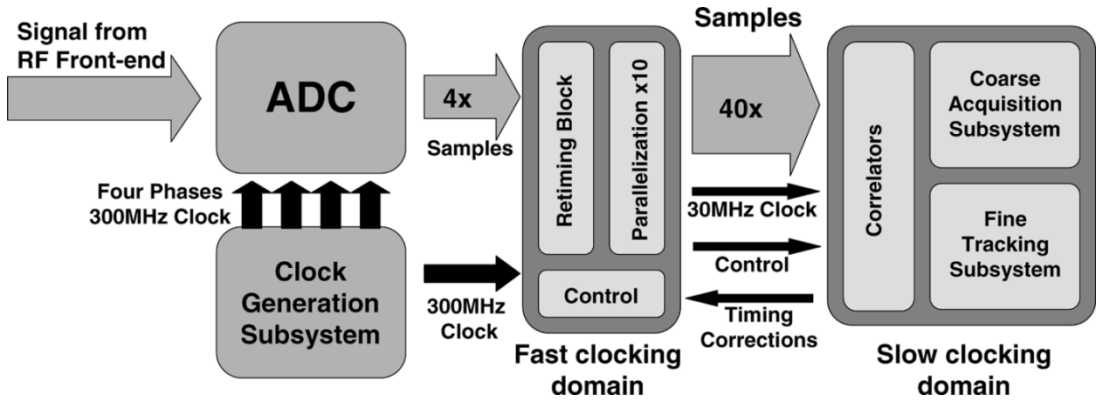

Fig. 2. Baseband processor block diagram.

III. System Tradeoffs

An important consideration in the receiver architecture is determining the analog/digital partition. In this paper, a digital architecture that also implements all the synchronization in the digital domain is chosen. The advantages are the simplification of the analog elements in the transceiver, its scalability, and the possibility of exploring digital channel adaptability and recovery. Performing the synchronization in the digital domain eliminates the need to feed a signal from the digital domain to the clock generation subsystem.

A digital architecture depends on the feasibility of the analog-to-digital converter (ADC) required to digitize the signal. To allow for an all-digital timing recovery, the ADC must sample at 1.2 Gsps, oversampling at twice the Nyquist rate. A Flash ADC architecture is well suited for such a high sampling rate. Since the power consumption in Flash ADCs scales exponentially with the number of bits of resolution, minimizing ADC resolution is crucial to reduce the power consumed in the receiver. Four bits of resolution are sufficient to be closer than 1 dB to the infinite resolution ADC curve for bit-error rate (BER). This is true in both a noise-limited environment where the signal is degraded by additive white Gaussian noise (AWGN) and in an interference-limited environment, where the signal is corrupted by a powerful narrowband sinusoidal interferer [6]. Fig. 1(a) shows the effects of ADC resolution on transceiver BER for different SNRs. Fig. 1(b) is the equivalent plot for the interference-limited case in terms of the signal-to-interference ratio (SIR).

The baseband UWB receiver uses a front-end that amplifies the incoming impulse signal [7]. After it, the baseband processor shown in Fig. 2 demodulates the signal. The following sections describe the blocks of the baseband processor: clock generation subsystem, the ADC, and the digital back-end.

IV. Clock Generation

Since the system implements a fully digital synchronization algorithm, the only input to the clocking system is the reference crystal clock, and its sole function is to track this reference. This is done with a phase-locked loop (PLL) based upon a design described in [8]. Fig. 3 shows the block diagram of the PLL. The jitter requirements are mostly constrained by the digital back-end of the receiver. Given that the probability of losing synchronization during a 1024-bits data packet with a rms clock jitter of 100 ps is smaller than 0.01, and the degradation in the SNR introduced in the ADC by the same jitter is smaller than 1 dB, a ring-oscillator-based VCO can be used to generate the 300-MHz clocks required for the ADC.

The ring oscillator consists of four differential inverters, producing the four 90-degree phase-shifted clocks that drive the

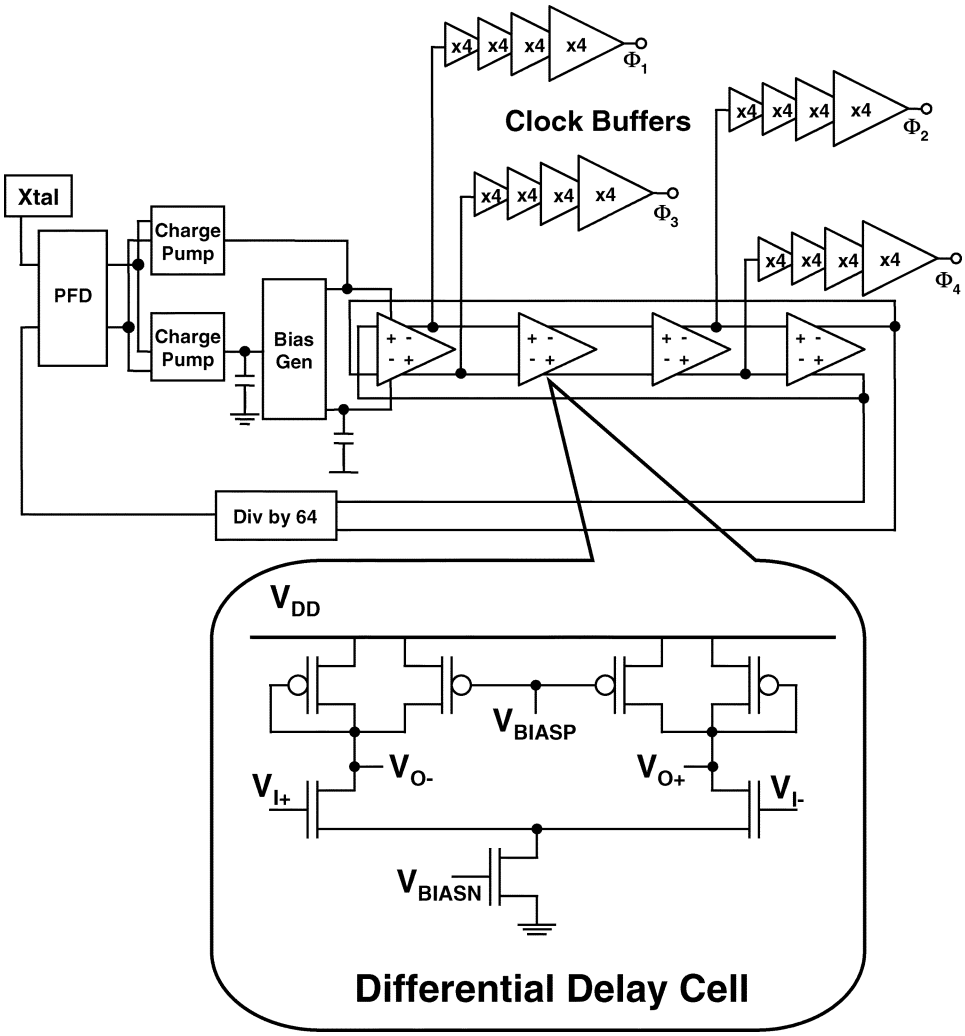

Fig. 3. Clock generation and distribution subsystem.

time-interleaved Flash ADC. The inverter buffers are differential pairs with symmetric loads.

The loop filter capacitor is 400 pF, which can be integrated on-chip. A second charge pump is used to provide a feedforward zero in the type III loop filter. The bias generator is a replica of the differential delay cells that is self-biased and generates the control voltages for the gate coupled logic loads and tail current so that they are biased at the maximum gain and swing points for the given current density and frequency of oscillation. The common circuit structures of the charge pump, bias generator, and delay cells allow this PLL to be process-independent and low-jitter.

One of the phases of the ring oscillator is used in the pulse transmitter included in the same integrated circuit. A reference pulse is generated by a counter that extracts one pulse every 50 cycles of the 300-MHz clock, achieving in this way the interpulse delay required between every two consecutive pulses. The sign of the pulses is controlled by the bit transmitted and by that of the Gold code. These digital signals are fed to a pulse amplifier, which consists of a 3:1 analog multiplexer, with inputs of power, ground, and a midpoint voltage. Positive pulses are relayed when power is latched to the output for 3.3 ns, and negative pulses when ground is latched for 3.3 ns. During idle time, the midpoint voltage is relayed, so that the pulse generator can transmit BPSK UWB signals. The template pulse that is formed from the mixed-signal pulse amplifier is filtered to approximate a Gaussian and is radiated by a monopole antenna of appropriate length.

V. Analog-to-Digital Converter

The 4-bit ADC in the UWB receiver is comprised of four Flash time-interleaved channels running on 300-MHz phase-offset clocks supplied by the PLL, achieving a sampling rate of $f_s = 1.2$ Gsps. Fig. 4 shows the block diagram of the ADC. Each channel contains a track-and-hold (T/H) circuit, followed by a bank of 15 preamplifiers, comparators, and a pipelined decoder. The T/H is comprised of only a switch and sampling capacitor due to the low resolution required by the system. The ADC core is fully differential for improved noise immunity and supports 500-mV peak input swings. These differential blocks are also shown in Fig. 4. The preamplifier reduces kickback noise to the input and reference nodes. The gain is provided by positive feedback in the two comparators. Comparator 1 is a track and latch comparator. Its speed advantage is due to its reduced swing. Comparator 2 is a StrongARM latch [9], slower but able to provide a full voltage swing. The pipelined decoder

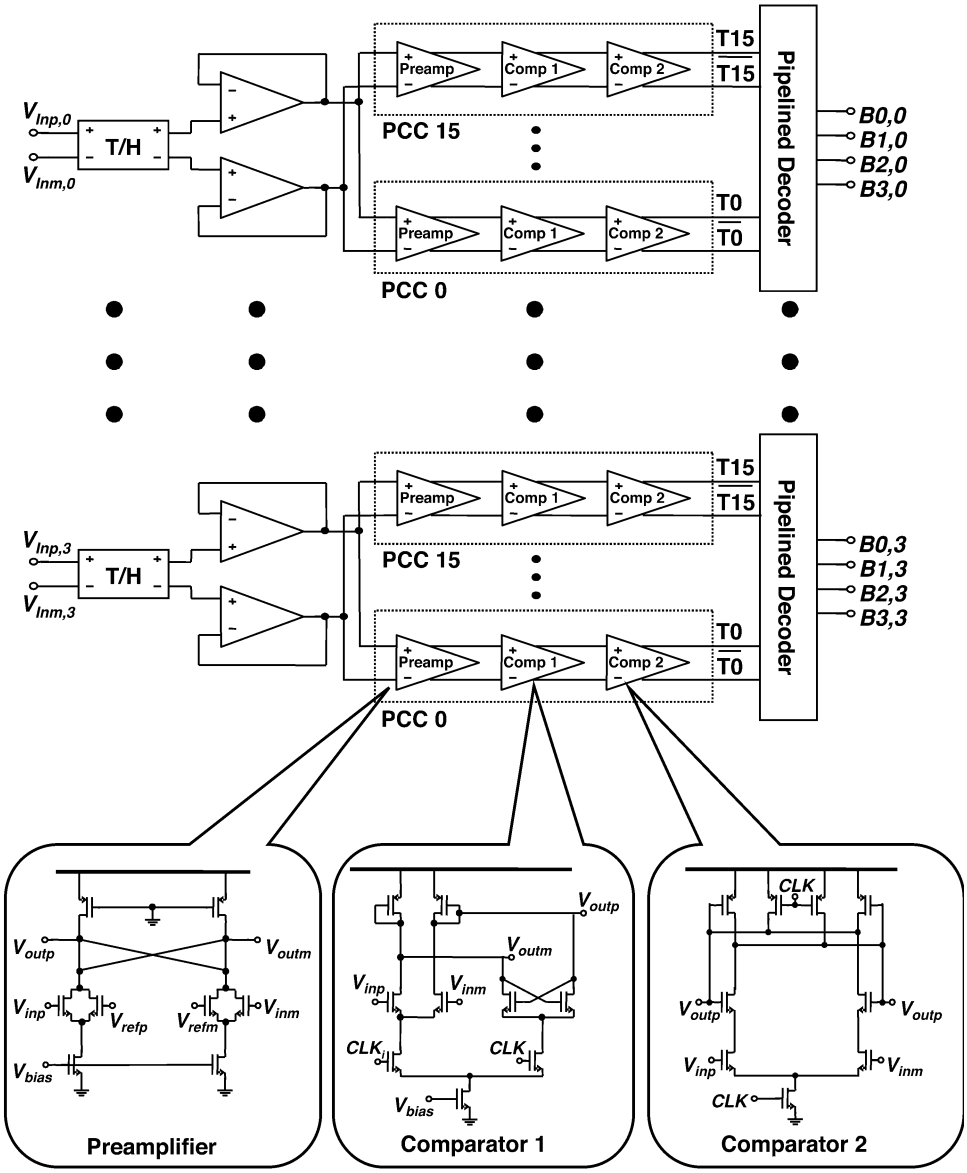

Fig. 4. Four-way time-interleaved Flash ADC and subcircuits.

uses an intermediate Gray code to reduce the impact of bubble errors in the performance of the ADC.

Traditionally, interleaved ADCs require calibration across channels. In this application, however, overall performance is determined by the average of all four channels rather than being limited by the worst case. This occurs because the digital back-end adds groups of four consecutive samples, coming from the four different channels, and treats the result as a single sample. This reduces the need for calibration across the channels as required in most time-interleaved ADCs.

The samples from the four channels are aligned to the same 300-MHz clock edge instead of creating a sample data rate clocked at 1.2 GHz. They are then presented in parallel to the digital back-end at this reduced data rate.

VI. Digital Back-End

The digital back-end implements the functionality required to synchronize and demodulate the data packet. The digital baseband implements the entire synchronization algorithm in the digital domain without feeding back any control signal to the other blocks, and to achieve packet synchronization in less than 70 μs. The input to this block is a vector of four consecutive samples given by the four-way time-interleaved Flash ADC, already aligned to the same edge of a 300-MHz clock.

A. Functionality

This receiver will be in one of two functional states: the coarse acquisition state looks for the presence of a data packet and achieves synchronization, and the fine tracking state performs the demodulation of the data packet after coarse acquisition was achieved. The following paragraphs present the specifications of the different blocks of the digital back-end.

1) Matched Filter: The digital back-end recovers the information contained in the data packets from the samples given by the ADC using an approximation to the matched filter [4]. A matched filter is the optimum demodulator of a signal in

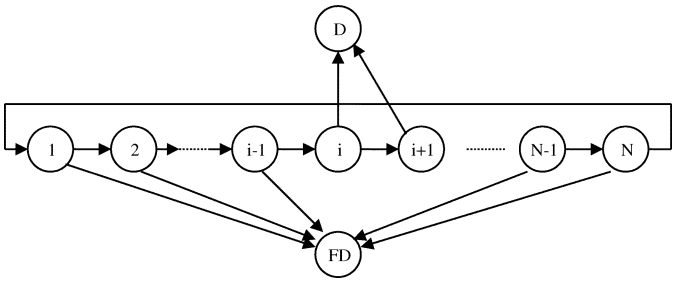

Fig. 5. Coarse acquisition process as a Markov chain (D = Correct detection; FD = False detection).

AWGN, and implies the correlation of the received signal with a local template synchronized to it and comprised of perfect replicas of the received pulses separated by the inter-pulse interval and whose signs coincide with that of the Gold code. This receiver uses a sequence of rectangular pulses instead of the perfect replicas, avoiding the use of multipliers. The correlation of the incoming signal with a rectangular pulse of width T_p is equivalent to adding four consecutive samples taken at 1.2 Gsps. T_p is chosen in order to maximize the processing gain. This simplification of the architecture comes at a cost of 1.7 dB of SNR for a Gaussian pulse compared to the ideal matched filter.

Since the correlation is implemented completely in the digital domain, the use of a larger T_p than the optimum does not accelerate the coarse acquisition process, and it has roughly the same number of mathematical operations as performing the correlation of the incoming signal with several differently delayed templates. Moreover, a larger than optimum T_p reduces the SNR at the output of the matched filter. Therefore, the width of the integration window is chosen to maximize the processing gain.

2) Coarse Acquisition: The coarse acquisition algorithm detects the presence of a data packet and estimates its delay with a precision of half the width of the pulse. For that purpose, it locates a peak of correlation between the incoming signal and a sequence of differently delayed local templates. The results of each of these correlations are compared to a predefined threshold, T_{h0}, and if this threshold is met, packet acquisition is declared. The complexity of this process grows with the signal bandwidth, the length of the Gold code, and the smaller duty cycle.

The difference in delay of two consecutive templates affects the number of opportunities available to detect the signal. A smaller difference implies a larger probability of detection of the signal, but also an increase in the complexity of the receiver. Setting this difference in delay equal to T_p gives a reasonable probability of detection of roughly 0.9 for a Gaussian pulse. This choice also simplifies the timing design in the receiver as all clocks have the same frequency.

If the transmitter and receiver clocks are assumed to have the same frequency, the coarse acquisition process may be modeled as a Markov chain, as shown in Fig. 5. The delay of the received pulses with respect to the different local templates is assumed to be constant. The states from 1 to $N = N_c \cdot N_f$ represent tests in which the received signal is correlated with a template with a different delay. The initial state can be any from state 1 to state $N = N_c \cdot N_f$. The states i and $i+1$ contain the pulses aligned with an error smaller than half the width of the pulse. If the pulse is detected there, it goes to state D, correct detection (with probabilities $P_{d,i}$ or $P_{d,i+1}$). The rest of the states do not contain the pulses. Any detection in those states implies a false detection (state FD) with probability P_{fd}. If no signal is detected, from each state it jumps to the next one.

TABLE I
MODEL RESULTS FOR A GAUSSIAN PULSE

P_{fd}	$E[k]$	P_{cd}
10^{-3}	526	0.48
10^{-4}	915	0.91
10^{-5}	1066	0.99

Using this information, it is possible to obtain the average number of iterations to achieve lock ($E[k]$) and the probability of correct detection (P_{cd})

$$E[k] = \frac{N\Delta + \sum_{n=1}^{N} nPr[n]}{1-\Delta} \tag{2}$$

$$P_{cd} = \frac{(P_i + P_{i+1}(1-P_i)) \prod_{j=1}^{i-1}(1-P_j)}{1-\Delta} \tag{3}$$

where $\Delta = \prod_{j=0}^{N-1}(1-P_j)$ and $Pr[n] = P_n \prod_{j=0}^{n-1}(1-P_j)$. It is assumed that the probabilities of declaring a detection for the slots j from 0 to $N-1$ is P_j. For states i and $i+1$, $P_i = P_{d,i}$ and $P_{i+1} = P_{d,i+1}$. For other values of j, $P_j = P_{fd}$.

Averaging these expressions for all the possible delays of the incoming signals, and choosing $N_c = 31$ and $N_f = 50$, Table I is obtained. The probability of correct detection drops sharply when $P_{fd} = 10^{-3}$. This happens because P_{fd} is comparable to $1/N$, and in N trials, a false detection may arise before there is an opportunity of testing the right delay. To ensure a reasonable probability of correct detection, P_{fd} must be much lower than $1/N$. The coarse acquisition algorithm designed for the implemented system assumes 50-way parallelization. Taking into account the values obtained for the Gaussian pulse and a $P_{fd} = 10^{-4}$, it results in an average time to declare coarse acquisition of 65 μs.

In this discussion, it has been assumed that the frequencies of the transmitter and receiver clocks were exactly the same. A large difference in frequencies spreads the total energy of the pulses across the correlation not only to two consecutive templates but three or more, reducing the probability of detection. It was proven that for the specified clock stability of 20 ppm, typical on crystal oscillators, the current specification is robust enough, and the loss in probability of detection is negligible.

3) Fine Tracking: Once packet synchronization is declared, depending on the length of the data packet, it may or may not be necessary to include a fine tracking algorithm to keep time synchronization for the duration of the data packet. For the specifications of the system, if the difference between the transmitter and receiver clocks is 20 ppm and there were no fine tracking mechanism, after 250 pulses half the energy of the incoming pulses is not included in the correlation with the local template with which the received signal was initially aligned. Since each bit is represented by $N_c = 31$ pulses, this allows a maximum of 8 bits in the packet.

For fine tracking, a classical delay-locked loop (DLL) [10] is used. It straddles the incoming pulses between two consecutive local templates, representing early and late versions of the

signal. The relative values of these correlations are used to estimate the delay of the incoming signal.

Since all of the timing control is performed in the digital back-end, it is not possible to continuously adjust the delay of the local templates generated in the receiver with respect to the incoming signal. The only feasible delay adjustments are an integer number of samples. Due to the architecture used, only a limited range of delay corrections can be applied. The architecture presented in the next section allows for corrections of -3, -2, -1, 0, 1, 2, and 3 samples. It was tested through simulation that this is sufficient for this system.

B. Architecture

The digital back-end is divided into a fast and a slow clocking domain, as shown in Fig. 2. The fast domain uses a custom layout, a 300-MHz clock coming from the PLL, and is composed of a retiming block, a block that performs a $10\times$ parallelization of the incoming ADC samples, and the main control of the digital back-end. The parallelization allows the slow domain to work with a 30-MHz clock. The correlations and other mathematical operations needed to implement the synchronization and demodulation are performed in 2's-complement arithmetic in the slow domain, using standard ASIC flow.

The retiming block provides the one-sample delay granularity required for fine tracking. The groups of four samples that are the inputs to the correlators may start with any arbitrary sample and may include samples belonging to two different ADC vectors. These groups are obtained by selectively delaying the outputs of one or more ADC interleaved channels in the retiming block. This block is controlled by a four-state finite state machine whose value is updated based on the relative delay of the incoming signal with respect to the local templates.

After the parallelization, the outputs of the fast clocking domain are processed by 10 correlators as shown in Fig. 6. In each 30-MHz clock cycle, the four samples at its input are added together, implementing the correlation with a rectangular pulse of width T_p. The result of this addition is either added or subtracted, depending on the value of the Gold code, to the 11-bit value stored in the shift registers five cycles before ($50 \cdot T_p$, equal to the time between two consecutive pulses). Since the time between two consecutive pulses is equal to five cycles, in order to cope with this duration, coarse acquisition is decided only upon $N_c - 1$ pulses instead of N_c pulses. All multipliers in Fig. 6 are implemented using 2-to-1 multiplexers because in each of them one of the coefficients represents a single bit. Each correlator performs five correlations at the same time, equivalent to the output of a finite impulse response (FIR) filter of $D_{\text{bit}} \cdot f_s = 6200$ coefficients with values equal to 1, -1, or 0. The outputs from the ten correlators (a total of 50 correlation values) are used by the coarse acquisition subsystem, but only the first two are active during fine tracking. The Gold code generator is implemented with two shift registers, each of them generating a linear recursive sequence of which both the coefficients of the generating polynomial and the seed values are programmable. If the incoming signal is properly aligned to one of the first local templates in a correlation, at the end of the iteration, the 50 correlations contain the samples of the channel impulse response. This information was not further used in this transceiver, but it may be used in future prototypes in a RAKE receiver to compensate for the multipath and in an MLSE to make up for the inter-symbol interference if the system uses an inter-pulse interval shorter than the channel impulse response.

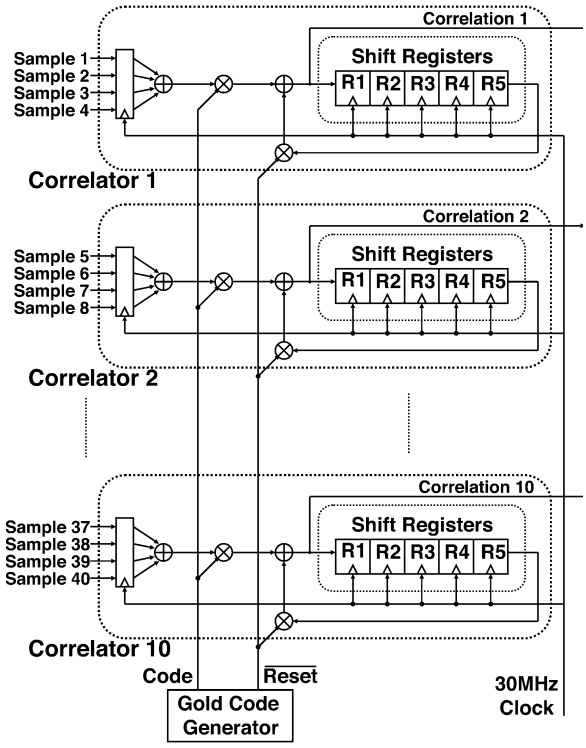

Fig. 6. Implementation of the correlation bank.

The duration of a correlation iteration is $(N_c+1)\cdot 133$ ns. The extra 133 ns provides additional time to compare the correlation results to T_{h0}, and, if packet synchronization is not declared, it is used to delay the position of the local templates T_f with respect to the incoming signal.

The fine tracking subsystem shown in Fig. 7 provides the functionality required to close the DLL. The division needed in the delay estimation is avoided by multiplying by an approximation to the inverse stored in a ROM. The ROM stores 32 seven-bit numbers and the five most significant bits of the numerator are used to choose the output. The coefficients of the filter are programmable, and it incorporates Baugh–Wooley multipliers. The delay decoder transforms the output of the filter into signals relevant to the fast clocking domain: the new state of the retiming block and indication of the need to start correlations a 300-MHz clock cycle before (signal Advance) or later (signal Delay). The fine tracking subsystem also provides a flag to restart coarse acquisition when the signal is lost. All the outputs of the fine tracking system are ready in six 30-MHz cycles. Since there are only five 30-MHz cycles between every two consecutive pulses, there is not sufficient time between the last pulse of the Gold code sequence and the first pulse of the next sequence to perform this operation. This is the reason why the DLL only uses $N_c - 1$ pulses to estimate the delay even if N_c pulses are used to recover the value of the transmitted bit. This leads to a negligible loss of processing gain for the delay estimation.

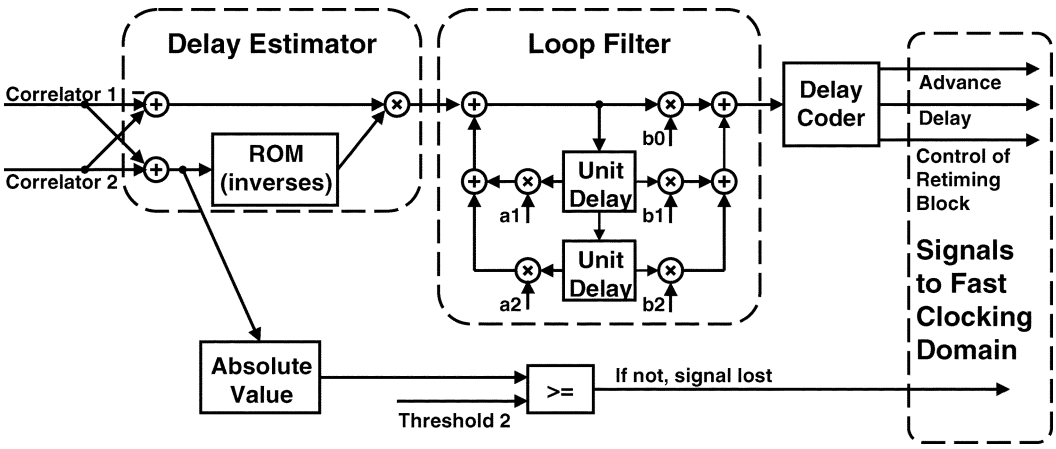

Fig. 7. Fine tracking subsystem block diagram.

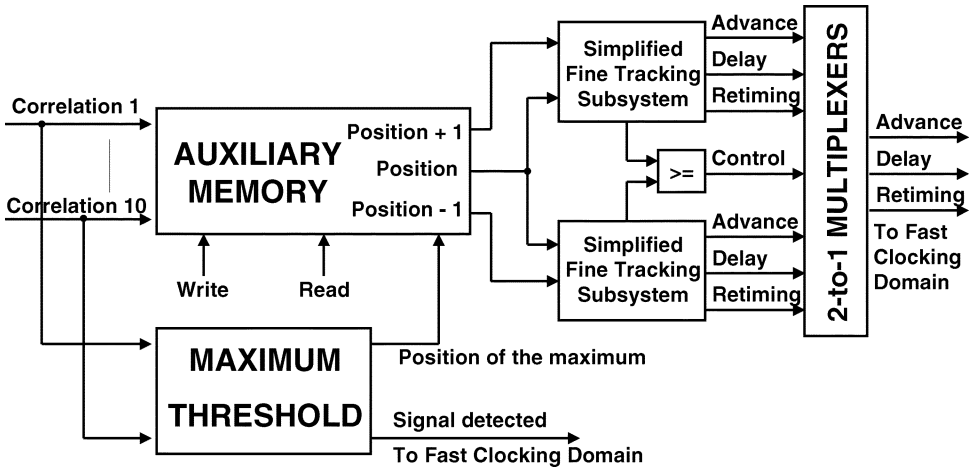

Fig. 8. Coarse acquisition block diagram.

As the 50 correlations are completed, they are read into the coarse acquisition subsystem, whose block diagram is shown in Fig. 8. The memory block provides not only the value of the maximum but also the values in the two adjacent positions. The two simplified fine tracking subsystems lack the loop filter shown in Fig. 7 except for its direct path (b_0). Only the simplified fine tracking subsystem using the two positions with the most energy will be used to initialize the DLL. Detection of the signal is given in six 30-MHz cycles, and the rest of the outputs are ready in seven cycles thereafter. In order to provide enough time interval for these operations, the evaluation of the 50 correlations starts only when N_c-1 pulses have been integrated. Still, during the change of state that happens after declaring packet synchronization, the first two pulses of the next bit are lost. This implies a negligible loss of processing gain in the demodulation of the first bit of the payload.

All thresholds, coefficients, and other parameters used in the digital back-end must be configured before utilization by using data fed through a serial port.

VII. Performance Results

Fig. 9 shows a photograph of the 0.18-μm ASIC. The PLL was verified at 300 MHz and can provide much higher clock frequencies (up to 2 GHz). The ADC is verified using the testing method presented in [11], in order to measure both its differential nonlinearity and its integral nonlinearity. As these values are either below or very close to half an LSB, the effective number of bits of this ADC is greater than 3. The digital back-end is completely functional at a clock frequency of 300 MHz. The frequency range for the coarse acquisition algorithm between a pair of transceivers is shown to be $\pm 3\%$. At 300 MHz, a data

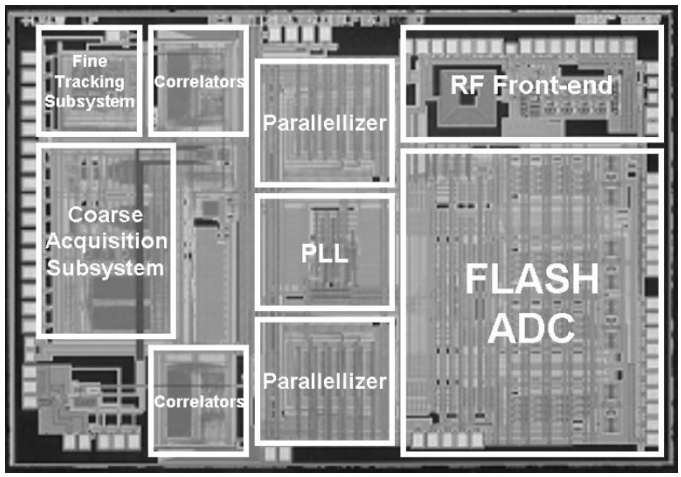

Fig. 9. Single-chip UWB transceiver photograph.

TABLE II
CHIP MEASUREMENTS

Chip specifications		
Process Technology	0.18 μm	
Die Size	4.3 mm $\times$ 2.9 mm	
Bit Rate	193 kbps	
ADC Performance	Abs. Accuracy	Rel. Accuracy
Static Performance	3.1 bits	3.9 bits
Dynamic Performance	3.0 bits	3.4 bits
Power Consumption		
ADC	86 mW	
PLL	45 mW	
CLK Buffers	65 mW	
Back-End	75 mW	
Total	271 mW	

rate of 193 kb/s was demonstrated. Table II contains a summary of overall chip measurements. Most of these circuits can scale to 500 MHz. This architecture is scalable to larger bandwidths.

VIII. CONCLUSION

The baseband processor for a UWB system-on-a-chip transceiver with a mainly digital architecture was presented in this paper. The chip was implemented in 0.18-μm CMOS technology at 1.8 V. The baseband processor consumes 271 mW. This processor includes the digital back-end, a 4-bit Flash time-interleaved ADC and a clock generation subsystem that includes a four-stage ring oscillator easily meeting the jitter rms requirement of 100 ps. Extensive parallelization was used in the digital back-end in order to reduce the time to packet synchronization to 65 μs, while maintaining the whole processing gain that the UWB signal can provide. The entire synchronization is performed in the digital domain without feeding any signals back to the front-end or the clock generation subsystem.

REFERENCES

[1] "Ultra-Wideband (UWB) First Report and Order," Federal Communications Commission, 2002.

[2] C. J. Le Martret and G. B. Giannakis, "All digital PAM impulse radio for multiple access through frequency selective multipath," in *Proc. IEEE Global Telecommunications Conf.*, 2000, pp. 77–81.

[3] ——, "All digital PPM impulse radio for multiple access through frequency selective multipath," in *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2000, pp. 22–26.

[4] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.

[5] K. Siwiak, P. Withington, and S. Phelan, "Ultra-wide band radio: The emergence of an important new technology," in *Proc. IEEE Vehicular Technology Conf.*, 2001, pp. 1169–1172.

[6] P. Newaskar, R. Blazquez, and A. Chandrakasan, "A/D precision requirements for an ultra-wideband radio receiver," in *Proc. IEEE Workshop on Signal Processing Systems (SIPS)*, 2002, pp. 270–275.

[7] F. Lee, D. Wentzloff, and A. Chandrakasan, "An ultra-wideband baseband front-end," in *Proc. IEEE Radio Frequency Integrated Circuits Symp.*, Fort Worth, TX, 2004, pp. 493–496.

[8] J. Maneatis, "Low-jitter process-independent DLL and PLL based on self-biased techniques," *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1722–1732, Nov. 1996.

[9] A. Chandrakasan, W. Bowhill, and F. Fox, Eds., *Design of High Performance Microprocessor Circuits*. New York: IEEE Press, 2001.

[10] H. Meyr and G. Ascheid, *Synchronization in Digital Communications: Phase-, Frequency-Locked Loops and Amplitude Control*. New York: Wiley Interscience, 1990, vol. 1.

[11] J. Dornberg, H. S. Lee, and D. A. Hodges, "Full-speed testing of A/D converters," *IEEE J. Solid-State Circuits*, vol. 19, no. 12, pp. 22–26, Dec. 1984.

Raúl Blázquez received the Ingeniero de Telecomunicación degree from the Technical University of Madrid, Madrid, Spain, in 1999. During 1998–2000, he was with the Signals, Systems and Radiocommunication Department, Technical School of Telecommunications Engineering, Madrid, Spain, as a Research Assistant. He is currently with the Massachusetts Institute of Technology, Cambridge, working toward the Ph.D. degree in electrical engineering.

His research interests include power efficient signal processing algorithms, low-power digital circuits, and ultra-wideband radio circuits and systems.

Mr. Blázquez was a co-recipient of the DAC/ISSCC Student Design Contest Award in 2004. He was the recipient of a "La Caixa" Fellowship in 2000 and an Intel Ph.D. Fellowship in 2004.

Puneet P. Newaskar received the B.S.E.E. degree from the California Institute of Technology (Caltech), Pasadena, in 2001, and the M.S.E.E. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 2003. His Masters research focused on the design of high-speed data converters for ultra-wideband radio transceivers.

He is currently a Design Engineer in the Wireless Products Division of Silicon Laboratories, Inc.

Mr. Newaskar was a recipient of the Presidential Fellowship at MIT.

Fred S. Lee received the B.S.E.E. and M.Eng. degrees from the Massachusetts Institute of Technology, Cambridge, in June, 2002, where he is currently working toward the Ph.D. degree in electrical engineering.

His research interests include ultra-wideband radio receiver circuits and systems, CMOS/SiGe circuit design, low-power mixed-signal circuits, and analog RF circuit design.

Mr. Lee was a co-recipient of the DAC/ISSCC Student Design Contest Award in 2004.

Anantha P. Chandrakasan (M'95–SM'01–F'04) received the B.S, M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1989, 1990, and 1994, respectively.

Since September 1994, he has been with the Massachusetts Institute of Technology, Cambridge, where he is currently a Professor of Electrical Engineering and Computer Science. His research interests include low-power digital integrated circuit design, wireless microsensors, ultra-wideband radios, and emerging technologies. He is a co-author of *Low Power Digital CMOS Design* (Kluwer, 1995) and *Digital Integrated Circuits* (Pearson Prentice-Hall, 2003, 2nd edition). He is also a co-editor of *Low Power CMOS Design* (IEEE Press, 1998) and *Design of High-Performance Microprocessor Circuits* (IEEE Press, 2000).

Dr. Chandrakasan has received several awards, including the 1993 IEEE Communications Society's Best Tutorial Paper Award, the IEEE Electron Devices Society's 1997 Paul Rappaport Award for the Best Paper in an EDS publication during 1997, and the 1999 Design Automation Conference Design Contest Award. He has served as a technical program co-chair for the 1997 International Symposium on Low Power Electronics and Design (ISLPED), VLSI Design '98, and the 1998 IEEE Workshop on Signal Processing Systems. He was the Signal Processing Sub-committee Chair for ISSCC 1999–2001, the Program Vice-Chair for ISSCC 2002, the Program Chair for ISSCC 2003, and the Technology Directions Sub-committee Chair for ISSCC 2004–2005. He was an Associate Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS from 1998 to 2001. He serves on the SSCS AdCom and is the meetings committee chair. He is the Technology Directions Chair for ISSCC 2006.