WHITE **PAPER**

cadence®

Xtensa Radar Library for Tensilica DSPs

Sep 2022

Radar technology is playing an increasingly important role in a wide range of applications, resulting in the need for easily adaptable and configurable software components to meet the demands of target applications. This paper introduces different radar signal processing modules provided as part of the Xtensa Radar Library for Tensilica DSPs. It also covers various features of the modules. The Xtensa Radar Library facilitates the development of radar signal processing software systems by choosing suitable variants of various module implementations readily available with the best possible performance. Suitable implementation for each module can be chosen based on the requirements and demands of various applications. The Tensilica ConnX DSP Instruction Set Architecture (ISA) is very well-tuned for radar signal processing applications. This library uses these ISAs efficiently to achieve the best performance.

Contents

Introduction

Radar applications today range from traditional military or security to civil and production technology, home security appliances, automotive, medical, and much more. Furthermore, radar sensing technology is growing at a significant pace. With the evolution of Automotive Driver Assistance Systems (ADAS) and the consumer electronics market, the radar segment is expected to grow significantly in many more areas in the coming years. Key sensing technologies used in automobiles are radar, lidar, camera, and ultrasonic, with radar being used inevitably and extensively due to its unique features.

Keeping the above requirements in perspective, this paper introduces Xtensa Radar Library on Cadence® Tensilica® DSPs, currently supporting the ConnX B10 and ConnX B20. In this library, various implementations of radar processing modules like Range FFT, Doppler FFT, 2D Constant False Alarm Rate (CFAR), and Object tracking are provided, which are described in detail in the following sections of the paper. These modules are available with different input and output precisions, data types, and configurable options to meet the needs of different target applications. Users can experiment and make suitable choices for each module. The paper also briefs about the capabilities and features of Tensilica ConnX B10 and ConnX B20 DSPs.

Xtensa Radar Library

The radar signal processing (RSP) chain is used primarily for object/target detection and tracking. This paper describes a typical RSP chain, as Figure 1 shows. Additionally, this section details a list of modules available in the current version of the Xtensa Radar Library.

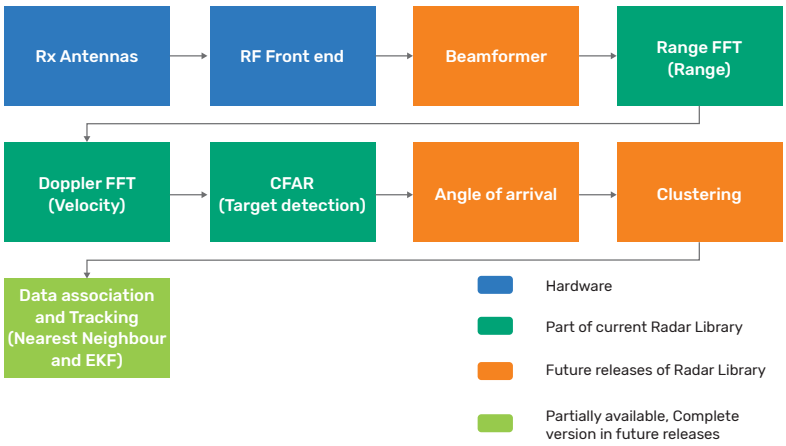

Figure 1: Typical Radar signal processing flow

Radar systems can have multiple transmit and multiple receive antennas. Depending on their numbers, the radar system can be categorized into Single Input Single Output (SISO, single transmitter, single receiver), Single Input Multiple Output (SIMO, Single transmitter, multiple receivers), Multiple Input Multiple Output (MIMO, Multiple transmitter, and multiple receivers) systems [7]. Generally, SISO can only provide the range and Doppler speed of the targets. The SIMO system in Figure 2 is useful when the direction (generally termed as the angle of arrival) of multiple targets needs to be found. A MIMO system can increase the resolution of the angle of arrival estimation. Additionally, multiple transmit antennas allow electronically changing the direction of transmission. The same can be done for reception with multiple receiver antennas, called beamforming. In the case of SIMO, the output of the beamformer is multiple 2D images corresponding to different directions, which FFTs then process. In the case of frequency-modulated continuous wave (FMCW) radar, Range FFT is used to find the range and Doppler FFT for the Doppler speed of the targets. The output of these modules is a Range Doppler Image (RDI) with the range and Doppler speed information of targets, which is proportional to the coordinates of pixels/cells in RDI with local maxima. Multiple cells with local maxima in RDI can correspond to the noise. To filter out the noise and detect cells associated with the target, a CFAR module is used. In the case of multiple antenna systems, we obtain multiple RDIs. We can use coherent or non-coherent integration algorithms to get one RDI, which CFAR processes. For SIMO, and MIMO systems, the angle of arrival of the signal, which is the direction of the target is computed with the Root MUSIC or angle FFT algorithm [9].

From all previous modules, we get measurements such as range, angle of arrival, and Doppler speed. Multiple measurements may correspond to the same target. To avoid unnecessary further processing of redundant data, clustering algorithms such as density-based spatial clustering of applications with noise (DBSCAN) can be used to extract one measurement for one target.

Kalman filter is a recursive algorithm that can be used to estimate the states, such as the position and velocities of the object using imperfect measurements observed over time under noisy conditions. For object tracking, Extended Kalman Filter (EKF) can be used to estimate the states of the nonlinear systems. Algorithms such as nearest neighbor are used to associate new measurements with objects tracked by the estimation algorithm. The current Xtensa Radar library has modules related to Range FFT, Doppler FFT, CFAR, and EKF.

APIs in the Xtensa Radar Library can be used for Pulse Doppler Radar, FMCW Radar, Digitally Modulated Radar, and its variants by appropriately selecting the configurations of applicable APIs. The RF front-end of the radar will change according to the chosen scheme. The received data from ADC will form a data cube for the SIMO system, as Figure 3 shows. Generally, the three dimensions of the data cube are range dimension (time sampling of one chirp with N_r samples), Doppler dimension (N_d chirps), and channel dimension (N_c antenna element). The dimension of the data cube depends on the sampling frequency, the number of chirps transmitted per frame, and the number of transmitters and receivers, etc. These quantities are decided based on radar processing requirements.

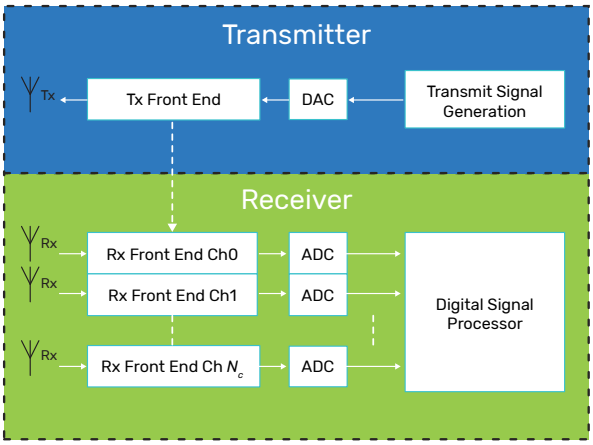

Figure 2: Typical SIMO radar system (N_c Receiver) system

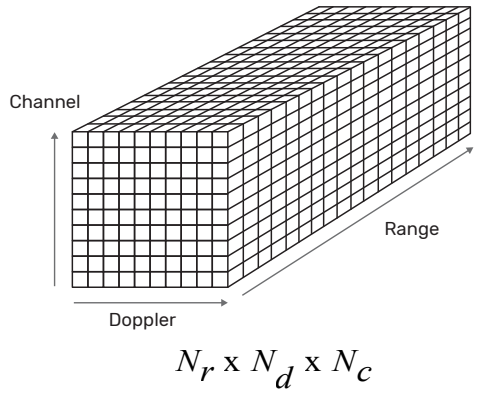

Figure 3: Data cube for SIMO FMCW Radar

Signal processing requirements may change between customers as per their application needs. To cater to this, the Xtensa Radar Library is abstracted to deliver a wide range of configurable options with a few top-level APIs. The following is the list of available modules in this release.

FFT

The received signal in Radar needs to be processed to extract parameters related to the target. These parameters can be the distance, velocity, and direction of the target with respect to Radar. Many such parameters can be estimated by analyzing signals in the frequency domain. Xtensa Radar library uses range FFT and Doppler FFT APIs to accomplish this. FFT is used to transform the signals, with range FFTs on the range dimension, followed by Doppler FFTs on the Doppler dimension. Each FFT API can use user-provided window coefficients. The FFT algorithm is based on the decimation-in-frequency (DIF) Kronecker product-based formalization [6].

Range FFT

Range FFT API can determine the range of targets in the FMCW Radar. The target's range is proportional to the frequency of the sampled baseband signal. FFT results in local peaks in the range dimension, which correspond to the range of the target.

Various types of FFT implementation will be required based on the design decision. These can be accessed through a single API that supports independent configurable options to choose from. The following are various configurable options:

- **Precision:** This feature enables choosing FFTs implemented in specific precision. Precision can be a 16-bit fixed point, 32-bit fixed point, or single precision (SP) floating point. SNR of FFTs improves with precision.
- **Windowing:** When sampling an unknown signal, the boundaries of the observation period can show discontinuities. Such discontinuity causes spectral leakage in the FFT output, which is not present in the original signal. Spectral leakage can be reduced by minimizing discontinuity. This can be done by multiplying input signal samples with a window whose amplitude gradually decreases to zero towards the boundary. Different types of windows can be used for different requirements. The Xtensa Radar Library gives the flexibility of calculating FFTs of input samples with a given window or without a window as per need. Window signal can be precomputed and provided as input to the FFT API.
- **Data type:** Configurable input data type from real or complex can be selected based on the input.
- **FFT Size:** Sizes of FFTs available are power of 2 from N_r = 16 to 4,096. A wide range of optimized FFT modules is available through a single API based on the desired resolution of the range.
- **Number of FFTs:** FFT of multiple signals of the same size can be computed in one go. API is implemented considering the block of input data, and thus, capable of computing FFTs of multiple chirps and channels.

Range FFT in the context of FMCW can be directly used in a radar processing chain. In a single input and single output (SISO) radar system, it can process N_d number of chirps with FFT size of N_r. The same API can be used for a SIMO or MIMO system by placing input signal data in one dimension. The signal from multiple channels can be processed using range FFT by appending (N_c) blocks of signal in L dimension of range FFT.

For more details about Range FFT API, refer to Xtensa Radar Library User Guide [1].

Doppler FFT

Each subsequent received chirp has a phase change if the target is moving. The phase change rate will be proportional to the target's speed. This can be captured in the frequency domain. This is where Doppler FFT API comes in handy. It calculates the FFT of the input complex signal. For catering to various design aspects, Doppler FFT API is packed with the following configurable features that can be selected independently.

- **Output Type:** Interfacing Doppler output to other modules will require various types of output. API can give output in the form of FFT, squared magnitude, magnitude, or log magnitude. Except for FFT, other outputs are useful in interfacing with the CFAR module. The FFT output will be useful for further processing and angle extraction of the target.
- **Precision:** This feature enables choosing FFTs, squared magnitude, magnitude, and log magnitude implemented in specific precision. Precision can be a 16-bit fixed point, 32-bit fixed point, or SP floating point. For 16-bit precision, to avoid precision loss, outputs are computed in 32 bits except for FFTs.
- **FFT Size:** The sizes of FFTs available are from N_d = 16 to 1,024 with powers of 2. A wide range of optimized complex FFT modules is available through a single API.

This module can process multiple streams of input data. Doppler FFT takes complex input in a streaming format to easily use the output of range FFT. This avoids unnecessary transposing of range FFT output. Like range FFT, Doppler FFT API provides a configurable option for applying a window to the input.

FFT can be used in Pulse-Doppler, FMCW Radar, or digitally modulated Radar (DMR) to obtain the target's velocity. In the context of FMCW Radar, Range FFT followed by Doppler FFT will generate Range-Doppler Image of multiple channels. Multiple calls of Doppler FFT API can be used to process data from multiple channels.

Refer to Table 1 for a summary of all configurable options in FFTs.

Parameters	Range FFT		Doppler FFT	
Input to Module	Block signals		Streaming signal	
	Window signal (Optional)		Window signal (Optional)	
Output of Module	1. FFT		1. FFT	
			2. Magnitude of FFT	
			3. Square Magnitude of FFT	
			4. Log Magnitude of FFT	
Input and Output Precisions	Input	Output	Input	Output
	16b Fixed point		16b Fixed point	1. FFT: 16b
				2. Magnitude of FFT: 32b
				3. Magnitude square of FFT: 32b
				4. Log Magnitude of FFT: 32b
	32b Fixed point		32b Fixed point	All Output type: 32b
	Single Precision Floating Point		Single Precision Floating Point	All Output type: SP
Input data type	1. Real		1. Complex	
	2. Complex			
FFT Size	N_r = 16 to 4096 (powers of 2)		N_d = 16 to 1024 (powers of 2)	

Table 1: Configurable parameters for FFTs

Local peaks in the range-Doppler image represent targets with range coordinates corresponding to the range and Doppler coordinates to the velocity of that target. Owing to noise and multipath interference, many of the local peaks in the Range-Doppler Image may not be actual targets. A constant false alarm rate module is used to avoid false target detection.

Constant False Alarm Rate (CFAR)

The CFAR kernel operates on the energy signal in the frequency domain, classifying each bin or cell under test (CUT) in the FFT energy signal that is a Range-Doppler Image either as a target bin or noise/clutter bin by comparing each CUT with a scaled estimate of noise + clutter. The scale factor and threshold used are determined by the probability of a false alarm (thus the name CFAR). Noise + clutter is estimated by considering the cells in a rectangular neighboring region for a given CUT, leaving out some immediate neighbors of the CUT (called the guard region) due to potential energy leakage from the CUT into its immediate neighbors. The cells used to compute the adaptive threshold for a given CUT are called training cells.[10]

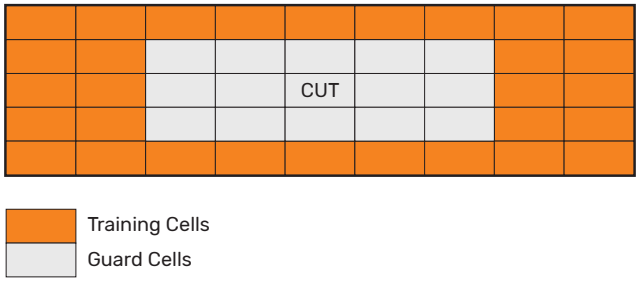

Figure 4: CFAR processing window

If the CUT value is greater than the scaled noise + clutter estimate, then the CUT is classified as a target, otherwise, the CUT is classified as noise/clutter. The CFAR algorithms are distinguished by how the noise estimate is formed using the training cells. For example, the Cell-Averaging Constant False Alarm Rate (CA-CFAR) noise estimate is formed by taking an average of the cells in the training area. In contrast, the Ordered Statistic Constant False Alarm Rate (OS-CFAR) noise estimate is formed by sorting the training area's cell values in descending order and using the Nth sorted value. Various combinations of CA-CFAR and OS-CFAR can also be used. The AND CFAR combination does a logical AND of the classification outcomes of the CA-CFAR and OS-CFAR for each CUT, while the OR CFAR combination does a logical OR.

The current release contains the API for CA-CFAR. The CFAR API is equipped with the following configurable features to cater to various design requirements:

- **Precision:** This feature allows choosing a CFAR algorithm implemented in a specific precision. Precision can be a 16-bit fixed point, 32-bit fixed point, or SP floating point. To avoid precision loss due to saturation, the release also contains the extra flavor of CFAR that uses wide vectors for either fixed point precision.
- **RDI Specifications:** CFAR API allows for configurable RDI, window, and guard size. Additionally, the scale factor can be given as input to the CFAR, which is used to scale the computed noise threshold to determine the target.
- **Noise threshold:** Xtensa Radar Library can check the computed noise threshold for all cells used to decide the target's presence.

Parameters	CA-CFAR		
Input to Module	Range Doppler Image size		
	Window size, Guard Size, Scale Factor		
Output of Module	Black and white image corresponding to the target		
	Noise threshold (Optional Output)		
Input and Output Precision	Input	Noise threshold Output	Target detect Output
	16b Fixed point	1. 16b Fixed Point	Binary output
		2. 32b Fixed Point	
	32b Fixed point	32b fixed point	
	Single Precision Floating Point	Single Precision Floating Point	
Input and Output Precision	1. 16b Fixed point		
	2. 32b Fixed point		
	3. Single precision floating point		
Implementation precision	1. 16b Fixed point		
	2. 32b Fixed point		
	3. 16b Fixed point with wider accumulators		
	4. 32b Fixed point with wider accumulators		
	5. Single precision floating point		
Output precision	Binary output		
	Noise threshold (Optional Output)		
Input data type	Real		

Table 2: Configurable parameters for CFAR

The output of CFAR gives information about the presence or absence of a target in the Range-Doppler image. There can be many bins or measurements associated with the same target. The measurement for the target may not be available for some time due to noise, occlusion, and other practical scenarios. To correlate past or noisy data and visualize or track a real target in the absence of data requires various filtering algorithms that track detected targets.

Tracking

Target tracking typically aims to estimate the number of targets, and their states, such as locations, velocities, radar cross-section, etc., given uncertainties in the measurements [8]. Typically, the Kalman filter and its other variants, such as the extended Kalman filter, unscented Kalman filter, and particle filters are used for state estimation of targets.

In this release of the Xtensa Radar Library, we provide an extended Kalman filter (EKF) module implemented in single precision. This module has two main APIs: one for prediction and another for updating the state. Multiple subroutines that are part of predict and update APIs also enable easy customization of filters according to different use cases. The state of targets with nonlinear dynamics can be estimated using the EKF module. Owing to the generic nature of EKF implementation, EKF API can be used for other use cases related to the state prediction of any nonlinear systems. The 2D constant velocity motion model for tracking targets is part of the Xtensa Radar Library suite that can be readily used for modeling targets in the radar use case. For the constant velocity model, measurement consists of range, target angle, and the target's Doppler speed. In the current implementation, one instance of EKF API can track one target. Multiple targets can be tracked using multiple instances.

Configurable options for EKF modules are

- **EKF specification:** State, measurement, and control size in multiple of 4 can be selected. Currently, a 2D constant velocity model with no input and state size N=4, and measurement size 3 (padded with a zero to get M=4 dimension), and control size C=0 is provided. (Note: EKF is tested only for N=4, M=4)
- **Transition and Jacobian Functions:** The library uses specific transition and Jacobian functions for a constant 2D velocity model. Users can use the configurable option to use their own implementations for state transition, state Jacobian, measurement transition, and measurement Jacobian function.
- **Initial condition of state, measurement, and process noise:** The user needs to provide these as input parameters per the user system model.

Refer to Xtensa Radar Library User Guide [1] for more details.

Parameters	EKF
Input to Module	The initial condition of the target
	Process and measurement noise
	Sensor Measurement
	Definitions of state transition and Jacobian functions
	Definitions of measurement transition and Jacobian functions
Output of Module	State estimate of the system
	Covariance
Input precision	Single precision
Output precision	Single precision
Input data type	Real
Constant velocity model	By default, optimized code of transition and Jacobian functions of a state and measurement are provided for a 2D scenario for sensor measurement of range, azimuth angle, and radial velocity.

Table 3: Configurable parameters of Extended Kalman Filter

A developer can refer to Xtensa Radar Library User Guide [1] for more details on API usage. For the Xtensa Radar library performance for these module APIs, please refer to the Xtensa Radar Library Performance documents for ConnX B10 [2] and ConnX B20 [3].

ConnX DSP Features and Configuration

The current Xtensa Radar Library is available on both ConnX B10 and B20 DSPs. These two DSPs are based on an ultra-high-performance NX architecture with a 10-stage pipeline for next-generation complex signal processing applications. ConnX DSPs combine SIMD architecture with an up-to-5-issue VLIW processing pipeline and a rich and extensible set of interfaces.

The ConnX B10 and B20 DSPs are built around a core vector pipeline consisting of 32 and 64 16-bit x 16-bit Multiply-Accumulate (MAC) units, respectively, along with a set of versatile pipelined execution units. ConnX B10 and B20 DSPs optionally support IEEE compliant single precision, half-precision, double-precision vector floating point Fused Multiply and Adds (FMAs). Some of the optionally supported units in ConnX B10 and B20 DSPs are shown in 8.

Features	Tensilica ConnX DSPs	
	ConnX B10	ConnX B20
16-bit x 16-bit MACs	64	128
32-bit x 32-bit MACs	16	32
Single-precision floating point FMAs	16	32
Half-precision floating point FMAs	32	64
Double-precision floating point FMAs	8	16

Table 4: Few optional features of Tensilica ConnX B10 and ConnX B20 DSPs

These units support flexible precision real and complex multiply-add, bit manipulation, data shift and normalization, data select, shuffle, and interleave. ConnX B10 and B20 DSPs have a configurable instruction set with predefined, pre-verified vector packages, including FIR, FFTs, single precision vector floating point unit (SP VFP), half-precision vector floating point unit (HP VFP), double precision vector floating point unit (DP VFP), and Scatter Gather. ConnX DSPs also include Xtensa 32-bit scalar ISA, ideal for efficient execution of control code. These features make the ConnX B10 and B20 DSPs ideal for building systems that combine high computational throughput with complex decision-making. For more details, please refer to ConnX B10 DSP User's Guide [4] and ConnX B20 DSP User's Guide [5].

Summary

This paper introduces various RSP modules provided by the Xtensa Radar Library. Module-level unified APIs with configurable options allow customers to quickly experiment, prototype, design, and develop radar systems for various applications. These modules are optimized for good performance on Tensilica DSPs.

Additional Information

For additional information on the unique abilities and features of Cadence Tensilica processors, refer to http://ip.cadence.com/ipportfolio/tensilica-ip

References

1. Xtensa Radar Library User Guide
2. Xtensa Radar Library B10 Performance
3. Xtensa Radar Library B20 Performance
4. ConnX B10 DSP User's Guide
5. ConnX B20 DSP User's Guide
6. Johnson et. al., A Methodology for Designing, Modifying, and Implementing Fourier Transform Algorithms on Various Architectures, Center for Large Scale Computation, NY 10036, Sept. 1989
7. García, Óscar Faus. Signal Processing for mmWave MIMO Radar. Diss. University of Gävle, 2015
8. Baruzzi, Aurora, and Marco Martorella. "Multi-sensor multi-target tracking based on range-doppler measurement." International Journal of Microwave and Wireless Technologies 8.3 (2016): 615-622
9. Hwang, H. K., et al. "Direction of arrival estimation using a root-MUSIC algorithm." Proceedings of the International MultiConference of Engineers and Computer Scientists. Vol. 2. Citeseer, 2008
10. Fixed-Point and Floating-Point FMCW Radar Signal Processing with Tensilica DSPs

cadence®

Cadence is a pivotal leader in electronic systems design and computational expertise, using its Intelligent System Design strategy to turn design concepts into reality. Cadence customers are the world's most creative and innovative companies, delivering extraordinary electronic products from chips to boards to complete systems for the most dynamic applications. **www.cadence.com**

 J14058 03/23 AA/KZ/PDF