Application Note

AWR2188 Waveform Configuration Explanation

Andy Yin *Central mmWave FAE*

ABSTRACT

This document was translated from a simplified Chinese source. (ZHCAGA4)

The AWR2188, TI's latest-generation high-performance eight-transmit, eight-receive monolithic microwave integrated circuit (MMIC), differs from the previous-generation MMIC, AWR2243, in its waveform configuration. This article describes the differences in RF and waveform configurations between the AWR2188 and AWR2243 and provides a detailed introduction to the AWR2188's waveform configuration, enabling customers to quickly configure and generate corresponding waveforms.

Table of Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

In response to the market demand for high-performance imaging radars, Texas Instruments (TI) introduced the next-generation high-performance mmWave radar, AWR2188. As shown in Figure 1-1, the AWR2188 is a monolithic microwave integrated circuit (MMIC) built on the RF-CMOS process and integrates the following subsystems:

- RF Analog Subsystem: RF and analog-related circuits consisting of Synthesizer, Power Amplifier (PA), Low Noise Amplifier (LNA), Mixer, Analog-to-Digital Converter (ADC), Voltage-Controlled Oscillator (VCO), Temperature Sensor, and so forth. This single-chip device allows eight transmit channels and eight receive channels to operate simultaneously.
- Radar Front-End Controller: The firmware of the RF controller runs on the built-in Cortex R4F, used to control the parameter configuration, calibration, and monitoring of the RF front end, and perform decimation filtering on the sampled signals.
- Host Interface Control Module: The built-in Cortex M4F runs the firmware provided by TI, used to configure the SPI or I2C interfaces for communication with external processors, and to configure the CSI interface for the transfer of ADC data.

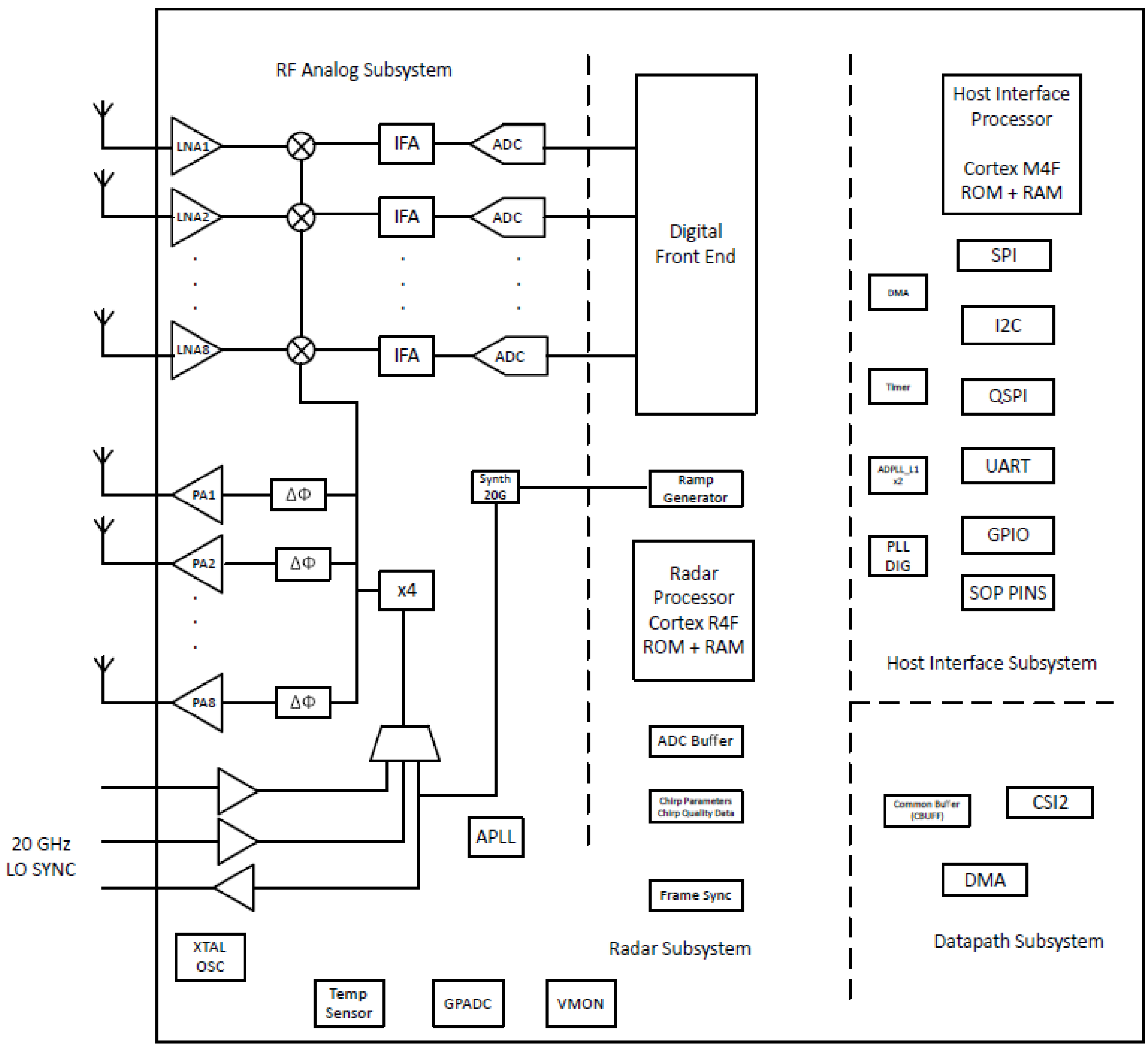

Figure 1-1. AWR2188 Chip Block Diagram

The AWR2188's RF performance also improves over the previous generation MMIC, AWR2243, as shown in Table 1-1.

Table 1-1. AWR2188 and AWR2243 Parameter Comparison

Parameter	AWR2243	AWR2188
Number of Transmit & Receive Antennas	3T4R	8T8R
Typical Transmit Power	13dBm	13.5dBm
Typical Receive Noise Figure	13db	10db
Phase Shifter Bit Width	6bit	8bit
Phase Shifter Accuracy	5°	3°
IF Bandwidth	20MHz	30MHz
ADC Sampling Rate	45Msps	66.67Msps
ADC Resolution	12bit	16bit
CSI2 Rate	2.4Gbps	6.8Gbps

As shown in Table 1-1, the AWR2188 single-chip device supports 8T8R and achieves improvements in indicators such as transmit power, noise figure, and sampling rate, significantly enhancing detection performance. With its physical antenna count increased and RF performance improved over the AWR2243 2-chip cascade, this single-chip device simplifies imaging radar system hardware design, reduces costs, and improves detection capabilities. The launch of the AWR2188 implementation will therefore bring a technological leap to the imaging mmWave radar market and accelerate market penetration for imaging radars.

To meet system design demands of different customers and provide greater flexibility, the AWR2188 is differentiated from the AWR2243 in its configuration for frequency-modulated continuous wave (FMCW), which is set to create new challenges for customers' software development. This article provides a detailed description of the FMCW waveform configuration for the AWR2188 to accelerate customers' software development related to waveform configuration.

2 FMCW Waveform Overview

Frequency-Modulated Continuous Wave (FMCW) is a type of continuous wave signal whose transmit frequency varies linearly over time. The target distance is calculated using the frequency difference between the echo and the transmitted signal, and the speed can further be calculated by leveraging the Doppler effect. This technology has relatively simple hardware processing, low cost, high-range resolution, strong immunity to interference, small size, and low power consumption. FMCW technology is widely used in automotive radar and advanced driver assistance systems (ADAS) to enable features such as blind-spot detection and adaptive cruise control. The smallest waveform unit of FMCW is a chirp. As shown in Figure 2-1, a typical chirp is described by parameters including Idle time, TX start time, ADC start time, ADC sampling rate, Freq slope, and so forth.

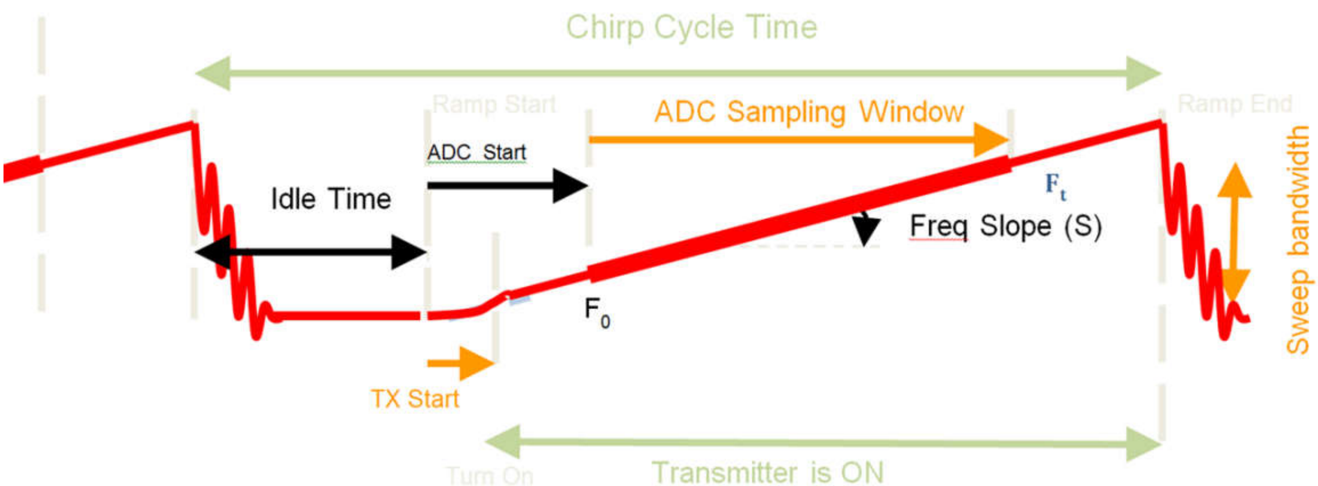

Figure 2-1. Typical FMCW Chirp

Figure 2-2 below illustrates the structure of a typical FMCW frame, which consists of multiple chirps and inter-frame idle time.

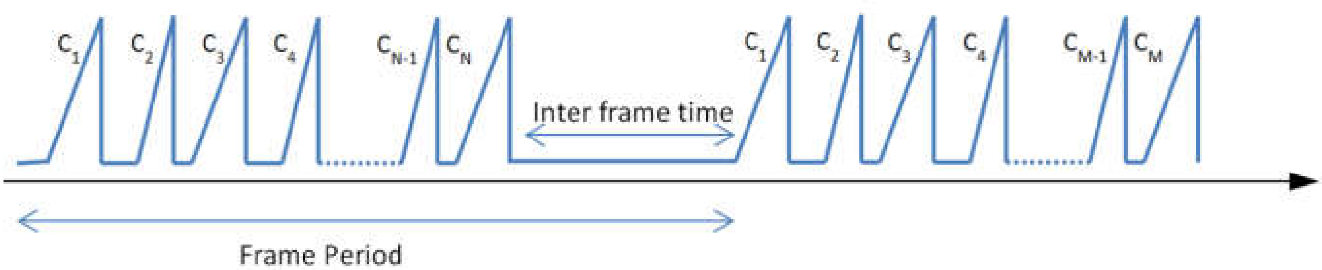

Figure 2-2. Typical FMCW Frame Structure

To support several detection modes within a frame, the advanced frame structure is introduced. Figure 2-3 below shows the structure of an advanced FMCW frame. Compared with the typical FMCW frame structure in Figure 2-2, the concepts of subframe and burst are introduced. An advanced frame consists of multiple subframes; a subframe consists of multiple bursts; and a burst consists of a series of chirps shown in Figure 2-2.

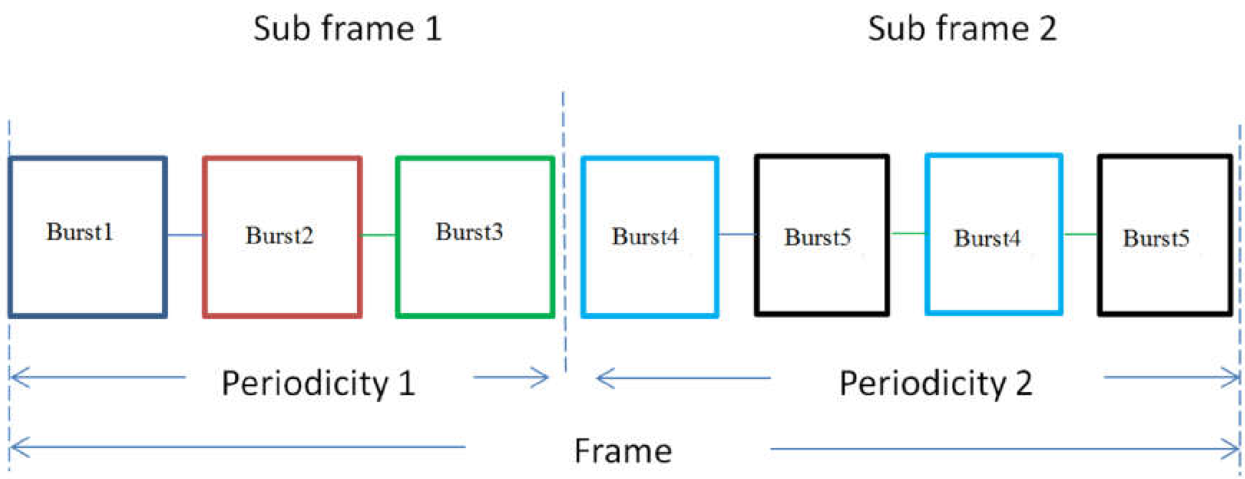

Figure 2-3. Advanced FMCW Frame Structure

3 AWR2188 FMCW Waveform Configuration

The configuration of TI's mmWave chip FMCW waveform involves profile, chirp, subframe, and frame. The profile configuration defines basic waveform parameters, including the start frequency, idle time, frequency slope, ADC sampling rate, sampling number, and so forth. Based on the profile configuration, the chirp configuration allows for parameter adjustment, resulting in different chirps. The subframe configuration defines the burst within a subframe and the chirp within a burst, including the burst idle time and subframe idle time. The frame configuration is associated with subframes, including the frame idle time. The waveform configuration for the AWR2188 is different from that for the AWR2243 and is relatively simpler, with the same process used for the configuration of typical and advanced frames. The AWR2188 versus AWR2243 waveform configurations are listed in Table 3-1 below.

Table 3-1. AWR2188 vs. AWR2243 Waveform Configurations

	AWR2243	**AWR2188**
Profile	Stored in the profile RAM; up to four profiles supported	Stored in the profile RAM; up to eight profiles supported. No difference in the configuration mode
Chirp	Two separate configuration modes are supported, either of which can be selected as needed. They differ in the configuration process. 1. Typical chirp: Stored in the chirp RAM; up to 512 chirps supported 2. Advanced chirp: Stored in the LUT RAM, adopting a look-up table approach for greater flexibility; up to 2048 chirps supported	Only one chirp LUT mode is available, and chirp-related parameter variations form a LUT, stored in the chirp RAM, which is obtained through the look-up table approach and forms a series of chirps. To configure any waveform, the chirp LUT must be configured.
Subframe	Advanced frame configuration must be invoked; up to four subframes are supported	Similar to the chirp configuration, the subframe configuration is stored in the chirp RAM. The configuration applies to both typical and advanced frames. Up to eight subframes are supported.
Frame	Available for typical and advanced frames, but calls to different APIs required	The single configuration applies to both typical and advanced frames.

As shown in Table 3-1, compared to the AWR2243, the waveform configuration of the AWR2188 is more flexible and unified, applying to both typical and advanced frames. The configuration of the AWR2188 is introduced in detail below.

3.1 Configuration Associations

Waveform configuration for the AWR2188 includes the following steps:

- Profile configuration: Includes basic parameter configuration. Each chirp must have a specific profile, which is stored in the specific profile RAM.
- Chirp parameter LUTs: Tunable chirp parameters are defined as a look-up table (LUT), which is stored in the specific chirp RAM.
- Subframe configuration: Includes chirp number, burst, and timing information, and chirp LUT configuration associated with the subframe, including the parameters that must use the chirp LUT, and the address, size, and look-up table approach of each parameter in the chirp LUT. Like chirp parameter LUTs, the subframe configuration is also stored in the chirp RAM.
- Frame configuration: Includes parameters such as the start address in the chirp RAM, subframe number, and so forth for subframes to be used in the frame.

Figure 3-1 below shows the storage locations and associations of each configuration in the RAM.

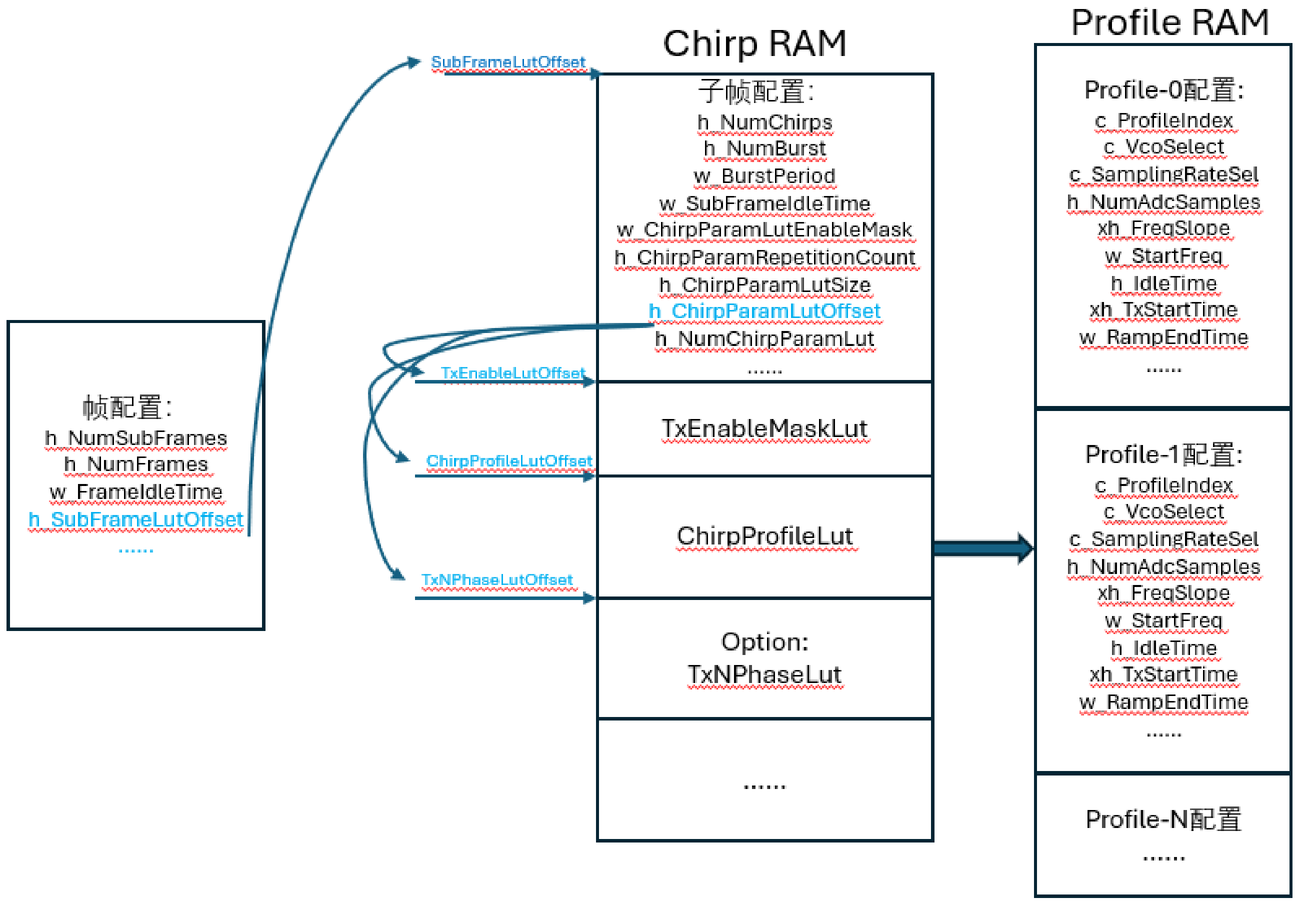

Figure 3-1. Waveform Configuration Storage Locations and Associations

As shown in Figure 3-1, for the waveform generation of each frame, first locate the frame configuration; then locate the corresponding subframe configuration in the chirp RAM based on the h_SubFrameLutOffset setting in the frame configuration; and finally index the profile and TxEnable configurations in the chirp RAM according to the defined chirp LUT configuration to generate the corresponding chirps.

For the AWR2188 waveform configuration, the basic API configuration flow is shown in Figure 3-2 below. The chirp parameter LUTs provide great flexibility for the generation of different waveforms.

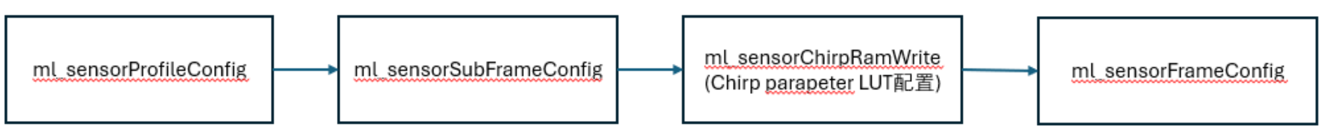

Figure 3-2. API Configuration Flow

The following is a detailed description of frame, subframe, chirp parameter LUT, and profile. See also Reference[5] for detailed configuration parameter structures and APIs.

3.2 Frame Configuration

In addition to the subframe number, transmitted frame number, and frame time, the frame configuration also includes a parameter, h_SubFrameLutOffset, making the device different from the AWR2243. This parameter corresponds to the address offset of the subframe configuration in the chirp RAM. It allows the user to locate the corresponding subframe configuration in the chirp RAM. If a frame contains multiple different subframes, the configurations of these subframes are stored consecutively starting from the address specified by h_SubFrameLutOffset. The API, ml_sensorFrameConfig, can be called for frame configuration.

Figure 3-3 below shows an example of frame configuration. As the figure shows, the frame configuration includes the subframe number, transmitted frame number, frame idle time, address offset of the subframe configuration in the chirp RAM, and address of the calibration and monitoring related parameter table in the chirp RAM.

```
/*Frame Config*/
T_ML_SENSOR_FRAME framecfg =
{
    .h_NumSubFrames = 1,
    .h_NumFrames = 0,
    .w_FrameIdleTime = 100000U * LINK_FRC_1US,
    .h_SubFrameLutOffset = 0x200,
    .h_CalMonScheduleLutOffset = 0x1000,
    .h_CalMonScheduleLutLength = 1,
};
/*sensorFrameConfig*/
status = ml_sensorFrameConfig(DEVICE_INDEX,&framecfg);
```

Figure 3-3. Frame Configuration Routine

3.3 Subframe Configuration

Subframe configuration is described here specifically in relation to chirp parameter LUTs, and other parameters are described in the reference manual[5]. Chirp-related parameters are obtained through the look-up table approach. For descriptions of chirp parameters that support the look-up table approach, see also the reference manual[5]. w_ChirpParamLutEnableMask determines which chirp parameters are obtained through the look-up table approach, where the corresponding bits for profile and TxEnable must be enabled. The table storage address for parameters supporting the look-up table approach is defined by h_ChirpParamLutOffset. The specific query and use methods of tables are defined by h_ChirpParamRepetitionCount, h_ChirpParamLutSize, and h_NumChirpParamLut. Table 3-2 below describes the chirp parameter LUT configuration in a subframe.

Table 3-2. Description of Chirp Parameter LUT Configuration in Subframe

w_ChirpParamLutEnableMask	Each bit corresponds to a chirp parameter, and a total of 17 parameters are supported, among which profile and TxEnable must be enabled
h_ChirpParamLutOffset	The storage start address offset of chirp parameter LUTs in the chirp RAM
h_ChirpParamRepetitionCount	The number of times each parameter in a Chirp LUT is used each time it is read
h_ChirpParamLutSize	The size of the chirp LUT in each burst
h_NumChirpParamLut	Defines the burst number for the reset of the chirp LUT. In other words, parameters in a chirp LUT are re-indexed from the start address after a specified number of bursts

Figure 3-4 shows the usage routine for the chirp parameter LUT configuration in a subframe. A set of configurations [A B C D E E] is written into the chirp LUT. h_ChirpParamRepetitionCount=2 indicates that after retrieval, each parameter is used consecutively two times. In other words, each parameter applies to two consecutive chirps. h_ChirpParamLutSize=3 indicates that three parameters in the LUT are used in each burst. h_NumChirpParamLut=2 indicates that the indexing loops back to the first parameter in the LUT after every two bursts are transmitted when using the parameters in the LUT for a subframe.

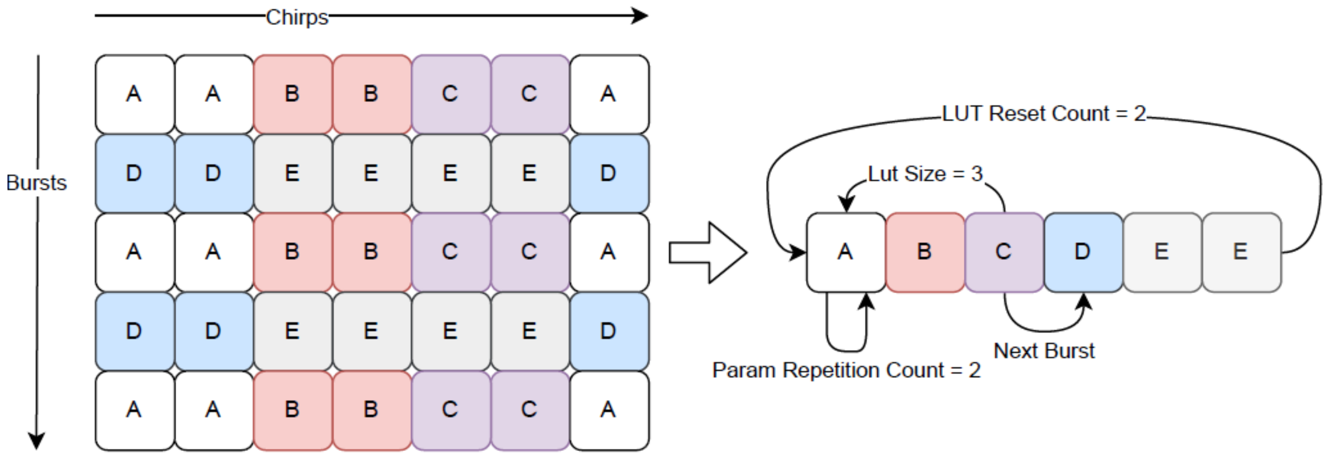

Figure 3-4. Usage Routine for Chirp Parameter LUT Configuration in Subframe

For subframe configuration, call ml_sensorSubFrameConfig to write the configuration of a single subframe into the chirp RAM, or call it multiple times for multiple subframes.

To configure multiple subframes, call ml_sensorChirpRamWrite to write multiple subframe configurations into the chirp RAM at a time, making the configuration more efficient.

Figure 3-5 shows the reference code for subframe configuration, which defines the burst number, chirp number per burst, burst period, subframe idle time, and chirp parameter LUT configuration.

In the example shown in Figure 3-5, the profile and TxEnable configurations are enabled. The profile configuration is stored in the chirp RAM at address offset 0x0, and the TxEnable configuration is stored in the chirp RAM starting from address offset 0x100. Only one entry is stored for each in the chirp RAM, and each configuration applies to all chirps. In the code, ml_sensorSubFrameConfig is called to write the subframe configuration into the chirp RAM at the address offset 0x200. Note that this address offset must match the h_SubFrameLutOffset setting in the frame configuration described in Section 3.2.

```
/*Subframe Config*/
T_ML_SENSOR_SUB_FRAME subframecfg =
{
    .c_SubFrameCpIndex = 0,
    .h_NumChirps = 256,
    .h_NumBurst = 1,
    .w_BurstPeriod =  ((70U * 256U) + 5000U) * LINK_FRC_1US, // (rampTime + idleTime) * numChirp + burstIdle
    .w_SubFrameIdleTime = 10000U * LINK_FRC_1US,
    .w_ChirpParamLutEnableMask = M_ML_SENSOR_CHIRP_LUT_MASK_PROFILE | M_ML_SENSOR_CHIRP_LUT_MASK_TX_ENABLE_MASK,
    .h_ChirpParamRepetitionCount =  {
            [M_ML_SENSOR_CHIRP_LUT_INDEX_PROFILE]        = (UINT8)1U,
            [M_ML_SENSOR_CHIRP_LUT_INDEX_TX_ENABLE_MASK] = (UINT8)1U,
    },
    .h_ChirpParamLutSize = {
            [M_ML_SENSOR_CHIRP_LUT_INDEX_PROFILE]        = (UINT8)1U,
            [M_ML_SENSOR_CHIRP_LUT_INDEX_TX_ENABLE_MASK] = (UINT8)1U,
    },
    .h_ChirpParamLutOffset = {
            [M_ML_SENSOR_CHIRP_LUT_INDEX_PROFILE]        = (UINT16)0x0,
            [M_ML_SENSOR_CHIRP_LUT_INDEX_TX_ENABLE_MASK] = (UINT16)0x100,
    },
    .h_NumChirpParamLut = {
            [M_ML_SENSOR_CHIRP_LUT_INDEX_PROFILE]        = (UINT8)1U,
            [M_ML_SENSOR_CHIRP_LUT_INDEX_TX_ENABLE_MASK] = (UINT8)1U,
    },
    .h_MiscControlMask = 0,
    .h_InterChirpPowerSaveCtrl = 0,
};
/*SensorsubFrameConfig*/
status = ml_sensorSubFrameConfig(DEVICE_INDEX, 0,0x200,&subframecfg);
```

Figure 3-5. Subframe Configuration Reference Code

3.4 Chirp Configuration

The AWR2188 supports variable parameters for chirps. These chirp parameters are stored in the dedicated chirp RAM in the form of a look-up table (LUT). See also Reference[5] for detailed chirp parameters. Among these chirp parameters, the profile LUT and TxEn LUT must be enabled, which respectively specify the profile for the chirp and the transmit antenna for the chirp.

Each chirp parameter LUT has its own format and unit. See also Reference[5] for more information.
A chirp parameter LUT can be written into the chirp RAM at the specified address offset using ml_sensorChirpRamWrite. Note that the total available space of the chirp RAM is 31KB. It stores chirp parameter LUTs and subframe configuration.

As shown in Figure 3-6, the profile and TxEnable LUTs are defined and written into the chirp RAM at offset addresses 0 and 0x100, respectively, by calling ml_sensorChirpRamWrite. Note that the write address offsets here must match the h_ChirpParamLutOffset setting in the subframe configuration described in Section 3.3.

```
uint8_t lutparam_profile[4] = {0};
uint8_t lutparam_Txenable[8] = {M_ML_SENSOR_MASK_ALL_TX};

/*LUT write for Profile parameter Mandatory*/
status = ml_sensorChirpRamWrite(DEVICE_INDEX,0x0,4,lutparam_profile);
while(status!=0);

/*LUT write for TX Enable Mask parameter Mandatory*/
status = ml_sensorChirpRamWrite(DEVICE_INDEX,0x100,4,lutparam_Txenable);
while(status!=0);
```

Figure 3-6. image10_chirp Parameter LUT Configuration Reference Code

3.5 Profile Configuration

The profile configuration of this device is no different from that of the AWR2243, which defines basic RF parameters, including sampling rate, sampling number, receive gain, transmit power back-off, frequency slope, and so forth. See also Reference[5] for more details. Profile configuration can be performed using ml_sensorProfileConfig.

Figure 3-7 shows the reference code for profile parameters and configuration, which defines one profile. To define and configure multiple profiles, add a set of configurations in profilecfg and call ml_sensorProfileConfig again to configure the profile.

```c
/*Profile Config*/
T_ML_SENSOR_PROFILE profilecfg[MAX_NUM_OF_PROFILES] =
{
    {    /*Profile 0*/
        .c_ProfileIndex = 0,
        .c_VcoSelect = M_ML_SENSOR_VCO_SELECT_AUTO,
        .h_MiscSettings = M_ML_SENSOR_HPF_FAST_INIT_ENABLE |  M_ML_SENSOR_CAL_MON_OVERRIDE_ENABLE,
        .c_SamplingRateSel = 16,
        .c_RxGainSelect = M_ML_SENSOR_RF_GAIN_SEL_38dB | (UINT8)44U,
        .h_NumAdcSamples = 512,
        .c_RxRfInitCalDataSel = 0,
        .c_HpfCutoffSel = {[0U] = M_ML_SENSOR_HPF_CUTOFF_FREQ_350KHz, [1U] = M_ML_SENSOR_HPF_CUTOFF_FREQ_350KHz},
        .c_TxBackoffs = { [0U] = 0U,[1U] = 0U,[2U] = 0U,[3U] = 0U,[4U] = 0U,[5U] = 0U,[6U] = 0U,[7U] = 0U},
        .c_TxClpcCalEnableMasks = { [0U] = 0xFFU,[1U] = 1U,[2U] = 1U,[3U] = 1U,[4U] = 1U,[5U] = 1U, [6U] = 1U, [7U] = 1U},
        .c_TxRfInitCalDataSel = {[0U] = 0U,[1U] = 0U,[2U] = 0U,[3U] = 0U,[4U] = 0U,[5U] = 0U,[6U] = 0U,[7U] = 0U},
        .xc_TxPhase = {[0]=0,[1]=0,[2]=0,[3]=0,[4]=0,[5]=0,[6]=0,[7]=0},
        .xw_StartFreqAccDelta = 0,
        .xc_TxPhaseAccDelta = {[0]=0, [1]=0, [2]=0, [3]=0, [4]=0, [5]=0, [6]=0, [7]=0},
        .h_StartFreqAccPeriod = 0,
        .h_TxPhaseAccPeriod = {[0]=0, [1]=0, [2]=0, [3]=0, [4]=0, [5]=0, [6]=0, [7]=0},
        .w_StartFreq = LINK_CONST_GET_SYNTH_FREQ(76),
        .xh_FreqSlope = LINK_CONST_GET_CHIRP_SLOPE(15.998),
        .xh_TxStartTime = 0 * LINK_RAMPGEN_1US,
        .w_RampEndTime = 40U * LINK_RAMPGEN_1US,
        .h_IdleTime = 30U * LINK_RAMPGEN_1US,
        .h_AdcStartTime = 6U * LINK_RAMPGEN_1US,
        .h_AdcSkipSamples = 0,
        .c_LoopbackFreqSelA = (UINT8)30U,
        .c_LoopbackFreqSelB= (UINT8)6U,
        .c_LoopbackGainA = (UINT8)63U,
        .c_LoopbackGainB = (UINT8)0U,
        .c_LoopbackMode = M_ML_SENSOR_LOOPBACK_DISABLE,
        .w_CalMonStartFreq = LINK_CONST_GET_SYNTH_FREQ(78),
        .xh_CalMonFreqSlope = LINK_CONST_GET_CHIRP_SLOPE(0.1),
        //.h_ProgFilterLength = 4U,
        //.h_ProgFilterStartIndex = 0,
    },
    /* Other Profiles can be added if required*/
};
/*sensorProfileConfig*/
status = ml_sensorProfileConfig(DEVICE_INDEX,&profilecfg[0]);
```

Figure 3-7. Profile Configuration Reference Code

4 Summary

This article compares the waveform configuration differences between the AWR2243 and AWR2188, details the AWR2188 waveform configuration process, and provides corresponding reference examples to help customers better understand and master the AWR2188 waveform configuration methods, enabling them to more quickly generate the waveform configurations required for their products.

5 References

1. *SWRA553A, Programming Chirp Parameters in TI Radar Devices*
2. *SWRS223D, AWR2243 Single-Chip 76- to 81-GHz FMCW Transceiver datasheet*
3. *SWRS339, AWR2188 Single Chip 8x8 Cascadable 76-to-81 GHz Transceiver in LOP package datasheet (Rev. A)*
4. *mmWave-Radar-Interface-Control.pdf (mmwave_mcuplus_sdk_04_06_01_02\mmwave_dfp_02_02_04_00\docs)*
5. *mmWaveLink_4_Interface_Control_Document_1_0.pdf(mmWave_DFP_04_00_06_02\docs)*