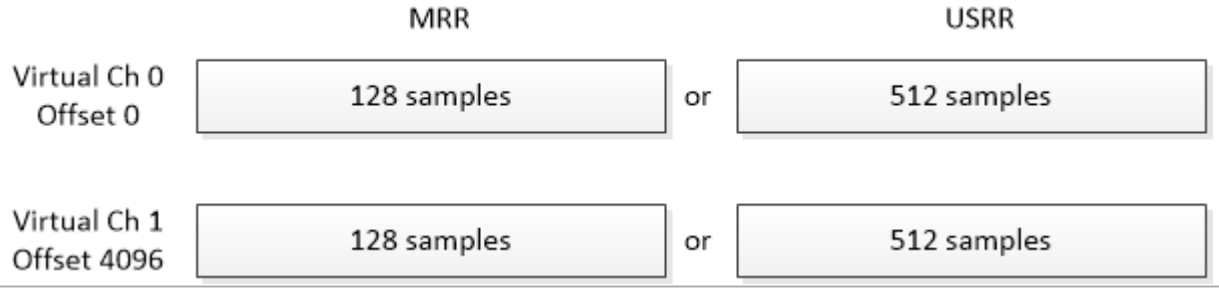
MRR Beamsteering HWA Demo Implementation Notes

Note

- Note:
  - The demo released in Radar Toolbox supports only MRR mode
    - \radar_toolbox_2_20_00_05\source\ti\examples\ADAS\mrr_beamsteering
  - There were some issues with the USRR mode but supporting this mode was not a mktg requirement and the team did not have sufficient time to debug it.

ADCBuf format/usage

1. First, ADCBuf must be configured to set “sampleInterleave” to I(MSB), Q(LSB) format for HWA usage. Otherwise, range bins will appear reversed in the range FFT output.
2. Second, this demo uses a fixed offset in the ADCBuf for each Rx antenna’s data. This is because the USRR data is not the same size as MRR and we don’t want to reconfigure the ADCBuf each sub-frame.
   - In MRR_DSS_dssDataPathConfigAdcBuf(), we set the channel offset = (16 * 1024) / 4.
   - In HWA_configRangeFFT(), we set .source.srcBidx = (16 * 1024) / 4.
3. Third, we use the shared mode where there is a single input buffer used for HWA, so that we do not have to EDMA copy samples into the HWA buffers.
4. The demos uses “single chirp” mode, meaning only 4 virtual antenna’s data are in the buffer at any point.
5. Format for each Rx antenna:



ADCBuf format/usage (cont.)

- **MRR_DSS_dssDataPathConfigAdcBuf();**
  - Notice in this function the hard-coded ADCBuf config
    - ADCBuf_dataFormat dataFormat;

    - dataFormat.adcOutFormat = 0; // Complex mode
    - dataFormat.channelInterleave = 1; // No Interleaving
    - dataFormat.sampleInterleave = 1; // Sample Interleave must be I(MSB) Q(LSB) for HWA usage

    - ADCBuf_RxChanConf rxChanConf;

```
/* Divide 16KB of the ADC buffer into 4 parts. Note that in the ADCBuf mode
* we are using, there is a single input buffer to the HWA for all chirps. */
rxChanConf.offset = ((16 * 1024) / 4);
chirpThreshold   = 1;
```

Advanced Frame Configuration

This demo runs three different chirp patterns, arranged into two sub-frames:

- Subframe 0: MRR, 3 Tx * 4 Rx, SIMO
  - 128 “Fast” Chirps, 256 samples, followed by
  - 128 “Slow” Chirps, 256 samples

- Subframe 1: USRR, 3 Tx * 4 Tx, MIMO
  - 32 Chirps per Tx, 512 samples

- There is a frame interrupt for each sub-frame, followed by the chirp interrupts, followed by the inter-frame period for frame processing of each sub-frame.

- **Note: In this release the USRR mode has not been validated. It should be used only in MRR mode.**

Radar Cube format

The radar cube was changed to IWR1443 interleaved format for HWA usage (see Appendix A). This is DPIF Format 2, with one exception: for MRR, the slow chirp data follows the fast chirp data for each virtual antenna:

Radar Cube format

For MRR, the cube is organized as follows:

- 256 Range Bins, each consisting of
  - 4 virtual antennas, having
    - » 128 fast chirp Doppler bins followed by
    - » 128 slow chirp Doppler bins each.
- (so Doppler bin is the fastest incrementing dimension; Range bin is the slowest)
- Total cube size = 256 * 4 * (128+128) * 4 = 1MB

For USRR, the cube is organized as follows:

- 512 Range Bins, each consisting of
  - 12 virtual antennas, having
    - » 32 Doppler bins each
- (so Doppler bin is the fastest incrementing dimension; Range bin is the slowest)
- Total cube size = 512 * 12 * 32* 4 = 768KB

HWA/EDMA 1D FFT Config (MRR and USRR)

- HWA source (see HWAutil_configRangeFFT)
  - A count = 255 or 511 (samples – 1)
  - A index = 4 (sample size)
  - B count = 3 or 11 (number virtual channels – 1)
  - B index = (16 * 1024) / 4 (channel offset)
- HWA dest
  - A count = 255 or 511 (range bins – 1)
  - A index = (4 or 12) * 4 (number virtual channels * word size)
  - B index = 4 (word size)
- EDMA source – not needed
- EDMA dest (see EDMAutil_configHwaTranspose)
  - SyncAB type, transposing
    - A count = (4 or 12) * 4 (number of virt channels * word size)
    - B count = 256 or 512 (number of range bins)
    - C count = (256 or 32)/2 (number of chirps / 2 (ping pong))
    - Src B index = A count
    - Dest B index = A count * (128 or 32) (A count * number of chirps (Doppler bins))
    - Src C index = 0
    - Dest C index = A count * 2 (2 accounts for ping and pong)

HWA/EDMA 2D FFT Config (MRR and USRR)

- HWA source (see HWAutil_configDopplerFFT)
  - A count = 127 or 31 (Doppler bins – 1)
  - A index = (4 or 12) * 4 number virtual channels * sample size)
  - B count = 3 or 11 (number virtual channels – 1)
  - B index = 4 (word size)
- HWA dest
  - A count = 127 or 31 (Doppler bins – 1)
  - A index = (4 or 12) * 4 (number virtual channels * word size)
  - B index = 4 (word size)
- EDMA source (see Config_2d_and_Azm_Edma)
  - SyncAB type
    - A count = 4 (word size)
    - B count = 128 or 32 (number of Doppler bins)
    - C count = 4 or 12 (number of virtual antennas)
    - Src B index = (4 or 12) * 4 (number of virtual antennas * word size)
    - Dest B index = 4 (word size)
    - Src C index = 4 (word size)
    - Dest C index = (128 or 32) * 4 (number of Doppler bins * word size)
- EDMA dest – not needed

1D FFT HWA and EDMA Paramset Usage

1D FFT		MRR Usage							
		HWA			EDMA				EDMA + HWA 1D sequence
	trigger	paramset	type		ch/param	type	ch/param	type	
		1	dummy	<-- ch 1	7	ping 1hot	66	link	7, 10 are 1hot to trigger du
	SW ch1 ->	2	FFT	-->	17	ping data	64	link	SW starts HWA channels 1
		3	dummy	<-- ch 2	10	pong 1hot	67	link	
	SW ch2 ->	4	FFT	-->	18	pong data	65	link	
		USRR Usage							
		HWA			EDMA				
	trigger	paramset	type		ch/param	type	ch/param	type	
		5	dummy	<-- ch 3	11	ping 1hot	70	link	
	SW ch3 ->	6	FFT	-->	19	ping data	68	link	

2D FFT HWA and EDMA Paramset Usage

2D FFT	MRR Usage													
		HWA			EDMA					HWA parms					
	trigger	paramset	type		ch/param	type	ch/param	type		paramset	trig mode	fft en	win en	log2 en	abs en
	SW ch5 ->	9	FFT rb1	<-- ch 5	35	ping 1hot	75	link		9	3	1	1	1	1
		10	FFT rb2		34	ping in	74	link		10	0	1	1	1	1
		11	log2mag	-->						11	0	0	0	0	0
		12	sum		21	ping out	72	link		12	0	1	0	0	0
	SW ch6 ->	13	FFT rb1	<-- ch 6	38	pong 1hot	79	link		13	3	1	1	1	1
		14	FFT rb2		37	pong in	78	link		14	0	1	1	1	1
		15	log2mag							15	0	0	0	0	0
		16	sum	-->	22	pong out	76	link		16	0	1	0	0	0

	USRR Usage						
	HWA			EDMA			
trigger	paramset	type		ch/param	type	ch/param	type
SW ch5 ->	9	FFT rb1	<-- ch 5		ping 1hot		link
	10	FFT rb2			ping in		link
	11	log2mag	-->				
	12	sum			ping out		link

EDMA + HWA 2D sequence:

SW starts channels 34 (ping), 37 (pong)

34, 37 copy from L3 to HWA input

34, 37 chain to 35, 38

35, 38 are 1hot to trigger HWA channels 5, 6

HWA completion triggers 21, 22 (HWACC_4, 5)

Appendix A – AWR1443 OOB Demo 1D Radar Cube Format

- The AWR1443 OOB Demo Radar Cube format is documented in mmWave SDK 2.1
- C:/ti/mmwave_sdk_02_01_00_04/packages/ti/demo/xwr14xx/mmw/docs/doxygen/html/index.html
- AWR1443 BOOST EVM antenna pattern is identical to AWR1843 BOOST EVM antenna pattern

Appendix A – AWR1443 OOB Demo
1D Radar Cube Format (cont.)