Application Note

xWRL6432 Low Power Radar - Power Optimization Techniques

Nathan Block, Kevin Ortiz

Table of Contents

1 Power Management Framework 2
2 Hardware Design Options for Low Power 5
3 Chirp Design Optimizations for Low Power 6
4 Reducing Power in the Interframe Idle and Deep Sleep States 15
5 Measuring Power 18
6 References 19
7 Revision History 19

List of Figures

Figure 1-1. xWRL6432 Device Architecture 2
Figure 1-2. Typical Application Flow during 1 frame of operation 3
Figure 3-1. Sequence of Events in a Chirp 9
Figure 3-2. IWRL6432BOOST Virtual Antenna Layout 13
Figure 4-1. Switchable and Unswitchable Memory 17

List of Tables

Table 1-1. Major Components in each Power Domain 2
Table 1-2. Application Flow State Definitions 3
Table 1-3. Power Domain Status in Different Power Modes 4
Table 4-1. xWRL6432 Main Memory map 16
Table 4-2. APPSS and Shared RAM Memory Bank 16

Trademarks

1 Power Management Framework

Texas Instruments' low power xWRL6432 radar sensors detect objects and motion by emitting and receiving Frequency Modulated Continuous Waves (FMCW). Designed for low-power applications such as video doorbells, security systems, gesture controlled HMI and automotive intruder detection, the xWRL6432 offers multiple low-power modes that allow the radar device to minimize power consumption depending on the use-case needs. This application note details the ways to reduce power consumption through effective system design, chirping profile, and application software. Additionally, this application note will cover techniques for measuring power on the xWRL6432 device.

Power Domains

The low power architecture of the xWRL6432 allows the radar device to completely or partially power off specific power domains. Figure 1-1 shows the block diagram of the xWRL6432 power domain architecture, and Table 1-1 details the specific sub-blocks within each power domain.

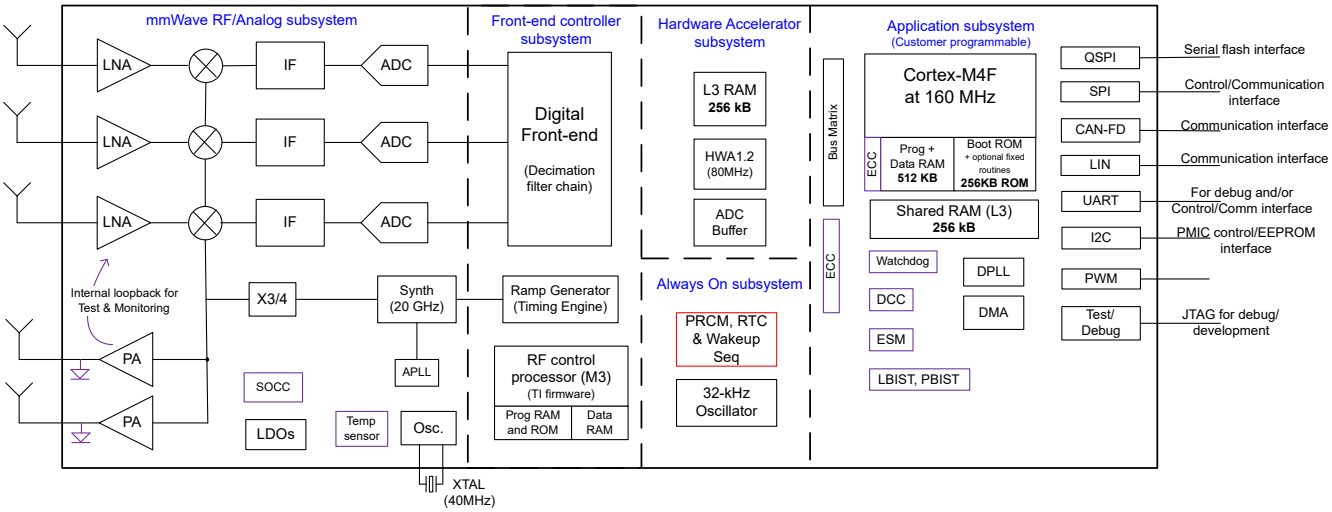


Figure 1-1. xWRL6432 Device Architecture

Table 1-1. Major Components in each Power Domain

Power Domain	Major Components
RF_ANA_PD (RF Analog Power Domain)	• Power Amplifiers (PA) • Low Noise Amplifiers (LNA) • Mixers • Intermediate Frequency Filters (IF) • Analog-to-Digital Converters (ADCs) • Synthesizer • 40 MHz Crystal Oscillator (OSC)
FEC_PD (Front-End Controller Power Domain)	• Cortex-M3 Processor (including memory) • Digital Front-End (DFE)
APPSS_PD (Application Subsystem Power Domain)	• Cortex-M4F Processor • Application Memory Banks • General Purpose Peripherals (Watchdog, UART, I2C, SPI, RS232...)
HWASS_PD (Hardware Accelerator Power Domain)	• Hardware Accelerator (HWA) • Memory for HWA
AON_PD (Always On Power Domain)	• Real-Time Clock (RTC) • Power, Reset, Clock Management (PRCM) Registers

Typical Radar Application Flow

Figure 1-2 shows a typical radar application flow. The terminology used in the flow is explained in Table 1-2.

Note

There are many variations on this flow possible that are beyond the scope of this document.

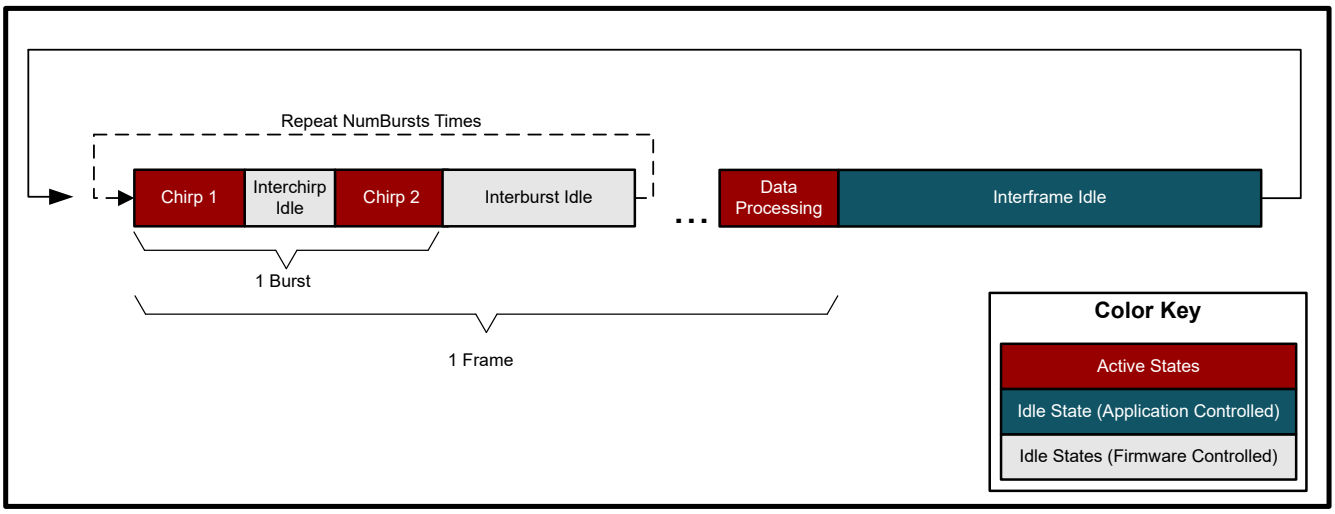


Figure 1-2. Typical Application Flow during 1 frame of operation

Table 1-2. Application Flow State Definitions

Chirp	**The period of time in which one or more transmitters emits/ receives an FMCW wave.**
Burst	A sequence of chirps. The xWRL6432 device has two transmit antennas, so a typical burst may transmit on one or both antennas.
Frame	A time period that consists of a sequence of bursts followed by data processing. Frames are periodic on a defined interval.
Interchirp Idle	The time period between chirps
Interburst Idle	The time period between bursts
Interframe Idle	The time period between the end of one frame and the beginning of the next frame
Data Processing	The time period in which the device processes the radar data collected in the previous frame

As the graphic above shows, the radar device chirps a number of times on regular intervals, processes all the data collected, then goes into a low-power Interframe Idle state. The next section describes each of the above states in greater detail.

Power State Descriptions

Active

The Active state is when the device is chirping or processing chirp data. In this state, the device can either be in a Data Acquisition substate, which is when data is being collected by transmitting and receiving chirps, or in a Data Processing substate, when the samples recorded in the Data Acquisition substate are being processed together. When the device is in the Data Acquisition sub-state, both the APPSS_PS and the FEC_PD must be on, however, when the device is in the Data Processing substate, the FEC_PD may be powered down to reduce power usage. **The Data Acquisition substate within the Active state draws the highest power level of the device.**

Idle

The Idle state occurs when the device is not actively chirping or processing data. There are three types of Idle states (Interchirp Idle, Interburst Idle and Interframe Idle). The Interchirp Idle and Interburst Idle states are completely handled by the device firmware. As the device cycles between chirps and bursts, it will go to these states automatically. By comparison, the Interframe Idle state that the device enters between frames may be configured and modified by the user. The time requirements for each idle state are as follows:

	Description	Minimum time	Relative power to other idle modes
Interchirp Idle	Low power mode in between two consecutive chirps in the same burst.	If low power modes are enabled : max(6 µsec – TX_START_TIME, 3.1 µsec) Else : Max(4 µsec – TX_START_TIME, 3.1 µsec)	Draws more power than Interburst Idle and Interframe Idle.
Interburst Idle	Low power mode in between two consecutive bursts in the same frame.	95 µsec	Draws less power than Interchirp Idle but more power than Interframe Idle.
Interframe Idle	Low power mode in between two frames.	135 µsec	Draws less power than Interchirp Idle and Interburst Idle.

Deep Sleep **:** *Optional state for Interframe-Idle time*

The Deep Sleep state is an application-driven option for the xWRL6432 when it is already in the Interframe Idle state. Deep Sleep is the **lowest possible power state designed state in the device**, where nearly all the device power domains, including the Application sub-system (APPSS), along with Hardware Accelerator (HWA) and Front-End Controller sub-system (FECSS) are powered off to save a significant amount of power. Deep sleep can be triggered by the application when the device enters the Interframe Idle state. Deep Sleep requires a nominal amount of time to deinitialize software and power down hardware, then power up hardware and reinitialize the software after the end of the deep sleep period. Typical times for this are around 2 milliseconds, but they will vary depending on the deep sleep options selected.

Even though the entire device being is almost completely powered down, it does not need to reboot after waking up from deep sleep. The contents of the device, such as the application image and chirp profile are retained across deep sleep cycles in the APPSS/FECSS memories.

A device may exit from the deep sleep state through the Sleep Counter, UART RX, SPI CS, GPIO, and RTC.

Power Domains in the Different Power Modes

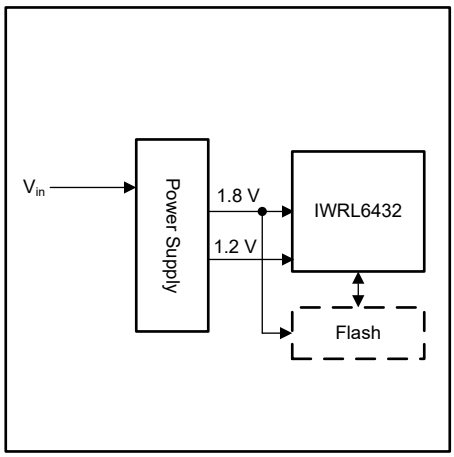
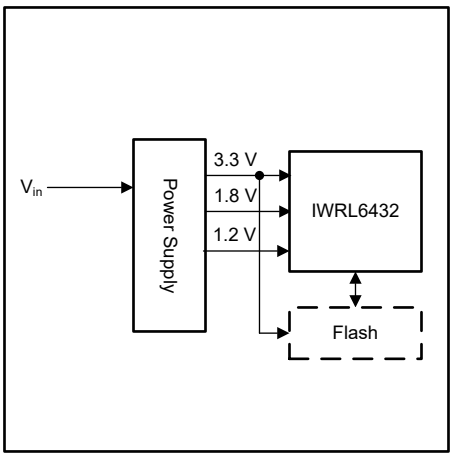
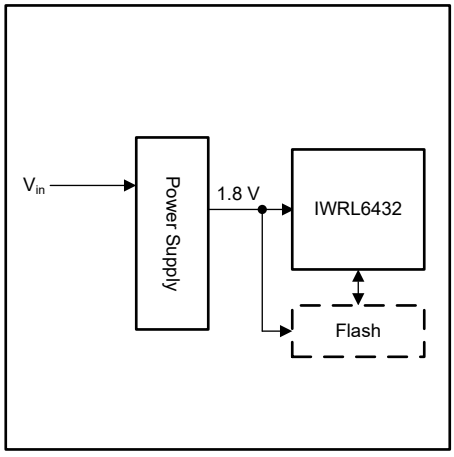
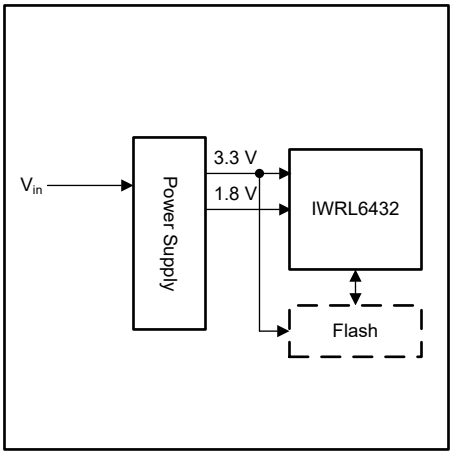
The table below shows the states of the different power domains in the different modes. Note that many of these may be modified, and different portions of the power domains may be clock-gated or powered down by the user.

Table 1-3. Power Domain Status in Different Power Modes

Power Domain	Active (Chirping)	Active (Data Processing)	Interchirp Idle	Interburst Idle	Interframe Idle	Deep Sleep
RF_ANA_PD	On	Off (Crystal Oscillator on)	On (PA, LNA off)	Off (Crystal Oscillator on)	Off (Crystal Oscillator on)	Off
FEC_PD	On	Off	On (Digital Front End clock-gated)	On (Digital Front End clock-gated)	On (Entire sub-system clock-gated)	Off
APPSS_PD	On (PLL off)	On (Running on Digital PLL)	On (PLL off)	On (PLL off)	On (Running on crystal clock)	Off
HWASS_PD	On (HWA clock-gated)	On	On	On (HWA clock-gated)	Off	Off
AON_PD	On	On	On	On	On	On

2 Hardware Design Options for Low Power

The xWRL6432 device is designed to be powered by a Power Management Integrated Circuit (PMIC) or discrete power supply solutions. Between 1-3 different supply voltage rails can be provided to the xWRL6432. This flexibility enables systems to be tailored to reduce power usage or BOM cost. When the device boots, it senses the number of input voltages provided, and adjusts its internal circuitry accordingly. The table below (with images taken from [1]) shows the four topologies that can be used to supply the power rails to the xWRL6432.

	1.8 V IO Mode	3.3 V IO Mode
Power Optimized	 Number of power supply rails needed : 2 **Lowest Power Topology**	 Number of power supply rails needed : 3
BOM Optimized	 Number of power supply rails needed : 1 **Lowest Cost Topology**	 Number of power supply rails needed : 2

The four topologies vary on two dimensions: IO voltage and the source of the 1.2 V power supply. The IOs on the device can run on either 3.3 V logic or 1.8 V logic depending on the application requirements. The 1.2 V power supply can be generated internally through a low dropout regulator (LDO) that generates it from a 1.8 V supply (BOM-Optimized Modes), or it can be provided to the device using an external voltage source (Power Optimized Modes).

3 Chirp Design Optimizations for Low Power

Active mode, and specifically the act of chirping during active mode, draws the most power of all the different states the xWRL6432 may enter. Chirping draws the most power because it turns on high power RF amplifiers for transmitting and receiving radar waves. Therefore, users may reduce the overall power consumption by reducing the amount of time spent chirping and/or by reducing the amount of power drawn while chirping. A non-exhaustive list of the most efficient ways to design a reduce power during chirping can be seen below.

Power Reduction Technique	Performance Impact
Increase the time between frames	Higher latency
Decrease the time spent in Interchirp Idle and Interburst Idle	Larger maximum velocity. Coarser velocity resolution
Decrease the chirping time	Reduced range, seen more noticeably at the edges of the Field of View
Reduce the number of transmitters / receivers	Reduced angular resolution. Reduced range, seen more noticeably at the edges of the Field of View
Reduce the transmit power	Reduced range, seen more noticeably at the edges of the Field of View

A number of these techniques involve modifying a configuration file, so a modified configuration file from MMWAVE-L-SDK version 05.01.00.04 is shown below and referenced for convenience. Different MMWAVE-L-SDK versions may name or order the parameters discussed differently, but the fundamental concepts should not change between versions of the SDK. Additionally, detailed explanations of each of the parameters can be found in [2], which is located in the docs/ folder of the MMWAVE-L-SDK.

Note

The mmWave Sensing Estimator Tool found online provides an intuitive interface for users to calculate chirp parameters and estimate power consumption.

```
% ****************************************************************
% MotionDetect:  Chirp configuration and Processing chain are
% designed to detect moving objects, including estimation of
% range, velocity and angle of the objects.  It is typically
% useful for detecting, localizing and tracking objects in
% indoor or outdoor settings.
% ****************************************************************
channelCfg 7 3 0
chirpComnCfg 8 0 0 256 4 28 0
chirpTimingCfg 6 63 0 75 60
frameCfg 2 0 200 64 1000 0
guiMonitor 2 1 0 0 0 1
sigProcChainCfg 16 8 1 0 4 4
cfarCfg 2 8 4 3 0 12.0 0 0.5 0 1 1 1
aoaFovCfg -60 60 -40 40
rangeSelCfg 0.1 12.0
clutterRemoval 1
compRangeBiasAndRxChanPhase 0.0 1.00000 0.00000 -1.00000 0.00000 1.00000 0.00000 -1.00000 0.00000
1.00000 0.00000 -1.00000 0.00000
adcDataSource 0 C:/ti/mmwave_lp_sdk/examples/datapath/common/testBench/major_motion/
adc_data_0001_CtestAdc6Ant.bin
adcLogging 0
lowPowerCfg 1
factoryCalibCfg 1 0 40 3 0x1ff000
mpdBoundaryBox 1 -1.5 1.5 0 3.8
sensorStart 0 0 0 0
```

Technique 1 - Increase the time between frames

Description

Taking more time in between frames allows the device to spend a greater proportion of its time in the Interframe Idle mode, and specifically, in the Deep Sleep state. Since the Interframe Idle / Deep Sleep state is the lowest power level of the device, increasing the amount of time the device spends in Deep Sleep reduces the overall

power consumption by reducing the impact of the power drawn from the higher-power modes, such as Active, Interchirp Idle and Interburst Idle. This brings the average power closer to the power drawn in Interframe Idle / Deep Sleep. The three equations below describe this mathematically.

$$Power_{Average} = \left(Power_{Interframe\ Idle} \times Time_{Interframe\ Idle} + Power_{Interburst\ Idle} \times Time_{Interburst\ Idle} + Power_{Interchirp\ Idle} \times Time_{Interchirp\ Idle} + Power_{Active} \times Time_{Active}\right) \div \left(Time_{Interframe\ Idle} + Time_{Interburst\ Idle} + Time_{Interchirp\ Idle} + Time_{Active}\right) \quad (1)$$

$$Power_{Interframe\ Idle} < Power_{Interburst\ Idle}, Power_{Interchirp\ Idle}, Power_{Active} \quad (2)$$

$$As\ Time_{Interframe\ Idle} > > >, \ Power_{Average} \rightarrow Power_{Interframe\ Idle} \quad (3)$$

Configuration File Location

Parameter	Command Line Argument Name	Command Line Argument Position	Units
framePeriodicity	frameCfg	5	Milliseconds

Performance Impact

Increasing the time between frames increases latency and potentially tracking performance. Tracking fast moving objects may not be feasible with a large time between frames.

Example

In the below example, the frame periodicity is set to 1000, meaning that a new frame starts every 1000 milliseconds.

```
frameCfg 2 0 200 64 1000 0
```

Technique 2 - Decrease the time spent in Interchirp Idle and Interburst Idle

Description

Decreasing the time spent in Interchirp Idle and Interburst Idle can increase the amount of time spent in Interframe Idle mode. Although the radar device goes into low power states in between chirps and bursts, the Interframe Idle / Deep Sleep mode that the radar can go into in between frames will draw much less power. Therefore, spending more time in the deep sleep mode and less time in Interchirp Idle and Interburst Idle decreases power overall.

Configuration File Location

Parameter	Command Line Argument Name	Command Line Argument Position	Units
burstPeriodicity	frameCfg	3	Microseconds
chirpIdleTime	chirpTimingCfg	1	Microseconds
chirpRampEndTime	chirpComnCfg	6	Microseconds
numOfChirpsInBurst	frameCfg	1	None

Performance Impact

Decreasing the amount of time between chirps is strongly recommended if the chirps use different transmitters. In the Time Division Multiplexing (TDM) and Binary Phase Modulation (BPM) schemes discussed in greater detail in [3], effective angle calculation assumes that both transmitters transmit at approximately the same time. In reality though, they transmit one at a time. By reducing the amount of time between chirps that use different transmitters, the error created by this assumption is reduced.

However, decreasing the time between bursts will affect the range of velocities that the radar device can detect and discern between[1]. This is because velocity is measured through differences in the radar returns on the same transmitter/receiver pair. With bursts happening faster in succession, the minimum velocity that the radar can

detect will shrink. Small movements may still be detected through the minor motion mode, which looks across frames for points, but the velocity of these minor motion points may not be possible to accurately estimate.

Note

The Interburst Idle and Interchirp Idle times must meet the minimum requirements discussed in Section 1. To detect faster velocities and reduce the power consumption lower than the bursting scheme allows, users may consider using a single burst per frame, and multiple chirps per burst, eliminating the minimum 95 µsec Interburst Idle requirement.

Example

In the below example, the Burst Periodicity is 200 µsec, the Chirp Idle Time is 6 µsec, the Number of Chirps in a Burst is 2, and the Chirp Ramp End Time is 28 µsec. So, the Interchirp Idle time is 6 µsec and the Interburst Idle time is 132 µsec.

$$Interburst\ Idle\ Time = Burst\ Periodicity - (Number\ of\ Chirps\ in\ Burst \times (Chirp\ Idle\ Time + Chirp\ Ramp\ End\ Time)) \quad (4)$$

```
chirpComnCfg 8 0 0 256 4 28 0
chirpTimingCfg 6 63 0 75 60
frameCfg 2 0 200 64 1000 0
```

Technique 3 - Decrease the Chirping Time

Description

Decreasing the chirp time reduces power by forcing the device to spend less time chirping. Efficient chirp design would ensure that all samples are taken quickly and that very little time is spent chirping before and after the samples are being taken. The sequence of events that happen before, during, and after a chirp are explained below. The total amount of time the device spends chirping is known as the Ramp End Time, and is calculated by the following.

[1] This assumes that burst mode is being used rather than normal mode. Both are described in detail in [2]

$$Ramp\ End\ Time\ =\ ADC\ Start\ Time\ +\ Chirp\ Time\ +\ Excess\ Time \quad (5)$$

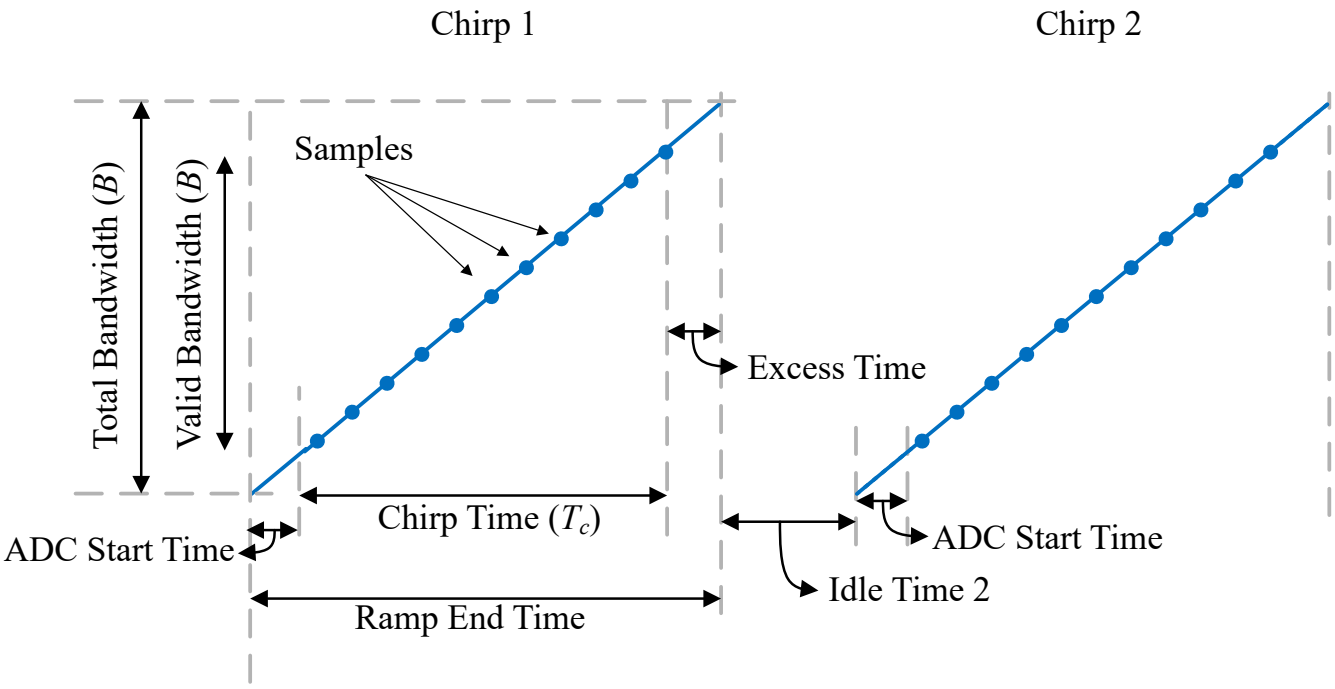


Figure 3-1. Sequence of Events in a Chirp

The procedure for setting an appropriate Ramp End Time is as follows:

1. Compute the Chirp Time, which is the time needed to collect all the ADC samples.

$$Chirp\ Time\ =\ Sampling\ Rate \times Number\ of\ Samples \quad (6)$$

2. Set an appropriate ADC Start Time. Typical values are around 5 µsec. Shorter ADC start times may result in less linear chirping slopes.

Note

The ADC start time is dictated by the ADC Skip Samples and the sampling rate. The minimum recommended ADC Skip Sample Number is max(24, 3 µs x Sampling Rate in MHz).

3. As per [2], use at least 7 samples of Ramp xEcess Time for the short filter (dfeFirSel = 1) and 10 samples of ramp excess time for the long filter (dfeFirSel = 0).

4. Set the Ramp Time equal to the sum of the Chirp Time, ADC Start Time and Ramp Excess Time.

$$Ramp\ Time\ =\ Chirp\ Time\ +\ ADC\ Start\ Time\ +\ Ramp\ Excess\ Time \quad (7)$$

Reducing the ramp time requires minimizing all three of its components. The ADC Start Time and Ramp Excess Time may be set to their minimum allowed values without affecting the range and velocity measurements. Sampling more quickly allows the Chirp Time to be reduced too. But, if the number of chirps and the frequency slope remain constant, reducing the chirp time would result in lower bandwidth and coarser range bin sizes (although with the benefit of increased maximum range). Valid Bandwidth (B) and Range Resolution are calculated as shown below, where c is equal to the speed of light (3E8 m/s)

$$B\ =\ ADC\ Sampling\ Time \times Frequency\ Slope \quad (8)$$

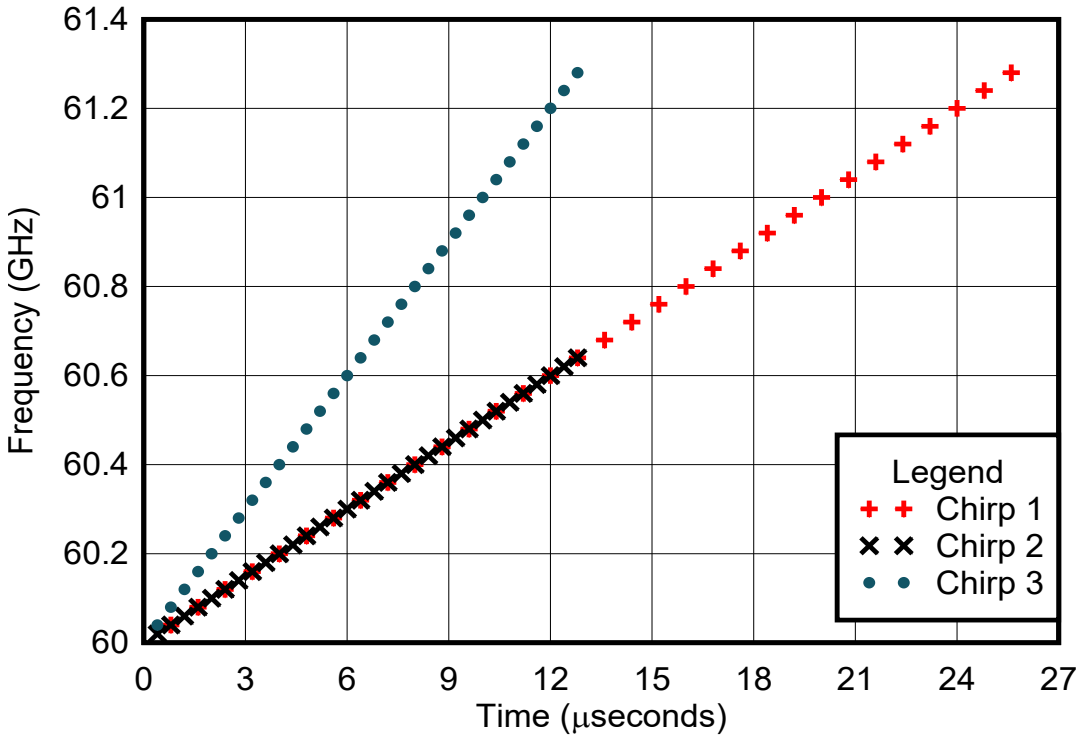
$$Range\ Resolution\ =\ \frac{c}{2B} \quad (9)$$

This strategy of increasing the sampling rate is illustrated with Chirp 2 below, which has all the same parameters as Chirp 1, but samples more rapidly, resulting in a coarser range resolution. However, the active time may be reduced without reducing the bandwidth if the device increases both the sampling rate and the slope of the chirp by the same factor, as illustrated by Chirp 3 below. By collecting the same data over the same bandwidth at a faster rate, the device can spend less time chirping and more time in lower power modes, decreasing overall power draw.

Note

Chirps 1-3 are meant to illustrate tradeoffs, not to advertise best performance.

	Chirp 1 (Slower Sample Rate)	Chirp 2 (Faster Sample Rate)	Chirp 3 (Faster Sample Rate, Larger Frequency Slope)
ADC Sampling Frequency	1.25 MHz	2.5 MHz	2.5 MHz
Number of ADC Samples	32	32	32
Chirp Time	25.6 µsec	12.8 µsec	12.8 µsec
Frequency Slope	50 MHz/µsec	50 MHz/µsec	100 MHz/µsec
Effective Bandwidth	1280 MHz	640 Mhz	1280 MHz
Range Resolution	0.117 m	0.234 m	0.117 m
Maximum Range	1.5 m	3 m	1.5 m



Configuration File Location

Parameter	Command Line Argument Name	Command Line Argument Position	Units
digOutputSampRate (ADC Sampling Rate)	chirpComnCfg	1	Decimator Value. Sampling Rate is set by 100 MHz/ digOutputSampRate
chirpRfFreqSlope	chirpTimingCfg	4	MHz/microsecond
chirpRampEndTime	chirpComnCfg	6	Microseconds

Parameter	Command Line Argument Name	Command Line Argument Position	Units
chirpADCSkipSamples	chirpTimingCfg	2	Number of Samples

Performance Impact

Decreasing the chirp time will decrease the maximum range at which people and objects can be detected by the radar. See the radar range equation in [4] for more information.

Example

```
chirpTimingCfg 20 63 0 30 60
chirpComnCfg 8 0 0 256 4 70 0
```

In the above configuration sample, obtain the sampling rate by dividing 100 MHz by the digOutputSampRate, which is set to 20, yielding 5 MHz.

$$Sampling\ Rate = \frac{100\ MHz}{digOutputSampRate} = \frac{100\ MHz}{20} = 5\ MHz \quad (10)$$

The number of ADC Skip Samples is set to 63, meaning that sampling will only start after 12.6 µsec

$$\text{Sample Starting Time} = \frac{63\ Samples}{5\ MHz} = 12.6\ \mu sec \quad (11)$$

12.6 µsec is more than enough time for the chirp ramp slope to become reliably linear. Reduce it to 5 µsec to save power. This results in 25 samples, which is acceptably above the minimum value of 24, yielding the following configuration file.

$$5\ \mu sec \times 5\ MHz = 25\ samples \quad (12)$$

```
chirpTimingCfg 20 25 0 30 60
chirpComnCfg 8 0 0 256 4 70 0
```

Now the ADC Start Time is 5 microseconds.

The Chirping Time is equal to the number of samples divided by the sampling rate.

$$\text{Chirping Time} = \frac{256\ Samples}{5\ MHz} = 51.2\ \mu sec \quad (13)$$

Since the dfeFirSel is equal to 0, it is recommended to take 10 samples of ramp end time (2 µsec) as per the instructions in [2]

$$10\ \text{samples} \times \frac{1}{5\ MHz} = 2\ \mu sec \quad (14)$$

Overall, this chirp requires 51.2 (Chirping Time) + 5 (ADC Start Time) + 2 (Ramp Excess Time) = 58.2 µsec. So, the ramp end time may be reduced from 70 µsec to 59 µsec, saving 11 µsec of time in the high power chirping state.

```
chirpTimingCfg 20 25 0 30 60
chirpComnCfg 8 0 0 256 4 59 0
```

Now that the chirp has been optimized to reduce time spent chirping before and after sampling ADC data, we can begin to reduce the time spent collecting data. Currently, the chirp's valid bandwidth can be computed as follows:

$$\text{B} = \text{Frequency Slope} \times \text{Chirp Time} = 30\frac{MHz}{\mu sec} \times 51.2\ \mu sec = 1536\ MHz \quad (15)$$

This chirp's active time could be decreased by increasing the chirpRfFreqSlope and the sampling rate by the same factor of 2. So the Sampling Rate becomes 10 MHz (which corresponds to a digOutputSampRate of 10) and the Chirp Frequency Slope chirpRfFreqSlope becomes 60.

```
chirpTimingCfg 10 25 0 60 60
chirpComnCfg 8 0 0 256 4 54 0
```

Now that the sampling rate is equal to 100/10 = 10 MHz, the number of ADC Skip Samples should be increased to 50 to keep the ADC Start Time at a constant 5 µsec.

```
chirpTimingCfg 10 50 0 60 60
chirpComnCfg 8 0 0 256 4 54 0
```

But, the chirping time is now equal to 25.6 µsec. So the sum of the ADC Start Time and the Chirp Time is now equal to 30.6 µsec. Allocating 10 more samples for the Ramp Excess Time requires 1 µsec, yielding a minimum of 31.6 µsec for the Ramp End Time. So, the Ramp end time can be set to 32 µsec, a 54% reduction in the chirping time from the original 70 µsec.

```
chirpTimingCfg 10 50 0 60 60
chirpComnCfg 8 0 0 256 4 32 0
```

Technique 4 - Reduce the number of transmitters/receivers

Description

Reducing the number of transmitters or receivers will decrease the amount of power drawn in active mode. Decreasing the number of transmitters and receivers reduces the amount of power needed to supply the transmitter and receiver circuitry for any particular chirp. If the device is using TDM (see [3]) to use multiple transmit antennas, decreasing the number of transmitters will also decrease the amount of time the device spends chirping, thereby reducing power. The transmitters draw more power than receivers, so TI advises reducing the number of transmitters before reducing the number of receivers.

Configuration File Location

Parameter	Command Line Argument Name	Command Line Argument Position	Units
rxChCtrlBitMask	channelCfg	1	Bit-mask indicating which receivers are on/off.
txChCtrlBitMask	channelCfg	2	Bit-mask indicating which receivers are on/off.

Performance Impact

Reducing the number of transmit/receive channels will directly reduce the angular resolution of the radar device. The angular resolution of the radar, defined as the minimum spacing needed to distinguish between two objects in the same range bin, is 2 / N radians, where N is the number of transmit/receive channels in that angle (azimuth vs elevation). For example, the IWRL6432BOOST Evaluation Board has 4 channels in the azimuth and 2 channels in the elevation. So in the azimuth it has 2 / 4 radians = 28.6 degrees of angular resolution, and in the elevation it has 2/2 = 1 radian = 57 degrees of angular resolution. Removing a transmitter will decrease the

number of available channels from 6 to 3, which will decrease the angular resolution in the azimuth and/or the elevation, depending on the antenna layout selected.

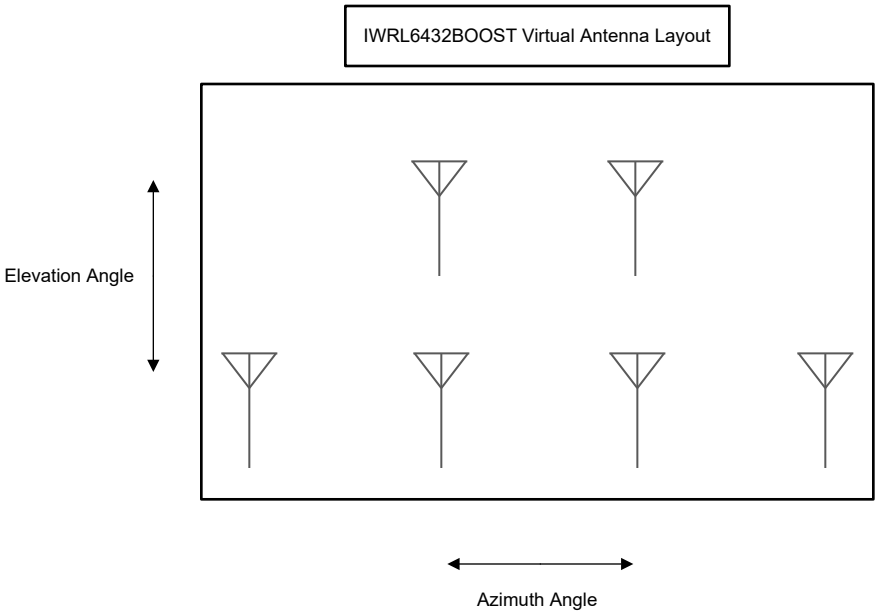


Figure 3-2. IWRL6432BOOST Virtual Antenna Layout

Additionally, removing a transmit/receive pair will diminish the SNR of the measurement, reducing the maximum range at which the radar can detect a given target. More transmit/receive channels increase the accuracy of a given burst by giving more observations of the same burst.

Example

rxChCtrlBitMask can be set to values between 0-7. When converted to binary, the number set is which receivers are powered on (indicated by a 1) and which are powered off (indicated by a 0). In the below example, the value 6 = 0b110 powers on the second and third receivers, while powering off the first receiver.

TxChCtrlBitMask can be set to values between and 0-3. In the same procedure as the rxChCtrlBitMask, when converted to binary, the number set is which transmitters are powered on (indicated by a 1) and which are powered off (indicated by a 0). In the below example, the value 3 = 0b11 powers on both transmitters.

```
channelCfg 6 3 0
```

Technique 5 - Reduce the transmit power

Description

One of the simplest ways to reduce power consumption during a chirp is by decreasing the amount of power broadcast out of the radar device. Users may set a TX Backoff power, which is the amount by which to decrease the power by from the maximum of 11 dBm of conducted output power.

$$Total\ Output\ Power = Maximum\ Output\ Power - TX\ Backoff\ Value \quad (16)$$

Configuration File Location

Parameter	Command Line Argument Name	Command Line Argument Position	Units
txBackoff	factoryCalibCfg	4	dB

Performance Impact

Decreasing the amount of output power may diminish the maximum range of the device. The amount of power broadcast by the device will determine the amount of power available for the device to receive. Once the amount of received power falls to the device's noise floor, the device will no longer be able to accurately detect targets. Decreases in output power can be compensated by equivalent decreases in the CFAR threshold scale to a certain extent, but not indefinitely. Once the CFAR threshold scale becomes too low, false positive detections will overshadow the true detections.

Example

In the below example, since the fourth value of the factoryCalibCfg line is set to 3, the total output power will be equal to 11-3 = 8 dBm.

```
factoryCalibCfg 1 0 40 3 0x1ff000
```

4 Reducing Power in the Interframe Idle and Deep Sleep States

Reducing Power in the Idle State

The Interframe Idle state is a software programmable state in the device. In this mode, the device is either waiting for the command from the external host or transferring the captured samples over the SPI or CAN interface. No radar processing or data acquisition is happening in the device while the oscillator circuitry and perhaps PLL (depending upon the host interface peripheral used) of the device is up and running. In the Interframe Idle state, the user may power down or clock-gate portions of the device, lowering power.

The xWRL6432 allows the following peripherals to be turned off with the sequences shown below.

Note

The Presence and Motion Detection Demo Example in the MMWAVE-L-SDK already shuts off the following peripherals when they are not being used.

Subsystem to be powered off	Method to power it off
Hardware Accelerator	HWA_close(HWA_Handle handle); HWA_deinit()
Front-End Controller Subsystem	MMWave_stop(gMmwMssMCB.ctrlHandle,&err); MMWave_close(gMmwMssMCB.ctrlHandle,&err); MMWave_deinit(gMmwMssMCB.ctrlHandle,&err);
UART	Drivers_uartClose() * Included in Drivers_close() as a part of sysconfig.
I2C	Drivers_i2cClose() * Included in Drivers_close() as a part of sysconfig.
EDMA	Drivers_edmaClose() * Included in Drivers_close() as a part of sysconfig.
QSPI	Drivers_qspiClose() * Included in Drivers_close() as a part of sysconfig.

Clocking

Instead of powering off peripherals completely, users may instead choose to reduce the amount of power they consume by altering the peripherals' clocking options. The most common ways to accomplish this are the following :

- *Changing the Input Clock Source*

Many peripherals can be fed from multiple different clock sources. Typically, faster clock sources cause peripherals to draw more power during operation. [5] enumerates the clock control (CLKCTL) registers available for changing the input clock to the peripherals.

- *Further Clock Reduction via Clock Dividers*

After the clock source to a peripheral has been set, it can be divided to reduce its frequency, which subsequently reduces the power it draws. The same clock control (CLKCTL) registers mentioned above, found in [5], allow users to divide the input clocks down to the desired frequency.

- *Clock-Gating*

Clock-gating keeps the input clock to a peripheral from ticking. This means the peripheral will not operate until it is no-longer clock gated. Clock-gating draws more power than shutting off the peripheral altogether, but it also requires less time and energy to power back on from the off state, making it advantageous for quicker responsiveness. The same clock control (CLKCTL) registers mentioned above, found in [5], allow users to gate the input clocks to the peripherals.

Reducing Power in Deep Sleep Mode

Deep Sleep is the lowest possible hardware power state of the device. In deep sleep, nearly the entire device is powered off to save a considerable amount of power. Just the vital contents on the device, such as the application image and chirp profile are retained in deep sleep. These preserve the device's context, allowing it to re-enter operation without a reboot.

The benefits of Deep Sleep mode are seen most prominently in low duty cycle applications like presence detection, in which the device is only active for short amounts of time. For the remainder of the time, when the device does not need to be active, the system can go into Deep Sleep mode, saving power and extending lifetimes for battery powered applications.

The power consumed in the Deep Sleep state depends mainly on two factors: the amount of memory retained and the states of the IO pins during deep sleep.

Memory Retention on the xWRL6432 in Deep Sleep

The xWRL6432 has 1 MB of memory divided across the APPSS, FECSS and HWASS Power Domains. More power is consumed in deep sleep mode when more memory is retained.

Table 4-1. xWRL6432 Main Memory map

Device Variant	APPSS_RAM	RADAR_DATACUBE_RAM	SHARED_RAM[2]	FECSS_RAM
xWR6432	512 KB	256 KB	256 KB	32 KB

The APPSS_RAM and SHARED_RAM are further divided into different memory banks as shown below.

Table 4-2. APPSS and Shared RAM Memory Bank

Memory Bank	Memory Location	Memory Space (in KB)
Bank 1	APPSS	256
Bank 2	APPSS	128
Bank 3	APPSS	128
Shared_RAM_1	Shared RAM	128
Shared_RAM_2	Shared RAM	128

The xWRL6432 device can save power by reducing the amount of memory retained in the Deep Sleep mode. On the xWRL6432, memory instances consume approximately 45 µW / 64kB in power-down retention mode. If memory within a cluster is not retained, it will consume approximately 20 µW/64 kB. To further save power when memory is not retained, some memory clusters on the xWRL6432 are grouped together, and power switches are deployed on the corresponding SRAM power rails. If all the clusters in a group are not retained during Deep

[2] SHARED_RAM is shared between APPSS_RAM and RADAR_DATACUBE_RAM

Sleep, switches for the group are opened, completely powering off the group, causing the memory to consume 0 µW/kB while completely powered off (no retention).

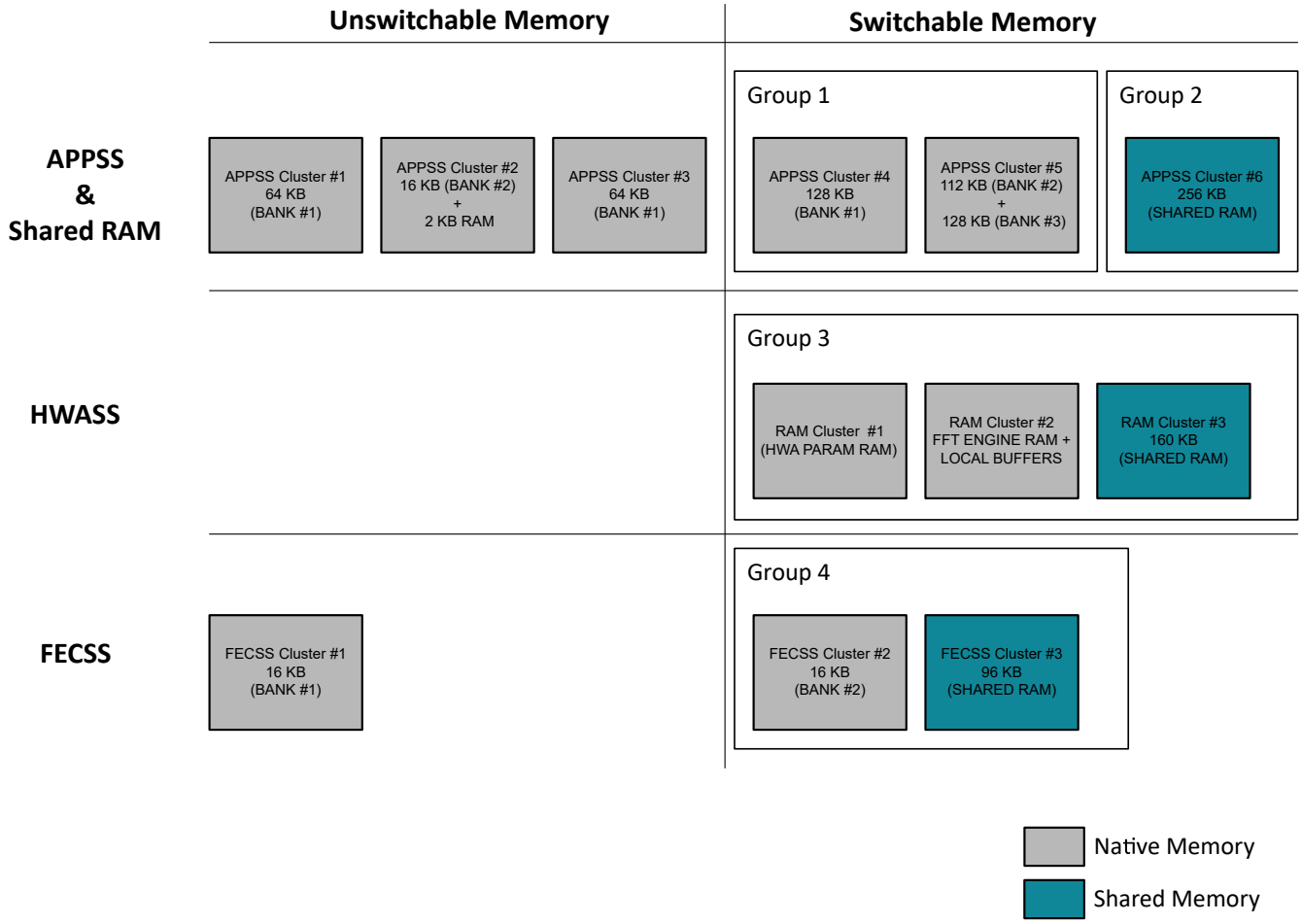


Figure 4-1. Switchable and Unswitchable Memory

CAUTION

The FECSS Clusters #1 and #2 are handled completely by the DFP. Not retaining memory on them may result in unexpected and erroneous behavior.

Powering down specific memory banks can be accomplished using SysConfig, a TI tool used to configure many TI devices. When disabling RAM Retention, ensure that none of the data needed for operation in the subsequent frame, such as device code, is stored in the RAM that is not retained. If this occurs, it will prevent proper operation, as the information the device needs to operate will have been erased.

Impact of Digital IO Power

There are various digital peripheral IOs, for instance, SPI, UART, CAN-FD, I2C, LIN, etc. in the xWRL6432 to transfer data, download images or establish a connection with an external MCU. Since the deep sleep power state draws very little power, any floating node can cause significant current leakage. It is therefore important to approriately park these IOs to their relevant pull up/pull down or high impendance state when the xWRL6432 device is in a deep sleep mode. This can be accomplished again using SysConfig.

To minimize power consumption, TI recommends parking all available pins in deep sleep. Pins needed to serve as wakeup sources for the xWRL6432 can remain in their default input states. Additionally, pins that route signals out of the xWRL6432 that need to maintain their state while the device is in deep sleep (say for example a reset line to another device, or an LED light) can also remain in their default output states.

5 Measuring Power

Power Measurement on the IWRL6432BOOST

The IWRL6432BOOST EVM allows users to measure the power consumed by the xWRL6432 device through the INA current sense amplifiers onboard. The motion and presence detection demo provided in the xWRL6432 SDK includes an output that shows the average current measured on the power rails provided to the device. Additionally, the I2C interface that reads from the INA current sense amplifiers can also be read from an external microcontroller for situations where recording the output on the xWRL6432 device is not feasible (e.g. measuring deep sleep power). See [2] for details on how to enable and interpret the power measurement packets.

CAUTION

The Rev A. and Rev B. IWRL6432BOOST do not provide the ability to measure power accurately due to the limitations of the INA226 devices placed onboard. The Rev A. and Rev B. IWRL6432BOOSTs require the following changes to measure power properly.

Reference Designator	Initial Device	Replacement Device
R200, R192, R193	PMR18EZPFV2L00 2 mΩ Resistor	WSL1206R0400FEB 40mΩ Resistor
R134	PMR18EZPFV2L00 2 mΩ Resistor	WSL1206R0200FEB 20mΩ Resistor
U6, U7, U8, U25	INA226AIDGST Current Monitor	INA228AIDGST Current Monitor

6 References

1. IWRL6432 Single-Chip 57- to 64-GHz Industrial Radar Sensor datasheet
2. Motion and Presence Detection Demo Tuning Guide
3. MIMO Radar
4. Programming Chirp Parameters in TI Radar Devices
5. IWRL6432 Technical Reference Manual

7 Revision History

DATE	REVISION	NOTES
April 2023	*	Initial Release

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265