SPU input and output DMA

TriCore™ TC3xx AURIX™ family

About this document

Scope and purpose

The second generation AURIX™ offers enormous computing power and fast responsive real-time behavior. It also supports low-, mid-, and high-end Radar applications. To ensure the fastest signal processing capabilities a high-performance signal processing unit (SPU) is integrated.

This document describes the input and output direct access memory (DMA) mechanism of the SPU using second generation AURIX™.

This application note is applicable for all second generation AURIX™ devices with at least one SPU instance available. (TC39x, TC35x, TC33x).

Intended audience

This document is intended for design engineers, technicians, and developers of electronic systems with basic knowledge of radar processing and the SPU.

Table of contents

1 Overview

1.1 Input and output DMA in the signal processing unit (SPU) subsystem

Figure 1 shows input and output DMA (marked in yellow) in the SPU subsystem. The input DMA engine can either passively accept data pushed by the analog-digital-converter (ADC) interfaces (RIF or Radar Interface) or actively load data of an existing data cube from the Radar Memory also referred to as Extension Memory (EMEM). The task of the input DMA is to structure data in the buffer memory for subsequent processing in the SPU pipeline. Thus, the input DMA has to make sure that all data required to perform the configured processing steps is fed into the processing pipeline at a time. Examples where a complete data set is required include:

(1) **Fast Fourier Transforms (FFT):** All data points to perform the FFT need to be available.

(2) **Integration of results across multiple antennas:** Data points from all antennas need to be available.

The output DMA engine is used to write results to Radar Memory (EMEM).

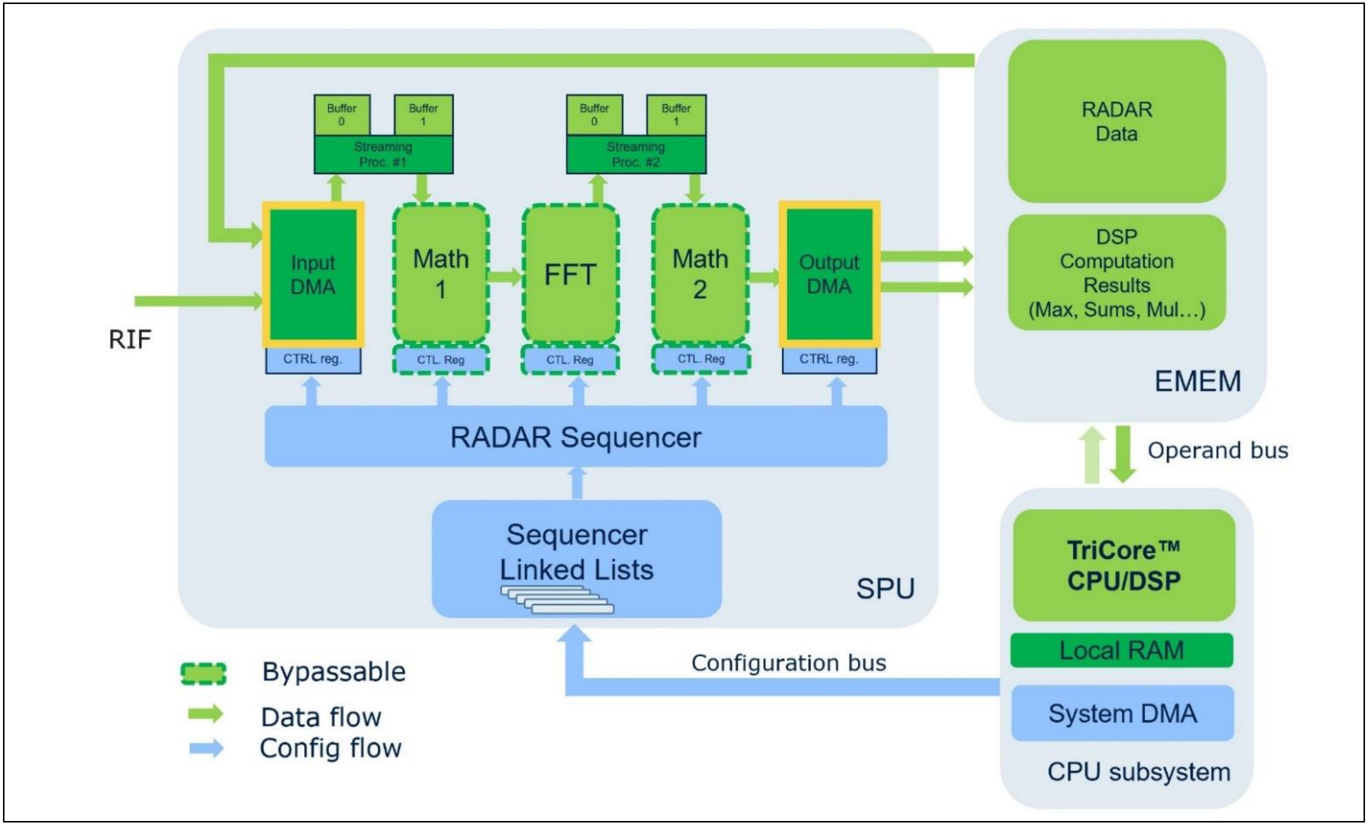

Figure 1 Input DMA and output DMA in the SPU system

2 Working principles of input DMA

2.1 Depiction of radar data in Radar Memory

In radar applications data can often be ordered in three dimensions: multiple samples taken along a frequency sweep, multiple subsequent frequency sweeps (ramps), and data recorded by multiple antennas. To account for these three dimensions of data in Radar Memory, the schematic shown in Figure 2 is used to represent the flat 1D data array stored in Radar Memory. The two left columns are used as indices for – depending on context – antenna, ramp or sample index.

For better understanding concrete values are depicted while these can be adapted to each dedicated use case. The values used are summarized in Table 1.

Table 1 Parameters and values of radar data dimensions used in this document

Parameter	Symbol	Value
Number of samples per ramp	$N_{samples}$	1024
Number of antennae	N_{ant}	4
Number of ramps	N_{ramps}	128
Number of range bins after 1st FFT	N_{rangeb}	512

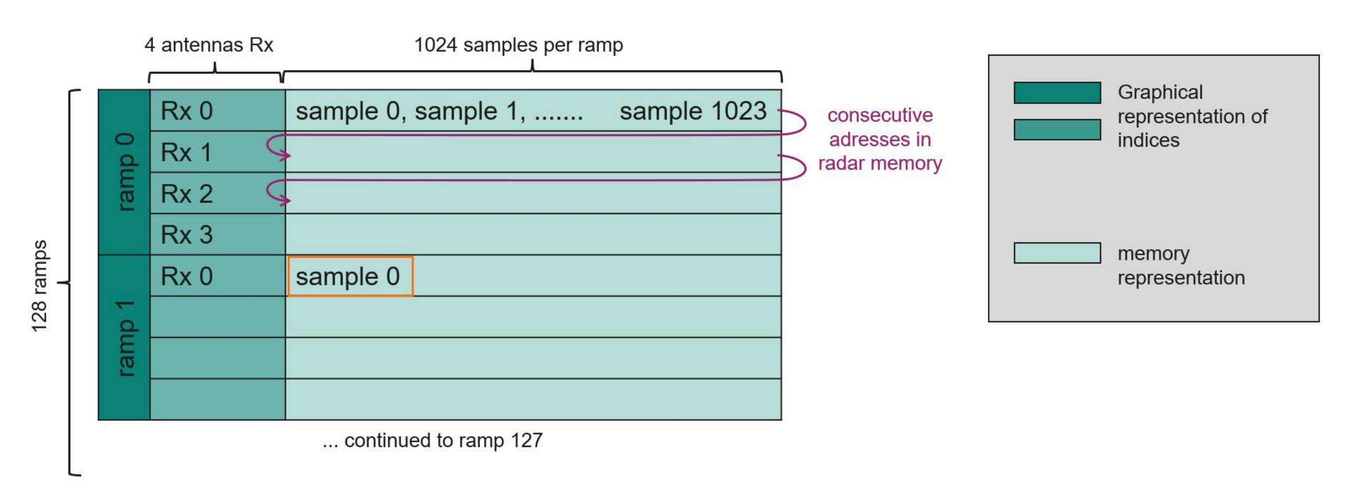
4 antennas Rx
1024 samples per ramp
128 ramps
ramp 0
ramp 1
Rx 0
Rx 1
Rx 2
Rx 3
Rx 0
sample 0, sample 1, ... sample 1023
sample 0
consecutive adresses in radar memory
... continued to ramp 127
Graphical representation of indices
memory representation

Figure 2 Structure of radar data in Radar Memory

Note: The marked memory location (orange square) corresponds to (ramp 1-antenna 0-sample 0).

2.2 Using EMEM input

The input DMA operates either with the RIF or the Radar Memory (EMEM) as data source (compare Figure 1). This section describes the use of EMEM data as input.

The input DMA engine is designed to perform data transfers from Radar Memory to the SPU processing pipeline using address offsets from a configurable base address. Data is read from Radar Memory in 256-bit words and the input DMA can read 256 bits of data every two clock cycles.

In order to access data in Radar Memory the three dimensionional structure is accounted for by three configurable loops that generate the address offset from the configured base address:

- Three offset values (two configurable and one that has to be set to sample size)
- Three configurable repeat values

These three nested loops allow rotation of the 3-dimensional data cube when the SPU is loading data from or writing data to the EMEM. The offsets are used in a "bin loop", an "inner loop" and an "outer loop" processing flow. The inner loop can potentially be executed multiple times for each execution of the outer loop and the bin loop can be executed multiple times for each execution of the inner loop. Code Listing 1 shows how the address offset is calculated from loop parameters.

Code Listing 1 Pseudocode illustrating the calculation of the adress offset from loop parameters

```
/* loop parameters (Input DMA configuration parameters bold)
adroffset: address offset
ol_cnt: outer loop counter
olr: outer loop maximum repeat value
ilo: outer loop offset
il_cnt: inner loop counter
ilr: inner loop maximum repeat value
olo: inner loop offset
bl_cnt: bin loop counter
blr: bin loop maximum repeat value
blo: bin loop offset
*/
for(ol_cnt=0;ol_cnt<olr;ol_cnt++){
    for(il_cnt=0;il_cnt<ilr;il_cnt++){
        for(bl_cnt=0;bl_cnt<blr;bl_cnt++){
            adroffset = blo*bl_cnt + ilo*il_cnt + olo*ol_cnt
        }
    }
}
```

2.2.1 Example data read for first stage (range) FFT

This section provides an example where the bin loop offset is set to sample size. Note that examples in Sections 2.2.1, 2.2.2, and 2.2.3 differ with respect to which loop is being set to sample size. The mandatory settings for addressing mode, processing mode and number of blocks described in each section also apply in cases of different concrete settings of e.g., number of ramps and in cases where the other two loops are switched.

In order to provide loop offset and loop repeat configuration parameters (Table 2) for performing a range FFT refer to the structured drawing in Figure 3. In this case, the bins in Radar Memory to be used as consecutive input bins for each FFT dataset are stored in consecutive addresses in Radar Memory hence, each 256 bit read fetches multiple samples for the same FFT, e.g., four, eight, or sixteen bins depending on the precision of the data used. Note that in the value of inner and outer loop offset the number of samples and the number of antennae needs to be considered.

Table 2 Example loop offset and loop repeat configuration parameters (related to Figure 2)

Name	Abbreviation	Value
Bin loop offset	blo	sample size in bytes
Inner loop offset	ilo	$N_{samples}$
Outer loop offset	olo	$N_{samples} \times N_{ant}$
Bin loop repeat	blr	$N_{samples}$
Inner loop repeat	ilr	N_{ant}
Outer loop repeat	olr	N_{ramps}

As consecutive samples to be used as inputs to the FFT are stored in consecutive addresses in Radar Memory address transposition is not required and the linear addressing mode must be selected. Bandwidth optimisation is not needed in this mode as all data fetched in a 256 bit read is used in the same FFT dataset (compare illustration in Figure 3). For details on address transposition and bandwidth optimization refer to Chapter 2.2.2.

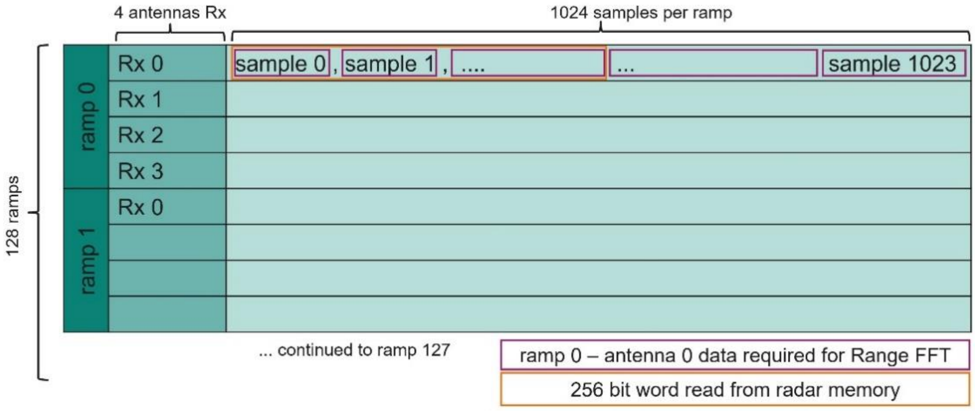

Figure 3 Radar data structure in Radar Memory prior to range FFT

Note: All data fetched in a 256 bit read is used in the same range FFT dataset.

To perform an integration across the results from multiple antenna in the Math2 unit of SPU, data from all antennae has to be available in Math2 for integration. This can be ensured by configuring the input DMA accordingly. By setting the Processing Mode to Integration the input DMA takes care that data from all antennae is available for subsequent integration.

Table 3 ID_RM_CONF register settings required when bin loop offset is set to sample size

Description	Field	Value	Comment
Addressing Mode	TRNSPS	**0 (linear)**	**Mandatory setting**
Processing Mode	PM	0,1 (default, integration)	Can be selected
Number of simultaneous data blocks from RAM (bandwidth optimization)	BLOCKS	**0 (1 data block)**	**Mandatory setting**

2.2.2 Example data read for second stage (Doppler) FFT: How to use transpose read

This section provides an example where the outer loop offset is set to sample size. Note that examples in Sections 2.2.1, 2.2.2, and 2.2.3 differ with respect to which loop is being set to sample size. The mandatory settings for addressing mode, processing mode and number of blocks described in each section also apply in cases of different concrete settings of e.g., number of ramps and in cases where the other two loops are switched.

In many cases in radar processing the first (range) FFT is followed by a second FFT referred to as Doppler FFT. This FFT is performed accross all ramps of a dedicated range bin of a dedicated anteannae. The data that the input DMA needs to read for this Doppler FFT is marked in purple in Figure 4. As consecutive samples to be used as inputs to the FFT are stored in non-consecutive addresses in Radar Memory, address transposition is required. This is supported by the input DMA with a hardware transpose unit and is enabled via selection of transpose addressing mode. As the outer loop offset is set to sample size, integration mode needs to be enabled as well (compare mandatory setting in Table 5).

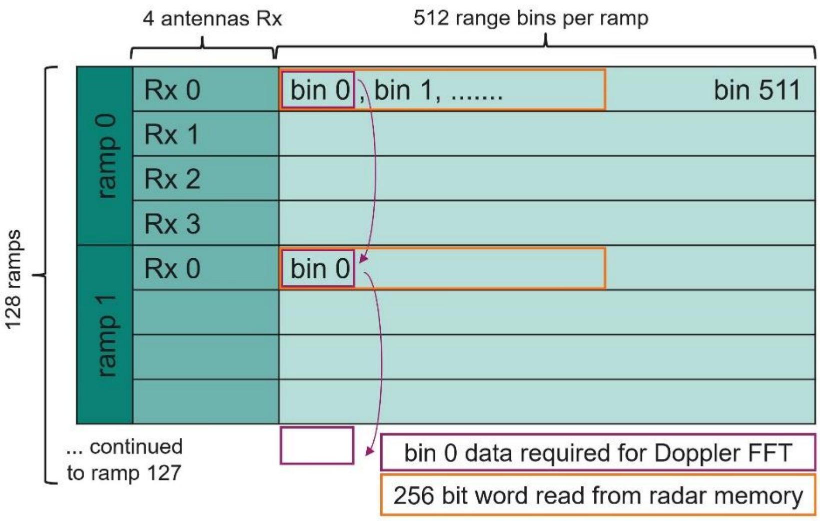

Figure 4 Radar data structure in Radar Memory after range FFT

Note: To make efficient use of the entire 256-bit word read bandwidth optimization can be enabled.

Loop parameter setting for the second stage FFT example are provided in Table 4. Note that the number of samples is set to the number of range bins in this case.

Table 4 Example loop offset and loop repeat configuration parameters for second stage FFT (related to Figure 4)

Name	Abbreviation	Value
Bin loop offset	blo	$N_{samples} \times N_{ant}$
Inner loop offset	ilo	$N_{samples}$
Outer loop offset	olo	sample size in bytes
Bin loop repeat	blr	N_{ramps}
Inner loop repeat	ilr	N_{ant}
Outer loop repeat	olr	$N_{samples}$

In the example sketched in Figure 4, the orange rectangle indicates the 256-bit word read of the input DMA from Radar Memory. When retaining the current configuration, the full 256-bit word read is not utilized efficiently, resulting in only the first range bin (e.g. bin '0' in the first read) being written to the buffer memory.

To avoid discarding data and wasting bandwidth, build multiple data blocks in buffer memory by setting the BLOCKS bitfield to a value other than '0'. This is referred to as bandwith optimization.

Table 5 ID_RM_CONF register settings required when outer loop offset is set to sample size

Description	Field	Value	Comment
Addressing Mode	TRNSPS	**1 (transpose)**	**Mandatory setting**
Processing Mode	PM	**1 (integration)**	**Mandatory setting**
Number of simultaneous data blocks from RAM (bandwidth optimization)	BLOCKS	0,1,3,7 (1,2,4,8 data blocks)	Can be selected

Setting the BLOCKS parameter to '0' results in one dataset being stored in buffer memory at a time. The first buffer is sketched in Figure 5.

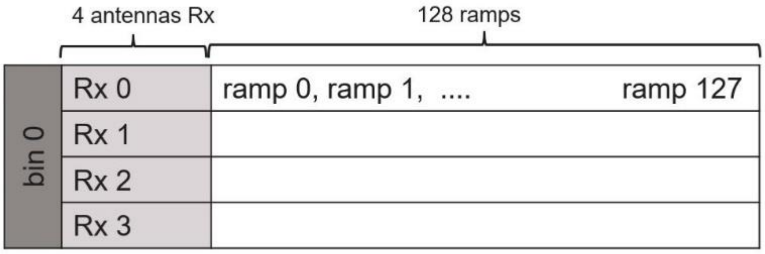

Figure 5 Data structure in first buffer of input DMA with bandwidth optimization disabled (BLOCKS = 0)

Setting the BLOCKS parameter to '3' results in four datasets being stored in buffer at a time. The first buffer is sketched in Figure 6.

Note: The bin offset and bin repeat values do not need to be adapted when changing the BLOCKS parameter.

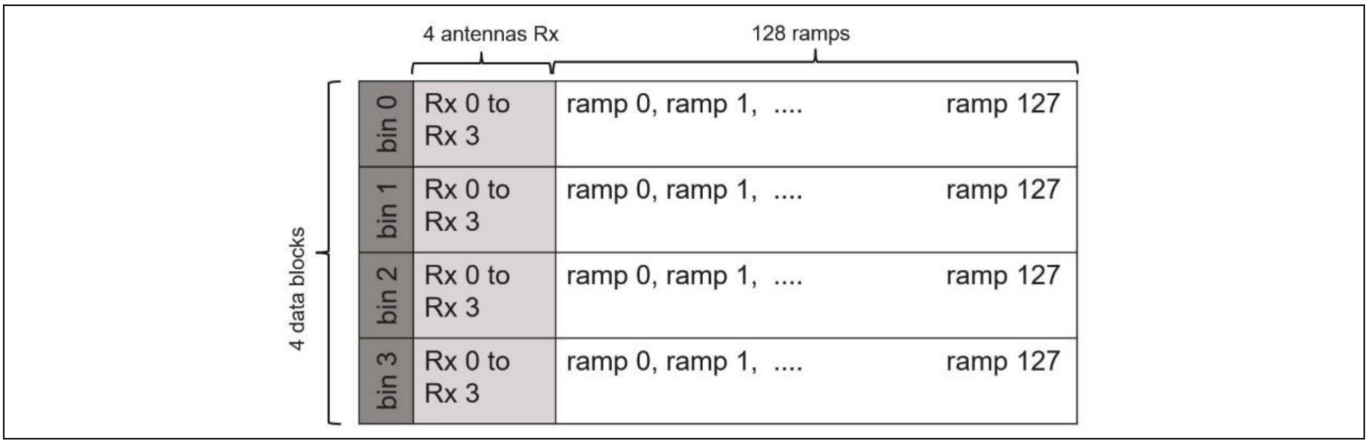

Figure 6 Data structure in first buffer of input DMA with bandwidth optimization enabled (BLOCKS = 3)

2.2.3 Setting inner loop offset to sample size

This section provides an example where the inner loop offset is set to sample size. Note that examples in Sections 2.2.1, 2.2.2, and 2.2.3 differ with respect to which loop is being set to sample size. The mandatory settings for addressing mode, processing mode and number of blocks described in each section also apply in cases of different concrete settings of e.g., number of ramps and in cases where the other two loops are switched.

The second case in which consecutive samples to be used as inputs to the FFT are stored in non-consecutive addresses in Radar Memory is sketched in Figure 7.

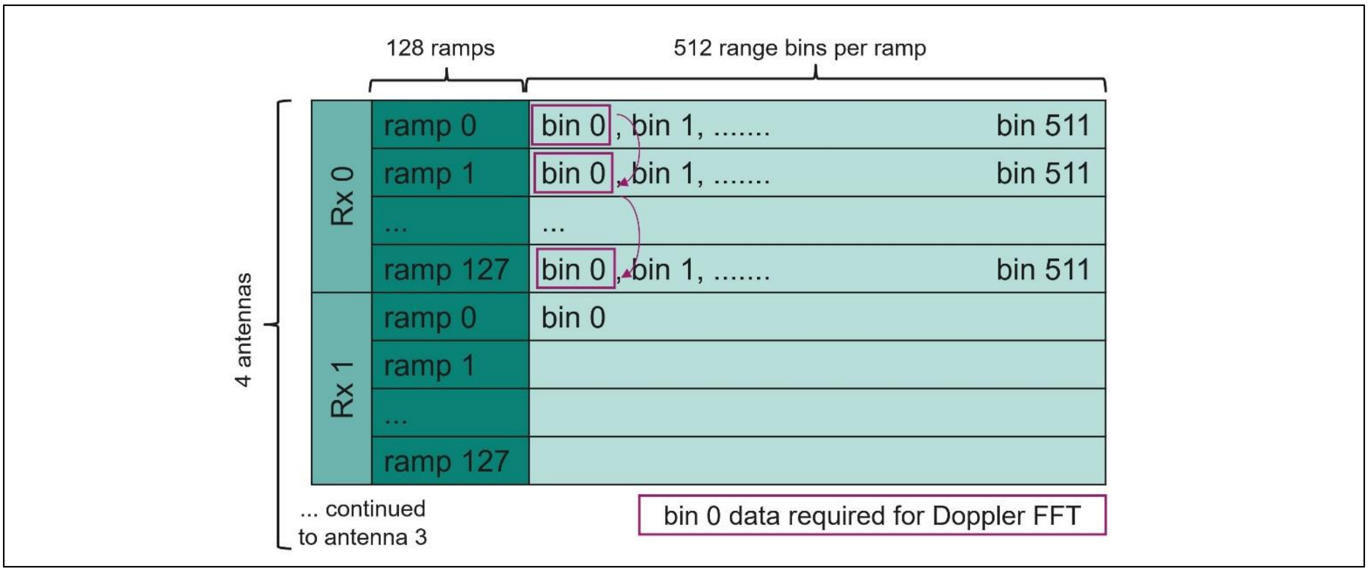

Figure 7 Radar data is structured in Radar Memory after range FFT

Transpose addressing needs to be selected.

Note: *In this case the inner loop offset is set to sample size and integration mode is not supported in this case (Table 6). Additionally, bandwidth optimization is not possible and the number of blocks must be set to '1' (number of blocks is equal to BLOCKS field value '+1').*

Table 6 Example loop offsets and repeat values (related to Figure 7 and Figure 2)

Name	Abbreviation	Value
Bin loop offset	blo	$N_{samples}$
Inner loop offset	ilo	sample size in bytes
Outer loop offset	olo	$N_{samples} \times N_{ramps}$
Bin loop repeat	blr	N_{ramps}
Inner loop repeat	ilr	$N_{samples}$
Outer loop repeat	olr	N_{ant}

Table 7 ID_RM_CONF register settings required when inner loop offset is set to sample size

Description	Field	Value	Comment
Addressing Mode	TRNSPS	**1 (transpose)**	**Mandatory setting**
Processing Mode	PM	0 (default)	integration mode not supported
Number of simultaneous data blocks from RAM (bandwidth optimization)	BLOCKS	**0 (1 data block)**	**Mandatory setting**

2.2.4 Additional limitations to consider

2.2.4.1 The buffer RAM size limit of 32 KBytes

Input and output DMA have a buffer RAM size limit of 32 KBytes that has to be considered. Applying a transpose read with a high number of ramps and bandwidth optimization enabled might cause violation of the buffer limits:

- Example 1: Buffer limit not violated: 256 ramps * 6 antennas * 4 Data blocks * 4 Byte per sample= 24.5 KByte
- Example 2: Buffer limit violated: 256 ramps * 6 antennas * 8 Data blocks * 4 Byte per sample = 49 KByte

2.2.4.2 32-byte data alignment in Radar Memory

Each dataset in memory starts at a 32-byte aligned address. This allows the bandwidth optimization hardware to function and is enforced by the output Data Manager. This restriction has to be observed when using software to build data arrays in memory for reading by the input Data Manager.

2.2.4.3 Maximum number of 2028 bins in an FFT dataset

The maximum number of bins supported in an FFT dataset is 2048.

Note: *Setting a Bin Loop Repeat value for an FFT size greater than ID_RM_BLR.BLR > 2047 is not supported and should be avoided.*

2.3 Using RIF input

The input DMA operates either with the RIF or the Radar Memory (EMEM) as data source (compare Figure 1). This section describes the use of RIF data as input.

The RIF provides a parallel memory interface to the SPU unit with a 32-bit bus width. When loading data from the RIF or RIFs, processing mode is always default and the input data format and number of antennae must be aligned with the RIF configuration settings.

The SPU expects 16 bit signed Qm.n integers as an input from the RIF, which is delivered to the SPU in a 32-bit packet. While the RIF accepts a wider range of formats it is responsible for adjusting the direction, data length (16 bits) and format of the incoming ADC data so it fits the SPU RIF input format.

Configuration setting for using the RIF as a data source for input DMA are:

- Data source (ADC IF 0 or ADC IF 1 or both)
- Ramps per measurement cycle
- Number of active antennae
- Input format (signed or unsigned, real, or complex)
- Samples per ramp

The input DMA implements a ramp counter (referred to as partial acquisition counter PACTR) which can be used to detect if excess data is received or to allow for a delay of start of processing of the SPU. The counter can be configured to generate an error or an interrupt if a particular value is reached.

3 Working principles of output DMA

The output DMA Engine has eight independent data channels with one output FIFO per result source (e.g., FFT, NCI, LOG2). The data width of the output DMA write path to the Radar Memory is 256-bit.

For each output channel, a base address can be configured. The write address will be incremented automatically so that the results of each channel are stored as an array in memory starting from the defined base address. Each FFT result always starts at a 256 bit (32 byte) aligned address. This is mandatory if the data is to be read back into the SPU as the read operations controlled by the input Data Manager require the start address of each dataset to be 256-bit aligned.

The exception is the "In Place FFT" mode for writing the FFT results to the same memory location as the input data used for the FFT. In this case, the start address is configurable while the address sequence is calculated using the input DMA Engine Loop Repeat and Loop Offset parameters. Input DMA Engine Loop Repeat and Loop Offset parameters are inherited from the input DMA setting. In this case the input and output data size in Bytes must be the same.

References

[1] Infineon Technologies AG: *AURIX™ TC3xx User Manual Part-1*; Available online

Revision history

Document revision	Date	Description of changes
1.0	2024-08-05	Initial release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2024-08-05

Published by

Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AN0012

Important notice
The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.