Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

Technical Report

CONFIDENTIAL

To	:	Michiel Baas Franc van den Adel Boris Lippe	Annexen:	195G091 01 USx ASIC Specification V2,0
Cc	:	Nikola Radeljic-Jakic Erwin Dumay		
From	:	A. Flikweert Tel.: 959		
Subject	:	USX application note		

Contents

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

Technical Report

1. Introduction

In this memo it is described how the USX probe is operated. Furthermore, it is described how pulse cancellation is used. The ASIC registers and memory addresses are described in 195G091 01 USx ASIC Specification V2,0. Furthermore, the interface specification of the specific customer is needed.

Operation of the FPGA and ASIC is described in Chapter 2. Additional information, which is optional for the pulsed mode and continuous wave mode, is given in Chapter 3. Default values for the registers are listed in Appendix A.

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

Technical Report

2. Operation

2.1. Introduction to operation

Two methods are available (depending on the customer) for sending data to and receiving data from the ASIC. The first (direct) method is writing to FPGA memory first and next issue a send or instruct command to transfer the data to the ASIC (Section 2.1.1). For the second method, FPGA ASIC memory is used which contains a copy of all ASIC registers (Section 2.1.2).

To setup a mode, first initialization is performed (Section 2.2). After initialization, one of the following modes is chosen:

- Pulsed mode (Section 2.3) for A-mode, B-mode, PW Doppler or Color flow. After setting up the pulsed mode, the beam is executed as described in Section 2.4.
- Continuous wave mode (Section 2.5).

2.1.1. FPGA send/receive memory

Preparing short or long commands (see Table 11 and Table 12 in ASIC specification) requires:

- AddrMask byte. This AddrMask byte is listed in Table 10 in ASIC specification; the high nibble defines the bitmask of the ASIC, e.g. 0x16 for 1st ASIC, 0x26 for 2nd ASIC, 0x36 for both ASICs. Note that the last nibble is always 6;
- CMD byte.
- Data of the register (only for long commands):
    - All core registers (name ending with "_DFF") are reordered (called "scrambling" in the ASIC specification) according to ASIC specification Section 7.2 and have a dummy byte behind the reordered data.
    - The output driver register data ("CONFIG_DRV") is reordered according to ASIC specification Section 7.3 and has a dummy byte behind the reordered data.
    - Periphery registers (starting with "REG_") are not reordered.
- All long commands have a CRC byte and a NOP at the end.
- All short commands have a NOP at the end.

For each command, a two-step process is used:

- The command is written to the FPGA send memory by using the "write" command.
- The command is transferred from the FPGA to the ASIC by using the "send" command.

Note that addresses and write/read lengths are in little endian format. Data is written byte-per-byte.
Since it is a two-step process, the user could also write all registers to the FPGA send memory first and next transfer all registers to the ASIC at once by issuing a send command covering the complete memory block containing all the commands.

2.1.2. FPGA ASIC memory

If FPGA ASIC memory is used, all registers are stored in the FPGA, reordered within the FPGA and copied to the ASIC. The user does not have to care about dummy bytes or CRC.

2.2. Initialization

Prior to usage, the probe needs to be initialized to default values.

When using "FPGA ASIC memory method" (ATDI):

- Write ASIC version number to address 0x400000: 0x02.

Write to FPGA status memory (addresses see interface specification):

- 0x08 ASIC power enable: value 0x01.
- Wait 100 ms after power up.
- 0x00 ASIC manual reset: value 0x00.
- 0x0E Divider for reading data from ASIC: value 0x09.
- 0x0D Sampling point for return data from ASIC: value 0x08.
- 0x0F Internal clock divider: value 0x01.
- 0x07 Clock PLL: value 0x66.
- Wait 2 ms (alternatively, send NOPs to the FPGA during 2 ms; see interface spec)
- 0x00 ASIC manual reset: value 0x01.
- 0x04 Trigger enable: value 0x00
- 0x01 Trigger: value 0x00.
- 0x02 LED disable: value 0x01.
- 0x14 Error reset: value 0x01 – only when using “FPGA ASIC memory method”.

Write/read registers to *all* ASICs (short and long commands, addrMask 0xF6):

- Clear external error pins on all ASICs (short command 0xD2) – only when not using “FPGA ASIC memory method”.
- Write configure built-in self-test: REG_BIST. Do not forget to set DataOutClkDiv to the same value as “Divider for reading data from ASIC”.
- Reading the ASIC status register (BistGetErrorStatus command) from each ASIC to ensure proper communication.
- Reading the ASIC serial number from the first ASIC using long command BistGetFuseMemory 0xCA with addrMask 0x16:
  - 24-bit serial number (optional).
  - 4-bit IbiasCal value (default is 0).
  - 4-bit reserved field.
- Write IbiasCal value into REG_BIST.
- Set Error_NMask_Out (see ASIC specification): REG_CONFIG: 00000000800000.
- Clearing any errors and reading the status:
  - Clear external error pins on all ASICs (short command 0xD2).
  - BistGetErrorStatus command: if no error bits are set, the ASIC has been correctly initialized.

2.3. Setup pulsed mode

To reduce the number of acoustic channels, the probe performs some beamforming steps, it however does not perform any image processing operations. Therefore, operation of the probe is split up in Pulsed Mode and Continuous-Wave Mode. Pulsed modes include A-mode, B-mode, HD2 B-mode, Pulsed wave Doppler, Color flow, etc. Pulsed Mode comprises of an initial setup that is not timing-critical and a timing-critical beam setup.

Short/long commands to ASICs:

- ClearCore.
- Configuration parameters: REG_CONFIG. Here is also selected which delays are used: full delays (Tx delays TXDELAY_DFF and Rx delays RXDELAY_DFF) or compressed delays (Tx delays REG_TXDELAY0 or Rx delays REG_RXDELAY0).
- Transmit pulse configuration: XMITWF_DFF.
- Timing configuration of receive and transmit: REG_BEAMTIMING. For Rx delay equalization see Section 3.1.

- Configuration for dynamic receive: REG_RXDYN. Note that for static receive delays this register is also needed; all slopes and durations must be set to 0. If dynamic receive delays are used, slopes and durations must be set accordingly.
- Analog group configuration: CONFIG_DFF. For Rx delay equalization see Section 3.1.
- Aperture reduction: CWSEL_DFF (if CONFIG_DFF.CW_EN=0 and DISABLE_DISABLE=0).
- Output driver configuration: CONFIG_DRV.

To reduce power consumption, it is recommended to write the CONFIG_DFF and CONFIG_DRV registers last because the changes made to those registers take effect immediately. Changes in any other registers take effect immediately after triggering a beam.

2.4. Run beam(s) for pulsed mode

The send memory in the FPGA can be used to construct a table of beams – sets of Tx delays, Rx delays and TriggerPage. In this way one beam or a scan with multiple beams (varying the steering angle) are stored in the FPGA memory. This is done by using the command “write” for the following long commands (registers) and short command:

- Long command TXDELAY_DFF (Tx delays, no compression) or REG_TXDELAYP0 (Tx delays, compression).
- Long command RXDELAY_DFF (Rx delays, no compression) or REG_RXDELAYP0 (Rx delays, compression).
- Short command TriggerPage00 (start a beam).

Note that REG_CONFIG.BeamRunRxCalc and BeamRunTxCalc define if compression is used or not.
For one beam, 3479 bytes (non-compressed delays) or 35 bytes (compressed delays) are written to FPGA “send memory”; see Table 1 and Table 2 respectively. Multiple beams are sequentially stored in “send memory”. Size of “send memory” is 1,048,576 bytes, thus a table of 301 non-compressed beams or 29,959 compressed beams could be stored in memory.
Transferring the delays from FPGA to ASIC takes ~70 µs for non-compressed delays and less than 7 µs for compressed coefficients.

After storing in memory, a beam is executed by issuing a “Send” command to the FPGA:

- Address (*i*-1)*3479 (non-compressed) or (i-1)*35 (compressed).
- Read length 0x0000.
- Write length 0x0D92 (non-compressed) or 0x0026 (compressed).

where *i* is the beam number (thus maximum 301 for non-compressed or 29,959 for compressed, because otherwise maximum address number is exceeded).

Table 1. Required bytes to set up one beam (non-compressed delays).

	MSK	CMD	Data	Dummy	CRC	NOP	Per ASIC	Total for USX2
Tx delays	1	1	11*64	1	1	1	709	1418
Rx delays	1	1	16*64	1	1	1	1029	2058
TriggerPage*	1	1	0	0	0	1	-	3
Total								3479

*AddrMask is set to 0xF6 to fire with all ASICs.

Table 2. Required bytes to set up one beam (compressed delays).

	MSK	CMD	Data	Dummy	CRC	NOP	Per ASIC	Total for USX2
Tx delay coeff.*	1	1	8	0	1	1	-	12
Rx delay coeff.*	1	1	16	0	1	1	-	20
TriggerPage*	1	1	0	0	0	1	-	3
Total								35

*AddrMask is set to 0xF6 to fire with all ASICs.

2.5. Continuous-Wave mode

Since Continuous-Wave mode runs continuously, there are no timing-critical operations present in its setup.

The following operations need to be performed to set up the ASIC for Continuous-Wave mode:

- ClearCore
- Output driver configuration: CONFIG_DRV (note different configuration than for default mode):
  - Disconnect and power down all ASIC output drivers from the cable that will be used for CW Transmit.
  - Connect all cables used for transmitting to 1 of the 4 CW phases.
  - Select one of two CW Receive modes:
    - Active:
      - Disconnect and power down all output drivers from the cable that will be used for receiving.
      - Connect all cables that will be used for receiving to 1 of 4 CW phases.
    - Passive:
      - Power up all output drivers used for receiving and connect their inputs to 1 of 4 CW phases.
- Assign every element to 1 of 4 CW phases by writing the CWSEL_DFF register.
- Enable all CW_EN bits by writing the CONFIG_DFF register.
- (Optional) Disable the ASIC clock: set internal clock divider (FPGA status memory address 0x0F) to 0.

It is important that the system does not send pulses into a cable that is connected to an output driver. Furthermore, the minimum and maximum allowed CW Transmit voltages are -0.5 V and 10 V.

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

Technical Report

3. Miscellaneous

3.1. Rx delay equalization

To diminish the dark pattern (seen at the center and certain steering angles), Rx delay equalization can be used. The registers (Section 2.3) must be set as follows:

- REG_BEAMTIMING (A.5): ExtRxSetup=1
- CONFIG_DFF (A.11): MATRIX_OFFSET_VAL=0..13 (where numbers equally spread over the groups, but randomly ordered). The following Matlab code is shown as an example for a probe with two USX chips:

```
MatrixOffsetVal=...
[12, 3, 5, 5, 1, 4,10, 5, 5, 3,10, 8, 6,13, 6, 2,...
7, 2, 7, 2, 3,10, 5, 9,12, 3,12,13, 7, 3, 0, 5,...
5, 1,12,12,13, 3, 0, 1, 4, 0,10, 1, 4, 8, 4, 7,...
0, 6, 8, 1, 9, 3, 4, 5, 7, 9, 7, 8,11,11, 4, 6,...
11,11, 5, 8,12,12, 4,13, 6,10, 4, 4, 7, 9, 1, 0,...
6, 9, 8, 6,12, 3, 7,13, 5,11,11,10, 8,13,10, 7,...
0, 7,10, 3, 2, 0,12,13, 2, 8, 9, 3, 0,10, 0, 0,...
4,11, 3, 9, 8,11, 0,13,10, 9, 3,11, 6, 4,12,12];
for i=1:128
    handles.asc.Parse(sprintf('SetParam:ConfigCore,MatrixOffsetVal(%g):%g',i-1,matrixOffsetVal(i)));
end
```

3.2. Delay resolution 5 ns

Normally, the Rx delay resolution is 20 ns (A.8 RXDELAY_DFF). In combination with REG_RXDYN (A.6) the Rx delay resolution can be decreased to 5 ns:

- StartPhase<0>=0, StartPhase<1>=1, StartPhase<2>=2, StartPhase<3>=3, StartPhase<4..7>=0.
- Multiplier<all>=0, MultiSign<all>=0.
- RX_ELEMENT_CLKSEL is set to 0—3 -> in this way, 0, 5, 10 or 15 ns delay is added to the RX_ELEMENTDELAY (which originally has 20 ns resolution).

3.3. Pulse cancellation

For pulse cancellation, two subsequent frames are used; the second pulse is the inverse of the first pulse. For pulse inversion, RX_ALWAYS_EN=1 cannot be used since cancellation is less than 20 dB. The following settings must be used:

- CONFIG_DFF.RX_ALWAYS_EN=0 (A.11).
- REG_CONFIG.BeamContTx=1 (A.2).

The pulses are set in XMITWF_DFF (Appendix A.4). There are two possibilities for pulse cancellation:

- First frame normal 100 ns pulse and second frame inverse 100 ns pulse: pulse cancellation of ~30 dB;
- Tripolar pulse, the second frame pulse is the inverse of the first frame pulse: pulse cancellation of ~40 dB.

The 100 ns pulse and its inverse are shown in Figure 1(a). For tripolar pulses (Figure 1(b)), the first frame consists of two different waveforms (Figure 1(c)), which are assigned in a checkerboard pattern to the elements. The second frame is the inverse of the first frame.
The schemes of the pulses are the following (rising↑, falling↓ edges; 1=half of elements, 2=all elements):

- Normal: 2↑ 2↓ ; inverted: 2↓ 2↑ .
- Tripolar1: 1↓ 2↑ 2↓ 1↑ ; tripolar2: 1↑ 2↓ 2↑ 1↓ .

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

Technical Report

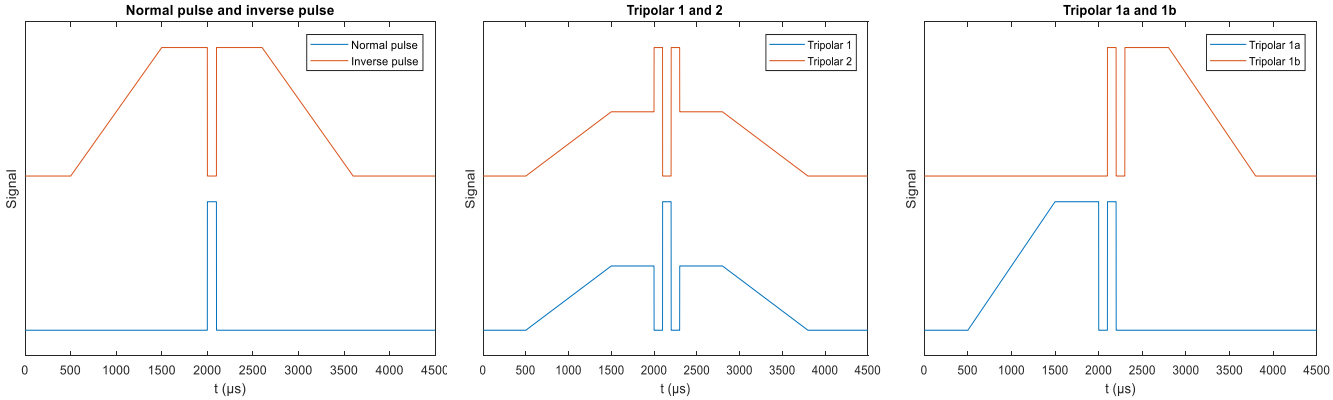

Figure 1. (a) Normal and inverse pulse; (b) tripolar pulses; (c) tripolar pulses 1a and 1b.

3.4. SELECT_DFF

Due to a silicon bug, SELECT_DFF does not work as intended.
For writing a certain group, per ASIC two steps must be followed:

1. Write to the first column (4 groups):
   SELECT_DFF=1 for groups (X,Y)=(0,0), (16,0), (32,0) and (48,0)
   SELECT_DFF=3 for groups that have to written
   SELECT_DFF=0 for other groups
   WriteChainSelect
   WriteChainCondXXX with new data
2. Pass the data from the first column to the other groups and write data to the first column
   WriteChainCondXXX with data for first column
   SELECT_DFF=0 for all groups
   WriteChainSelect

Note that data for WriteChainSelect and WriteChainCondXXX are scrambled (Section 7.2 ASIC spec, but only one column / 4 groups in total).

Example of Matlab scripts are given below.

3.4.1. Write data to single group

```
iGroup=2; % starts at 1
data_SingleGroup=GetRandomHex(nNibbles/2);
% Set all groups to bypass, 4 groups X=0 to no bypass, iGroup to bypass+takedata
CoreSelect_Group=repmat(SELECT_CORECHAIN_BYPASS,1,64);
CoreSelect_Group(0*16+1)=SELECT_CORECHAIN_NOBYPASS; % [X,Y]=[0,0]
CoreSelect_Group(1*16+1)=SELECT_CORECHAIN_NOBYPASS; % [X,Y]=[0,1]
CoreSelect_Group(2*16+1)=SELECT_CORECHAIN_NOBYPASS; % [X,Y]=[0,2]
CoreSelect_Group(3*16+1)=SELECT_CORECHAIN_NOBYPASS; % [X,Y]=[0,3]
CoreSelect_Group(iGroup)=SELECT_CORECHAIN_BYPASS_TAKEDATA;
selectDff=asicregisters.SELECT_DFF(CoreSelect_Group);
asicusb.LongCommandWrite(handles.ftd,'000021','16',asicusb.WriteChainSELECT,[selectDff,'00'],0);
% write data the first time
asicusb.LongCommandWrite(handles.ftd,'000021','16',writeChainCondCmd,repmat(data_SingleGroup,1,4),3); % Change data of
groups X=0
dataFirstColumn=repmat(' ',1,nNibbles*4);
for iGroupY=1:4
    dataFirstColumn((iGroupY-1)*nNibbles+(1:nNibbles))=xmitWfDffChain((iGroupY-1)*16*nNibbles+(1:nNibbles));
end
% write data the second time, setting groups X=0 to the 'old' values, since they are corrupted
```

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl


```
asicusb.LongCommandWrite(handles.ftd,'000021','16',writeChainCondCmd,dataFirstColumn,3); % Change data of other groups,
repair X=0
selectDff=asicregisters.SELECT_DFF(repmat(SELECT_CORECHAIN_NOBYPASS,1,64)); %  SELECT_CORECHAIN_NOBYPASS
asicusb.LongCommandWrite(handles.ftd,'000021','16',asicusb.WriteChainSELECT,[selectDff,'00'],0);
```

3.4.2. *Write same data to all groups*

```
data_SingleGroup=GetRandomHex(nNibbles/2);
% Set all groups to takedata, except X=0 -> to nobypass
CoreSelect_Group=repmat(SELECT_CORECHAIN_BYPASS_TAKEDATA,1,64);
CoreSelect_Group(0*16+1)=SELECT_CORECHAIN_NOBYPASS; % [X,Y]=[0,0]
CoreSelect_Group(1*16+1)=SELECT_CORECHAIN_NOBYPASS; % [X,Y]=[0,1]
CoreSelect_Group(2*16+1)=SELECT_CORECHAIN_NOBYPASS; % [X,Y]=[0,2]
CoreSelect_Group(3*16+1)=SELECT_CORECHAIN_NOBYPASS; % [X,Y]=[0,3]
selectDff=asicregisters.SELECT_DFF(CoreSelect_Group);
asicusb.LongCommandWrite(handles.ftd,'000021','16',asicusb.WriteChainSELECT,[selectDff,'00'],0);
asicusb.LongCommandWrite(handles.ftd,'000021','16',writeChainCondCmd,repmat(data_SingleGroup,1,4),3); % Change data of
groups X=0
asicusb.LongCommandWrite(handles.ftd,'000021','16',writeChainCondCmd,repmat(data_SingleGroup,1,4),3); % Change data of
other groups
selectDff=asicregisters.SELECT_DFF(repmat(SELECT_CORECHAIN_NOBYPASS,1,64)); %  SELECT_CORECHAIN_NOBYPASS
asicusb.LongCommandWrite(handles.ftd,'000021','16',asicusb.WriteChainSELECT,[selectDff,'00'],0);
```

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

Technical Report

4. Document revisions

Version	
B	PLL clock divider
C	Aperture reduction: use CWSEL_DFF (element disable) instead of REG_CERXTXCONF
D	A.11: CONFIG_DFF.ANALOG_RESET_ATNOTRX must bet set to 0
E	2.4. Send memory size corrected
F	2.2 After changing PLL CLK, wait 2 ms 2.2 LED disable 2.2 Trigger off 3.4 Example for SELECT_DFF A.5 and A.11: LNA_AUTO_POWERDOWN must be 0. To save power: LNA_AUTO_POWERDOWN=1 and RunTxTime>=100. A.14 SELECT_DFF added

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

Technical Report

A. Registers

Three types of registers are present in the ASIC:

- The periphery registers with “REG_” have one instance per ASIC.
- The core registers with “_DFF” have one instance per group. Usually, a block of 64 times the register is written to an ASIC. They are reordered (“scrambled” in the ASIC specification) and a dummy byte is put behind.
- The output driver (CONFIG_DRV) is also reordered (in a different manner than “_DFF”) and a dummy byte is put behind.

A.1. REG_BIST

Built-in self-test

Address	Bits	Name	Default value
<7:0>	8	ClearTime	0
<15:8>	8	ConfigIn	0
<16>	1	DOUTAutoDis	0
<17>	1	DOUTAlwaysEn	0
<21:18>	4	DataOutClkDiv	9*
<23:22>	2	iVal	0
<24>	1	ErrorGetVersion	0
<28:25>	4	IbiasCal	0**
<29>	1	BistEn	0
<31>	1	FuseWriteEn	0
<39:32>	8	Reserved	0

** DataOutClkDiv must match FPGA parameter “Divider for reading data from ASIC” (Section 2.2).*
*** Use the value obtained by the BistGetFuseMemory command.*

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

A.2. REG_CONFIG

Various configuration parameters

Address	Bits	Name	Default value
<7:0>	8	TxClkDivider	20
<10:8>	3	TXDELAY_SHIFT	USX2: ASIC1='010', ASIC2='100'
<12:11>	2	TXDELAY_LSBOPT	0
<14:13>	2	TXDELAY_LSBRAND	0
<15>	1	RXDELAY_OFFSET	0
<18:16>	3	RXDELAY_SHIFT	USX2: ASIC1='010', ASIC2='100'
<20:19>	2	RXDELAY_LSBOPT	0
<23:21>	2	RXDELAY_LSBRAND	0
<24>	1	UseAnalogReset	1
<25>	1	RandomizeDynUpdate	0
<26>	1	BeamRunRxCalc	0 (non-compr), 1 (compr)
<27>	1	BeamRunTxCalc	0 (non-compr), 1 (compr)
<28>	1	BeamRxClkInit	1
<29>	1	BeamContRx	0
<30>	1	BeamContTx	0*
<31>	1	BeamWaitRun	0
<39:32>	8	Error_NMask_Out	128
<47:40>	8	Error_NMask_XMit	0
<48>	1	LockDuringBeam	0
<52:49>	4	Error_Stopmask	0
<53>	1	BeamTxAutoStop	0
<55:54>	2	TxClk_Config	0

** BeamContTx=1 for pulse cancellation (Section 3.3).*

A.3. REG_CERXTXCONF

Receive/transmit element enable for aperture definition

Due to a silicon bug, this register does not work as intended. Leave it on the default values and use the register CWSEL_DFF instead to disable elements.

Address	Bits	Name	Default value
<5:0>	6	RX_XMin	0
<13:8>	6	RX_XMax	63
<19:16>	4	RX_YMin	0
<27:24>	4	RX_YMax	15
<37:32>	6	TX_XMin	0
<45:40>	6	TX_XMax	63
<51:48>	4	TX_YMin	0
<59:56>	4	TX_YMax	15
<63>	1	n/a	Silicon bug
<39:32>	8	TX_MinVal	Silicon bug, not available
<47:40>	8	n/a	Silicon bug

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

Technical Report

A.4. XMITWF_DFF

Transmit waveform setup

XMITWF_DFF exists for every group in the ASIC, thus 8*64 +1=513 bytes are sent per ASIC

In this example, a single pulse of 100 ns is used:

Address	Bits	Name	100 ns
<3:0>	4	TX_STARTSAMPLE	6
<7:4>	4	TX_REPSAMPLE	7
<11:8>	4	TX_ENDSAMPLE	14
<13:12>	2	TX_PRECHARGE	‘10’
<19:14>	6	TX_PATTERNREP	0
<24:20>	5	TX_CLAMPTIME	8
<28:25>	4	TX_PRECHARGETIME	15
<26:25>	2	TX_PRECHARGE_RANDOM_CONF	0
<31:29>	3	TX_CHSEL_PATTERN	0
<63:32>	32	TX_WF_SETUP	0x003FC000

0x003FC000 = ‘00 00 00 00 00 00 00 11 11 11 11 00 00 00 00 00’ (LSB first)

As a second example, pulse cancellation is used; two pulses of opposite polarity are constructed. Normal and inverse pulse of 100 ns give a pulse cancellation of ~30 dB (if RX_ALWAYS_EN=0 and BeamContTx=1). Tripolar pulse and its inverse give a pulse cancellation of ~40 dB.

For pulse cancellation, the following settings are used:

Address	Name	100 ns pulse cancel		Tripolar	
		1st frame	2nd frame	1st frame	2nd frame
<3:0>	TX_STARTSAMPLE	0	0	0	0
<7:4>	TX_REPSAMPLE	4	4	4	4
<11:8>	TX_ENDSAMPLE	11	11	11	11
<13:12>	TX_PRECHARGE	‘10’	‘10’	‘10’	‘10’
<19:14>	TX_PATTERNREP	0	0	0	0
<24:20>	TX_CLAMPTIME	8	8	8	8
<28:25>	TX_PRECHARGETIME	15	15	15	15
<26:25>	TX_PRECHARGE_RANDOM_CONF	0	0	0	0
<31:29>	TX_CHSEL_PATTERN	0	1	2	2
<63:32>	TX_WF_SETUP	0x555AA555	0x555AA555	0xA00FF005	0xAFF00FF5

0x555AA555 = ‘10 10 10 10 10 10 01 01 01 01 10 10 10 10 10 10’ (LSB first)
0xA00FF005 = ‘10 10 00 00 00 00 11 11 11 11 00 00 00 00 01 01’ (LSB first)
0xAFF00FF5 = ‘10 10 11 11 11 11 00 00 00 00 11 11 11 11 01 01’ (LSB first)

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

Technical Report

A.5. REG_BEAMTIMING

Timing configuration for transmit and receive

Address	Bits	Name	Default value
<7:0>	8	AnaResetStopTime	2
<15:8>	8	SetupRxTime	0
<23:16>	8	RunTxTime	0***
<31:24>	8	RunRxTime	29*
<39:32>	8	StopTxTime	255
<47:40>	8	StopRxTime	74
<51:48>	4	TGCIncrement	6
<55:52>	4	TGCStart	15
<59:56>	4	TGCDecrement	3
<62:60>	3	TGCDecrementCnt	0
<63>	1	ExtRxSetup	0**
<64>	1	ExtTxRunTime	0
<65>	1	ExtRxSetupTime	0
<71:66>	6	-	Reserved

** RunRxTime must be >=SetupRxTime+29*
*** For Rx delay equalization (diminishing dark pattern): ExtRxSetup=1 (Section 3.1.). In addition, MATRIX_OFFSET_VAL must be used (See A.11).*
*** If LNA_AUTO_POWERDOWN=1, RunTxTime must be >=100. Note that RunRxTime could also be increased to >=100

A.6. REG_RXDYN

Dynamic receive

Address	Bits	Name	Default value
<2:0>	3	Multiplier<0>	7*
<3>	1	MultiSign<0>	1*
<6:4>	3	StartPhase<0>	4
..			
<58:56>	3	Multiplier<7>	7*
<59>	1	MultiSign<7>	0*
<62:60>	3	StartPhase<7>	4
<67:64>	4	Duration<0>	0**
..			
<95:92>	4	Duration<7>	0**
<103:96>	8	Slope<0>	0**
..			
<159:152>	8	Slope<7>	0**
<161:160>	2	RandomOpt	0
<167:162>	6	-	Reserved

**Multiplier[0..7]=[7,5,3,1,1,3,5,7]; MultiSign[0..7]=[1,1,1,1,0,0,0,0]*
***Duration and slope are 0 for static Rx; need to be optimized if dynamic Rx*

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

A.7. TXDELAY_DFF

Transmit delays (non-compressed)

TXDELAY_DFF exists for every group in the ASIC, thus 11*64 +1=705 bytes are sent per ASIC. Example below is for plane wave (all Tx delays=0).

Address	Bits	Name	Default value
<4:0>	5	TX_ELEMENTDELAY<0>	0
..			
<79:75>	5	TX_ELEMENTDELAY<15>	0
<87:80>	8	TX_GROUP_STARTCOUNTER	0

A.8. RXDELAY_DFF

Receive delays (non-compressed)

RXDELAY_DFF exists for every group in the ASIC, thus 16*64 +1=1025 bytes are sent per ASIC. Example below is for static Rx delays.

Address	Bits	Name	Default value
<4:0>	5	RX_ELEMENTDELAY<0>	14
<7:5>	3	RX_ELEMENT_CLKSEL<0>	0
..			
<124:120>	5	RX_ELEMENTDELAY<15>	14
<127:125>	3	RX_ELEMENT_CLKSEL<15>	0

A.9. REG_TXDELAYP0,1

Transmit delays (compressed)

Example values are for focus 50 mm, steering (α_x,α_y)=(0°,0°).

Address	Bits	Name	Value
<7:0>	8 U	C0	19
<15:8>	8 S	C1	0
<23:16>	8 S	C2	-18
<31:24>	8 S	C3	0
<39:32>	8 S	C4	0
<47:40>	8 S	C5	-16
<55:48>	8 S	C6	-1
<63:56>	8 S	C7	0

S=signed; U=unsigned

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

Technical Report

A.10. REG_RXDELAYP0,1

Receive delays (compressed)

Example values are for focus 50 mm, steering (α_x,α_y)=(30°,0°).

Address	Bits	Name	Value
<7:0>	8 U	C0	1
<15:8>	8 S	C1	-50
<23:16>	8 S	C2	-11
<31:24>	8 S	C3	0
<39:32>	8 S	C4	-64
<47:40>	8 S	C5	0
<55:48>	8 S	C6	0
<63:56>	8 S	C7	48
<71:64>	8 S	C8	0
<79:72>	8 S	C9	8
<127:80>	48	-	Reserved

S=signed; U=unsigned

A.11. CONFIG_DFF

Analog group configuration

CONFIG_DFF exists for every group in the ASIC, thus 5*64 +1=321 bytes are sent per ASIC.

Address	Bits	Name	Value
<3:0>	4	ISEL_LNA	4
<7:4>	4	ISEL_ODRV	3
<11:8>	4	ISEL_RESCNTL	6
<15:12>	4	ISEL_DCGND	8
<19:16>	4	ISEL_PCHVP	8
<20>	1	BIAS_EN	1
<21>	1	ODRV_EN	1
<22>	1	RX_ALWAYS_EN	1*
<23>	1	CW_EN	0***
<24>	1	LNA_EN	1
<25>	1	RES_CNTL_OVERWRITE	0
<28:26>	3	RES_CAL	4
<29>	1	ANALOG_RESET_ATNOTRX	0**
<30>	1	AUTO_RESET	1
<31>	1	LNA_AUTO_POWERDOWN	0*****
<32>	1	TX_PRECHARGE_RANDOM_ISRC	0
<33>	1	RANDOM_REG_UPD	0
<37:34>	4	MATRIX_OFFSET_VAL	0****
<38>	1	PCHVP_PS_EN	0
<39>	1	DISABLE_DISABLE	0

** Due to a silicon bug in ASIC-v2, RX_ALWAYS_EN needs to be set to 1 for normal B-mode. For pulse cancellation, RX_ALWAYS_EN=0, BeamContTx=1 (Section 3.3).*
*** Due to the same silicon bug, ANALOG_RESET_ATNOTRX must be set to 0.*
**** For CW mode: set CW_EN=1.*
***** In combination with ExtRxSetup=1 (see A.5), MATRIX_OFFSET_VAL is used for Rx delay equalization (reducing dark pattern; see Section 3.1).*
****** To consume less power, LNA_AUTO_POWER_DOWN=1 and RunTxTime>=0*

Oldelft B.V.
P.O.Box 5082, 2600 GB Delft
Elektronicaweg 10, 2628 XG Delft
The Netherlands
Tel. +31 15 26 98 900
Fax +31 15 26 98 954
www.oldelft.nl

A.12. CWSEL_DFF

CW configuration or element disable

If CONFIG_DFF.CW_EN=1:

Address	Bits	Name	Value
<1:0>	2	CW_SEL_ELE<0>	0—3
<2>	1	TX_GND_CW<0>	0
<3>	1	TX_HVP_CW<0>	0
..			
<61:60>	2	CW_SEL_ELE<15>	0—3
<62>	1	TX_GND_CW<15>	0
<63>	1	TX_HVP_CW<15>	0

If CONFIG_DFF.CW_EN=0 and CONFIG_DFF.DISABLE_DISABLE=0:

Address	Bits	Name	Value
<0>	1	Rx disable	0
<1>	1	Tx disable	0
<3:2>	2	Reserved	0
..			
<60>	1	Rx disable	0
<61>	1	Tx disable	0
<63:62>	2	Reserved	0

A.13. CONFIG_DRV

Output driver configuration

Address	Bits	Name	Normal	CW transmit	CW receive passive	CW receive active
<0>	1	DRV_ENABLE	1	0	0	0
<3:1>	3	DRV_BIASSEL	4	0	0	4
<7:4>	4	DRV_SELINCW	0	0	0	‘0001’—‘1000’
<11:8>	4	DRV_SELOUTCW	0	‘0001’—‘1000’	‘0001’—‘1000’	0
<12>	1	DRV_OUTCON	1	0	0	1
<13>	1	DRV_SELINNOCW	1	0	0	0
<14>	1	DRV_RES_EN	0	0	0	1
<15>	1	DRV_FF_nFB	0	0	0	0

A.14. SELECT_DFF

Conditional write selection

Address	Bits	Name	Value
<1:0>	2	Write conditional select	0