“Neuropix”

Phase IIIA System And Probe User Manual

Version: July 28, 2016

Number:

Issued by: Imec

Title Neuropix Phase IIIA System and Probe User Manual

Author(s) Jan Putzeys

Reviewed by Silke Musa

Revision V1.9

Date July 28, 2016

Keywords: Neural probe, neural recording system

Revision history

Version	Date	Description	Responsible
V1	January 20, 2016	Initial release	Jan Putzeys
V1.1	February 02, 2016	Chapter 2.4 Figure SW13 replaced Chapter 4.1 IP address typo corrected Chapter 5.1 and 5.4: updates on LEDs.	Jan Putzeys
V1.2	March 23, 2016	Added information on probe description Added electrode selectivity and reference selection Updates on start of neural recording Updates on BISTs Added description of probe handling and cleaning Added description of gain/noise measurements	Silke Musa, Jan Putzeys
V1.3	April 11, 2016	API version 3.5, chapter 5.2 updated	Jan Putzeys
V1.4	April 20, 2016	Trigger and synchronization enabled ZIF connector pinout described	Jan Putzeys
V1.5	April 25, 2016	ADC calibration and gain correction separated (chapter 5.2)	Jan Putzeys
V1.6	April 28, 2016	Loading ADC and gain correction from csv files (chapter 5.2)	Jan Putzeys
V1.7	May 13, 2016	Added information on start trigger measurements (chapter 5.11) Chapter 2.2: LED disable	Jan Putzeys
V1.8	May 24, 2016	Bug removed in API version number (chapter 1)	Jan Putzeys
V1.9	July 28, 2016	API version updated	Jan Putzeys

Related documents:

- *Xilinx: UG810 - KC705 Evaluation Board for the Kintex-7 FPGA User Guide*
- *2016-07-28 Neuropix Phase IIIA system user API V1.12.docx*
- 2016-01-20_Electrode_mapping.xlsx

Contact address data: Kapeldreef 75, 3001 Heverlee

Contents

List of abbreviations

AP	Action Potential
LFP	Local Field Potential
ASIC	Application Specific Integrated Circuit
HS	Headstage
BSC	Base Station Connect (board)
SR	Shift Register
FPGA	Field Programmable Gate Array
FPGA dev	FPGA development (board)
PCB	Printed Circuit Board
BIST	Built-In Self-Test
API	Application Programming Interface
ADC	Analog-to-Digital Converter
IC	Integrated Circuit
EEPROM	Electrically Erasable Programmable Read-Only Memory
SDRAM	Synchronous Dynamic Random Access Memory
LED	Light Emitting Diode
FMC	FPGA Mezzanine Card
SMA	SubMiniature version A
JTAG	Joint Test Action Group
GPIO	General Purpose Input/Output
ESD	Electro-Static Discharge
SPI	Serial Peripheral Interface
PBS	Phosphate buffered saline
MOPS	3-(N-morpholino)propanesulfonic acid
MES	2-(N-morpholino)ethanesulfonic acid

List of symbols

Definition of Technical Terms

- **Probe Shank:** Implanted part of the probe.
- **Probe Base:** Non-implanted part of the probe.
- **Probe Flex:** Flexible carrier PCB on which the probe is assembled
- **Headstage (HS)**: Miniature board located close to the probe, which configures and calibrates the probe and serializes probe data.
- **Basestation connect (BSC) board:** Small board with deserializer chip, which connects to the FPGA development board.
- **FPGA development board**: Commercial KC705 FPGA development board (http://www.xilinx.com/products/boards-and-kits/ek-k7-kc705-g.html)

1 Scope

This document describes the Neuropix Phase IIIA recording system and provides guidelines on how to use the Phase IIIA probes with the system.

This manual is valid for the following versions of hardware and API driver:

- HS PCB version 00-00-03
- HS FPGA code version V00-R02-S03, V00-R03-S04 & V00-R04-S00. The LED disable functionality is only valid for V00-R04-S00.
- BSC connect board PCB version 00-02-01
- BS FPGA code version 4.1 & 4.2
- API version 4.2 & V 4.3

2 Hardware description

An overview of the Neuropix Phase IIIA system (excl. commercial FPGA dev board) is shown in Figure 1. It consists of a flex PCB with the neural probe, the headstage, a dual-microcoax cable, and the basestation connect board.

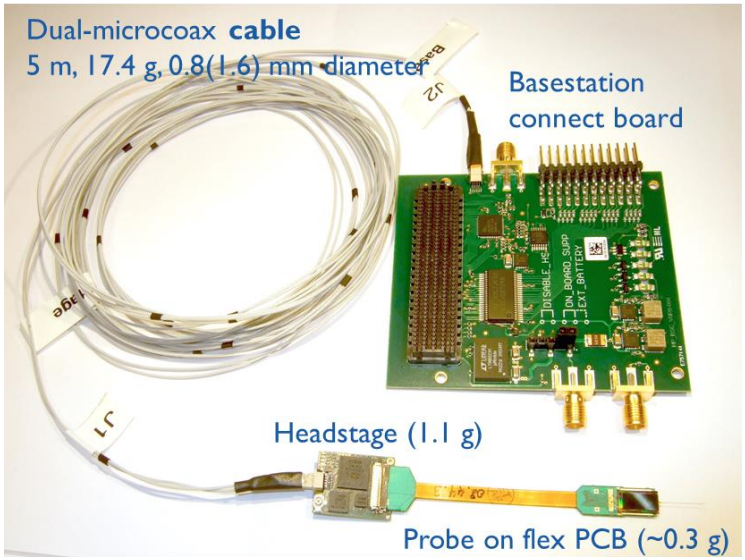

Figure 1: Neuropix Phase IIIA system (excl. FPGA dev board).

2.1 Flex PCB and probe

The probe is assembled on a flexible polyimide PCB. Other components assembled on the backside of the flex under the probe are passives for biasing and decoupling, and an IC generating a low-noise voltage reference. Indicated in blue are the vias for soldering REF and GND wires. These can

be shorted during experiments. On the backside, the 2 square pads next REF and GND vias and the 1 large rectangular pad are connected to GND.

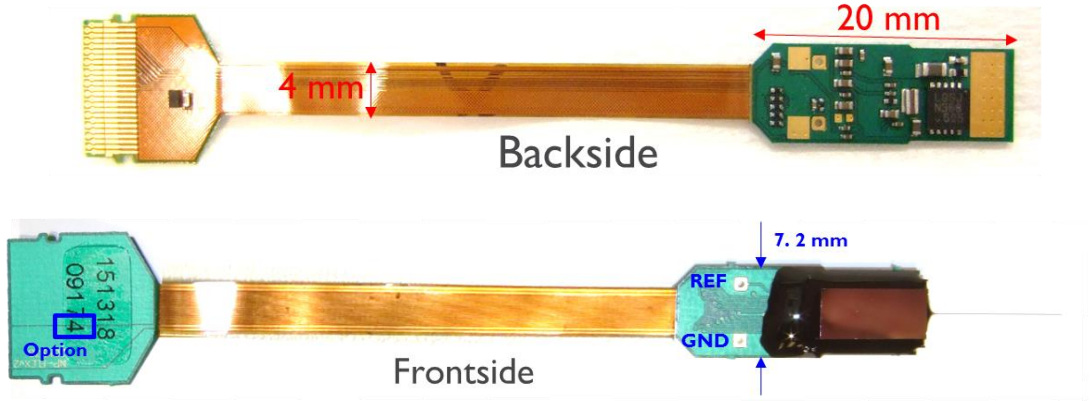

Figure 2: Flex PCB with assembled components: top: backside, bottom: frontside (with probe).

At the other end of the flex, an EEPROM is assembled, which contains the probe ID code and gain and ADC calibration parameters. The 11-digit probe ID code is printed on the frontside. The last digit indicates the probe Option (1-4). The EEPROM contains the same code except of the first digit (1). Additional info on how to call the probe ID via the API can be found in the document *2016-07-28 Neuropix Phase IIIA system user API V1.12.* The full assembly including probe weighs ~0.3 g. The flex PCB dimensions are indicated in Figure 2.

The probe base is covered with a 300 µm thick Si spacer and encapsulated in black biocompatible epoxy (Masterbond, EP42HT-2MED Black) (Figure 3). Additional epoxy is also applied to the probe neck and PCB backside (Figure 4). Both Si spacer and black epoxy serve as light shields. The assembly including epoxy and backside components has a thickness of ~2.2 mm (Figure 5). The probe dissipates a power of ~30 mW (in the base).

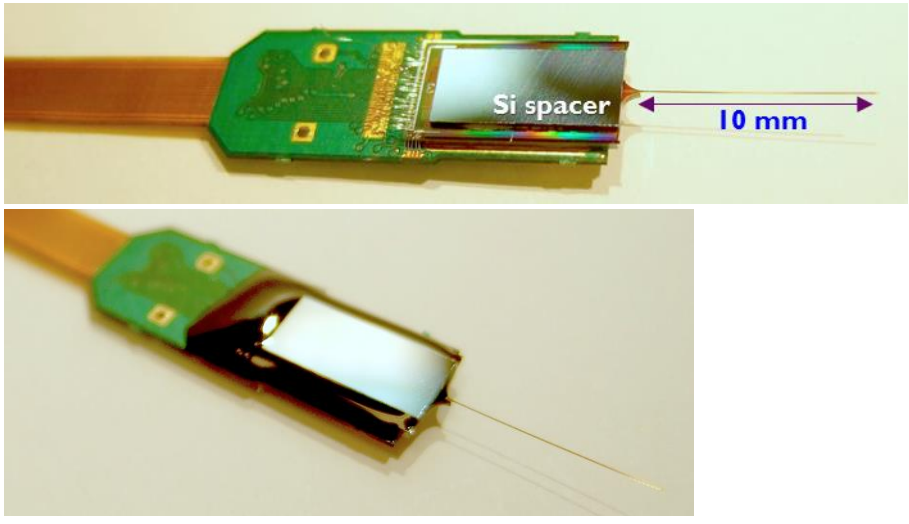

Figure 3: Top: Wirebonded probe on flex with Si spacer glued on top of the base. Bottom: Probe encapsulated with black epoxy.

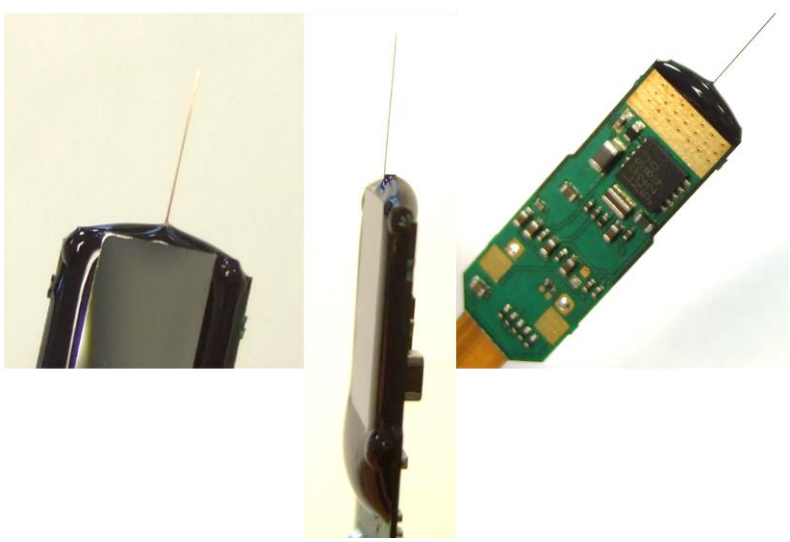

Figure 4: Epoxy encapsulation to seal the bondwires and to act as a light-shield.

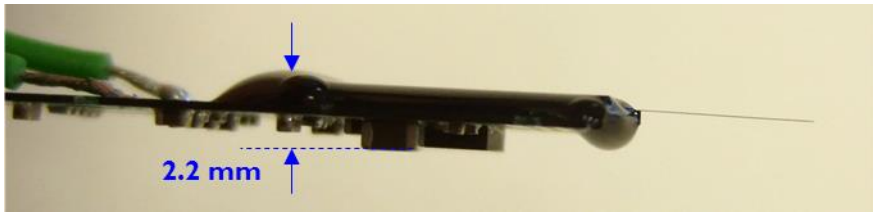

Figure 5: Thickness of the probe assembly including epoxy and backside components.

The electrode layout of the four shank Options is shown in Figure 6.

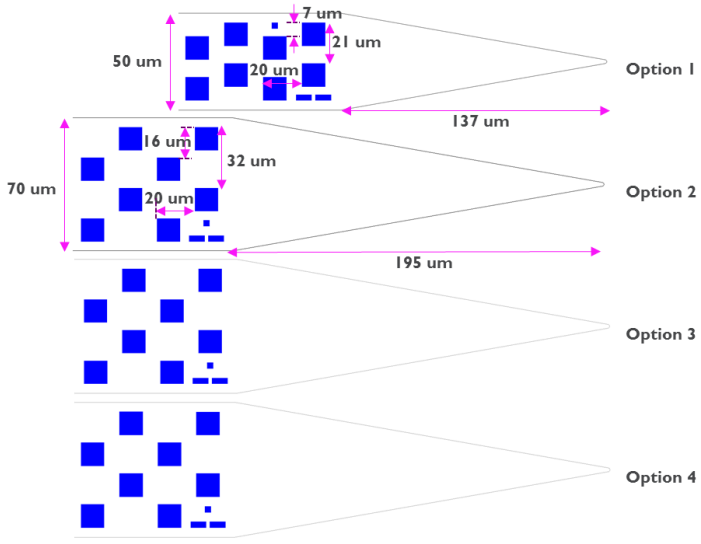

Figure 6: Electrode layout of the four shank Options.

On the probe shank, Morse-code-like symbols indicate the electrode numbers (Figure 7).

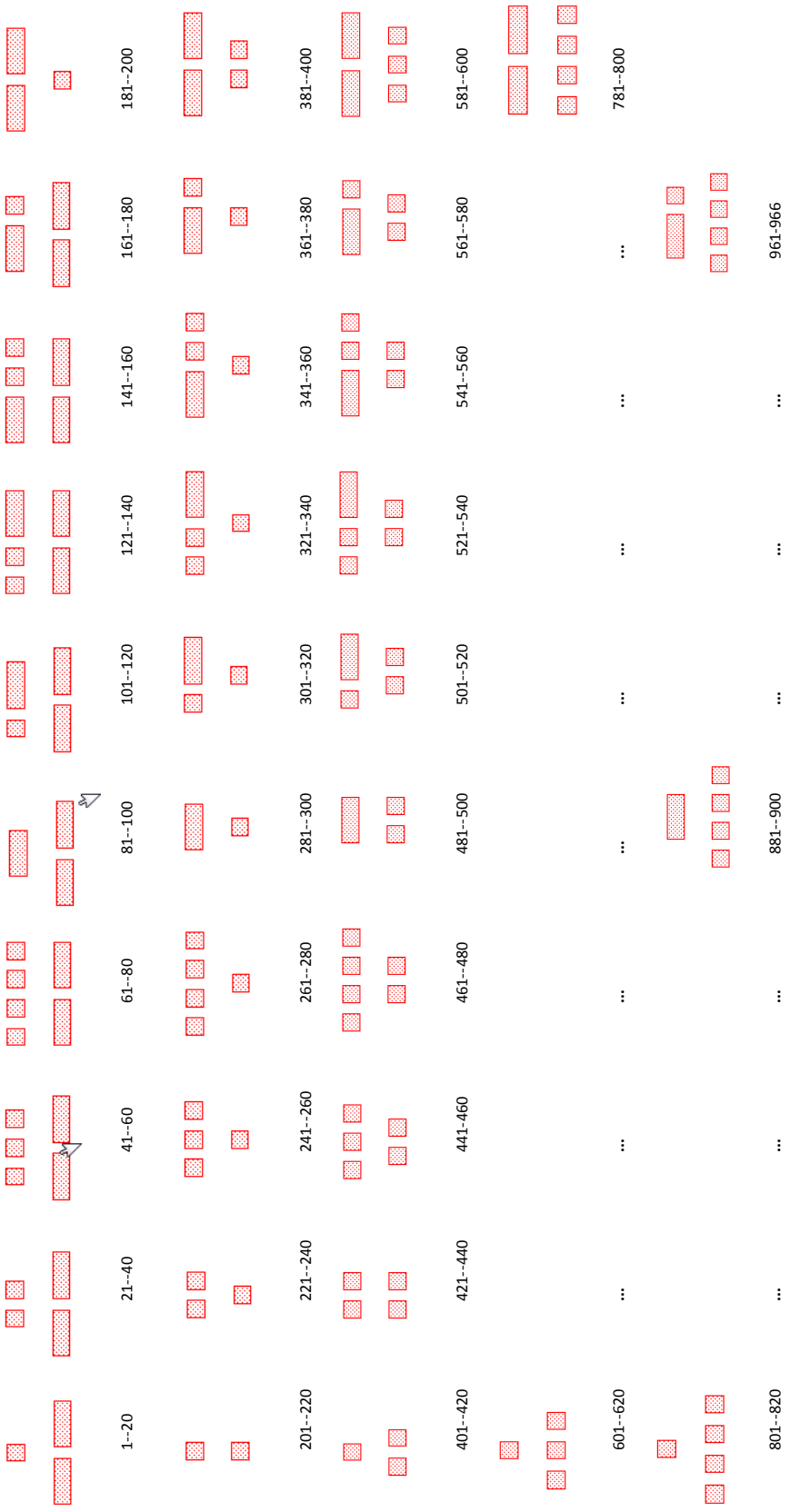

Figure 7: Morse-code-like symbols indicating electrode numbers.

2.2 Headstage

A picture of the HS including dimensions is shown below. The HS weighs 1.1 g.

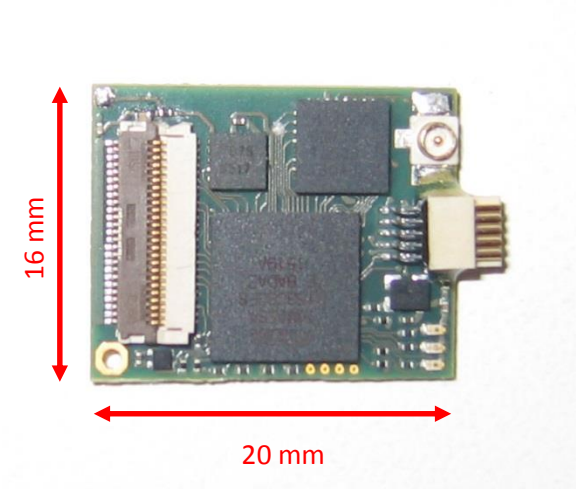

Figure 8: Headstage.

The HS contains a number of LDOs for power supply, a serializer chip for communication to/from the BSC board, and an FPGA for probe control and data transfer from probe to serializer. The probe flex connects to the 50-pin FH29B ZIF connector. The 10-pin Omnetics connector connects to the microcoax cable. The power consumption of the HS is ~500 mW.

Three LEDs are provided on the HS: blue, orange, green-yellow. These LEDs are used for verification of the functionality of the recording system via the BIST.

- Blue: used for BIST#1: The LED is continuously blinking at a rate of 2Hz. It indicates that the HS receives power and the FPGA is successfully configured.
- Orange: used for BIST #5 and BIST #8.
- Green-yellow: used for BIST#3 and start trigger.

The LEDs can be disabled via an API function (neuropix_ledOff).

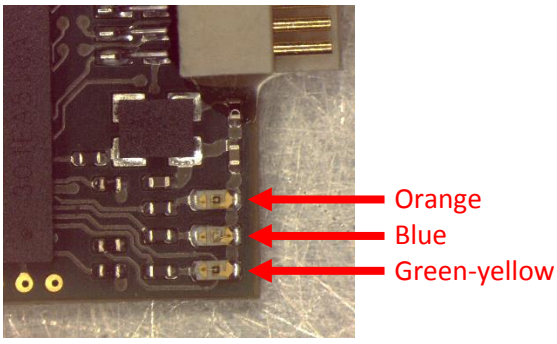

Figure 9: Headstage LED colours.

The pinout of the ZIF connector is given in the picture below. Pins 1-25 are not accessible when the probe flex is plugged in. Pins 26-50 are difficult to access via the PCB pads, but can be accessed via the pins on top of the ZIF.

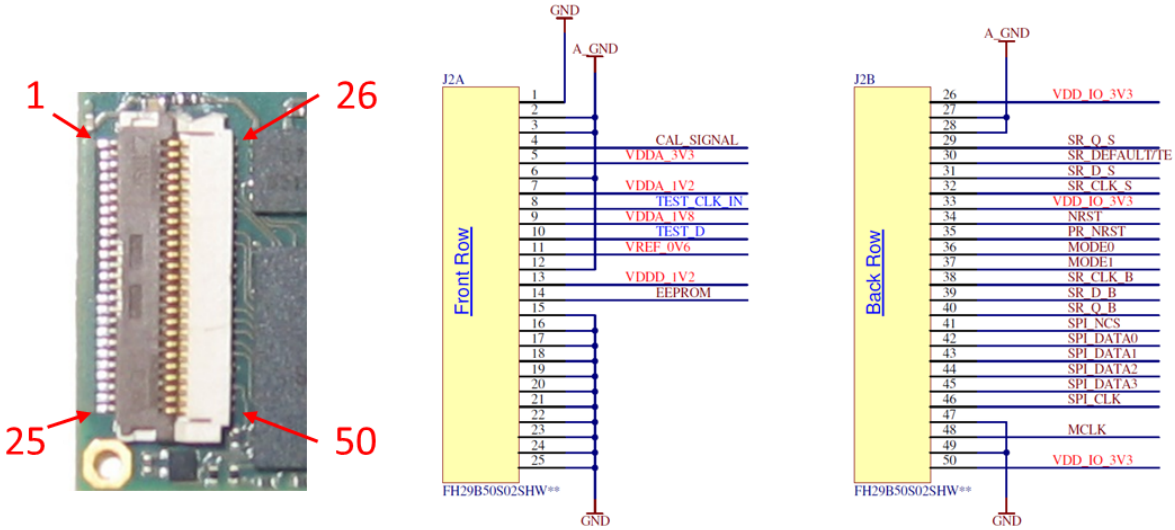

Figure 10: Flex ZIF connector pinout and connector pin numbering.

2.3 Microcoax cable

A cable consisting of two 0.81 mm thick microcoax strands provides power from the BSC board to the HS and transfers control/neural data to/from the HS.

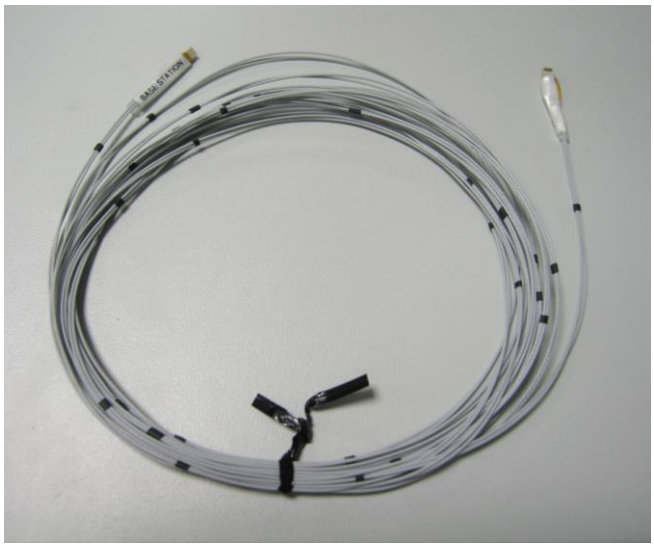

Figure 11: Microcoax cable.

Both ends of the microcoax cable are terminated with Omnetics connectors. The side tagged ‘basestation’ connects to the BSC board and is terminated with an 8-pin Omnetics connector. The side tagged ‘headstage’ connects to the HS and is terminated with a 10-pin Omnetics connector. The microcoax cable is 5 m long and weighs 17.4 g.

Sharp bending of the cable can degrade cable quality and signal integrity and must therefore be avoided. Avoid a bending radius ≤ 2 cm.

2.4 Basestation connect board

The BSC board provides an interface between the microcoax cable and the FPGA dev board. It contains some power supply components and a deserializer chip for bidirectional communication with the HS over the microcoax cable.

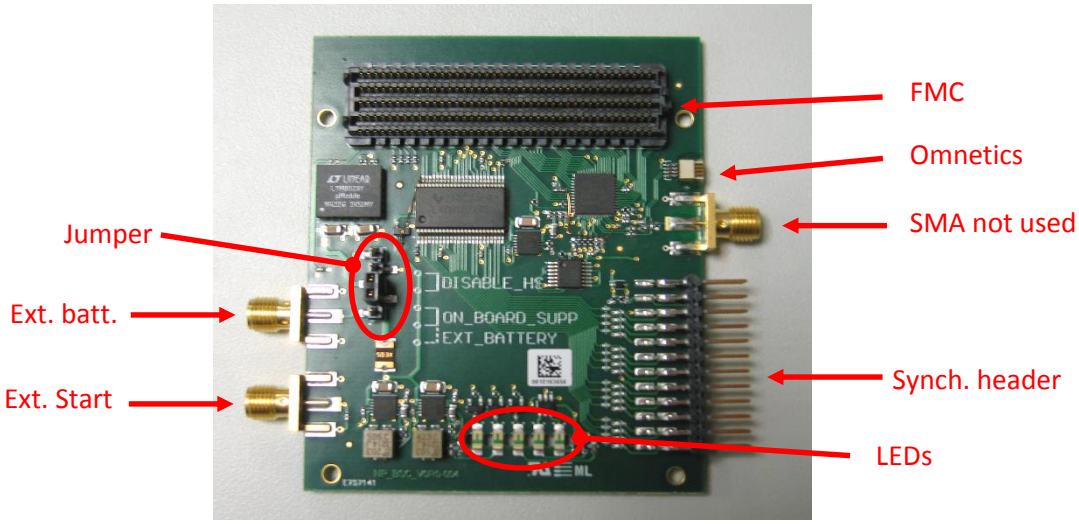

Figure 12: Basestation connect board.

The serializer/deserializer chipset uses a Hamming code for error detection and correction. The error status bits in registers on the deserializer are monitored by the FPGA dev board.

2.4.1 *Connectors*

The following connectors are provided on the BSC board:

1. A mating 400-pin FMC connector for the FMC connector (J22) on the FPGA dev board.
2. A mating 8-pin Omnetics connector for the microcoax cable.
3. A start trigger I/O SMA: (Ext. Start). This connector can be used to connect a digital trigger signal to the neural recording system. Via the API it can be configured as input or output. If configured as an output signal, the Ext. Start signal is a 3.3V pulse signal. Connect a high impedance load to the signal. If configured as an input, connect a 5V logic signal to the connector. The Ext. Start signal is also available on the Synch. connector.

4. A 24-pin header connector (Synch.): A 16-bit parallel signal can be connected to this connector and is recorded together with the neural data stream. Connect 5V logic signals to the connector (low level treshold = 1.5V; high level treshold = 3.5V). The pinout of the header is indicated in the picture and table below.
5. An SMA connector for connecting an external power supply (Ext. batt.). This connector is currently not used.
6. An SMA connector labelled 'To Headstage': Not used.

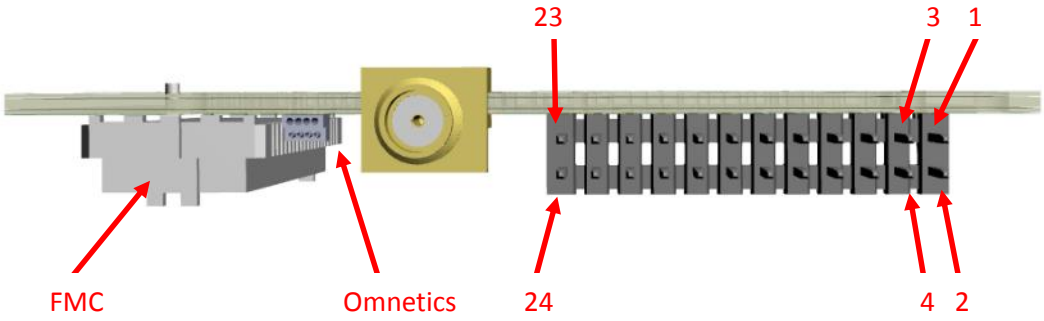

Figure 13: Synch. header pin numbering.

Table 1: Synch. header pinout.

Pin number	Connection	Pin number	Connection	Pin number	Connection
1	Sync_0	9	Sync_7	17	Sync_13
2	Sync_1	10	VSS	18	Sync_14
3	Sync_2	11	Sync_8	19	Sync_15
4	Sync_3	12	Sync_9	20	VSS
5	VSS	13	Sync_10	21	NC
6	Sync_4	14	Sync_11	22	EXT_START
7	Sync_5	15	VSS	23	VSS
8	Sync_6	16	Sync_12	24	VSS

2.4.2 *LEDs*

A number of status LEDs are provided on the BSC board:

- D1: DES_LOCK (red): Indicates an error in communication between BSC board and HS.
- D2: SW_HBEAT (red): Not used.
- D3: NEUR_HBEAT (blue): Used for BIST #2.
- D4: NEUR_ENABLE (green): Indicates when the recording system is initialized to transmit neural data.
- D6: DES_ERROR (red): Indicates an error in communication between BSC board and HS.

2.4.3 *Jumper*

One 5-pin header controls the power supply configuration of the board. The jumper must remain in its default position: ON_BOARD_SUPP; as shown on the picture below.

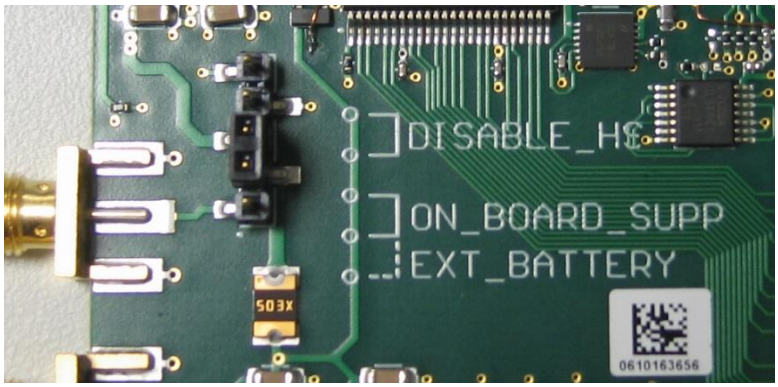

Figure 14: BSC board default power supply jumper default position.

2.5 FPGA development board

Below is a picture of the commercial KC705 FPGA dev board.

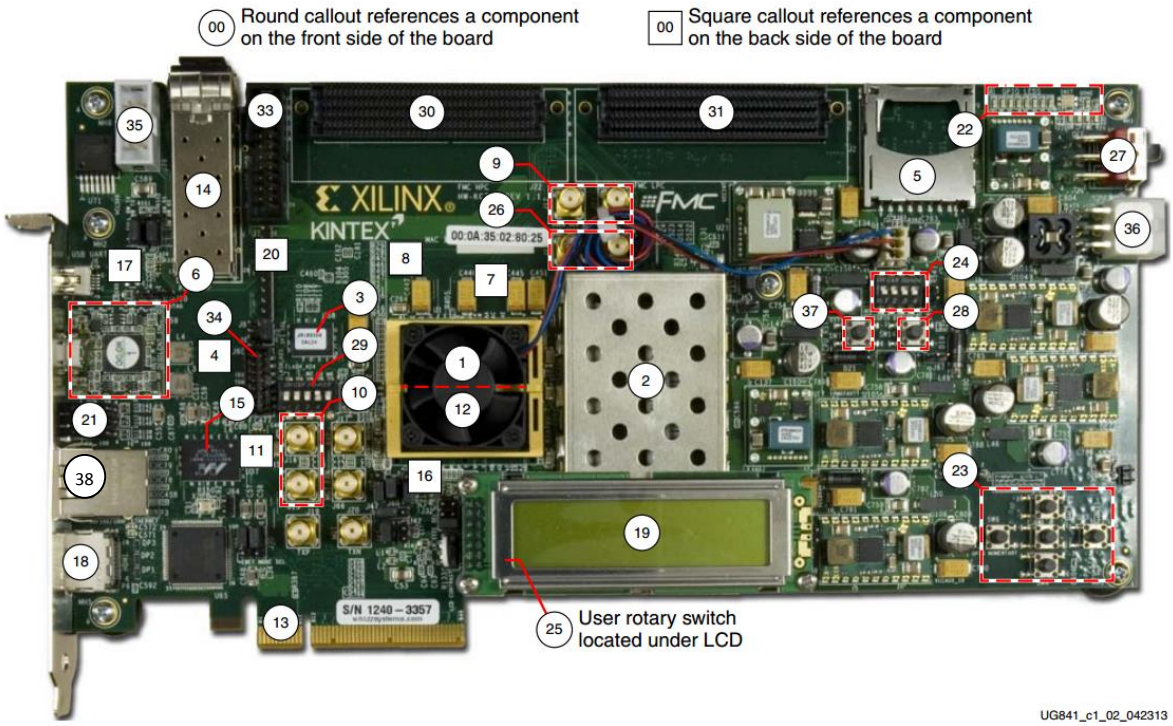

Figure 15: KC705 board components.

Only the components important for the Neuropix application are described in the following.

Connectors:

- 30: FMC connector (J22): Used to connect the BSC board.
- 36: power supply connector (J49): Used with the power supply adaptor delivered with the KC705 board.
- 6: USB JTAG module and USB connector (J60): The USB interface is used for writing the FPGA boot code bitstream to the KC705 FPGA or to the configuration memory.
- 38: Ethernet RJ45 connector (P3): A TCP/IP interface port is used to connect the recording system to the PC, for probe control and data transfer.

LEDs:

- 22: A number of status and user LEDs are provided on the KC705 board. The GPIO user LEDs are used by the Neuropix application for indicating the status of the recording system.

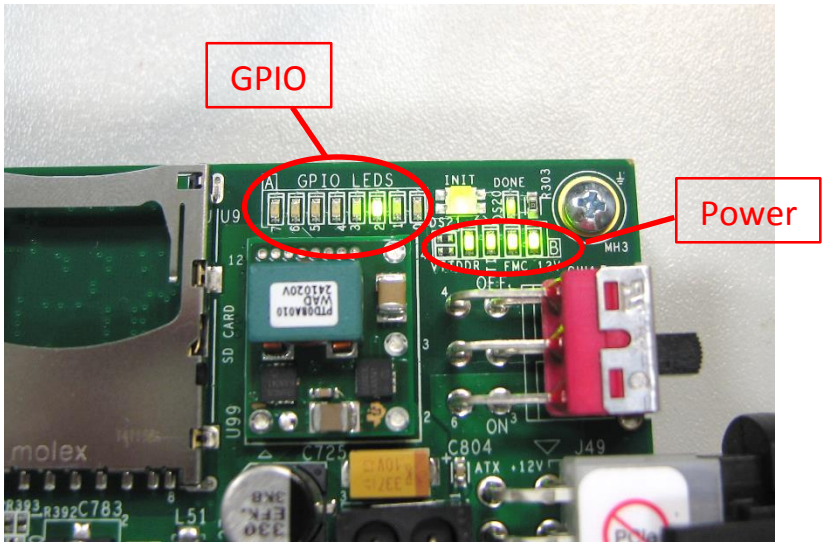

Figure 16: LEDs on FPGA dev board.

  - A: GPIO LEDs 0 to 7:
    - LED0: status of data link with PC over TCP/IP
    - LED1: status of configuration link with PC over TCP/IP
    - LED2: status of DRAM controller
    - LED3: SDRAM FIFO overflow
    - LED4-7: not used.
  - INIT: indicates successful initialization of the FPGA. Green under normal operation.
  - DONE: indicates successful configuration of the FPGA. Green under normal operation.
  - B: Power LEDs: all 4 should switch on at power-on.

LCD:

- 19: At power-on, the LCD remains empty. When a session is successfully opened from the PC, the LCD displays the text ‘Neuropix BaseStation’. The LCD is also used to show the status of some BIST tests.

Switches:

- 23: Push buttons: SW2 resets the FPGA dev board. SW3-6 are not used.
- 28: Push button (SW14): Reconfigures the FPGA. Not used for operation of the recording system.
- 37: Push button (SW7): CPU reset button. Not used for operation of the recording system.
- 24: 4-position DIP switch (SW11): GPIO DIP switch. Not used for operation of the recording system.
- 29: 5-position DIP switch (SW13): Defines the configuration behavior at power-up and location of the configuration on the Flash memory. The switches are set to the position as shown below.

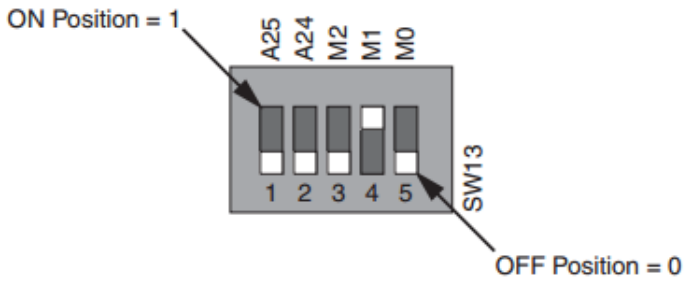

Figure 17: Default settings configuration switch (SW13).

- 27: power switch (SW15): switches the recording system on/off.
- 25: rotary switch (SW8): controls the intensity of the LCD.

FPGA with cooling fan (1 & 12):

- The KC705 FPGA configures the recording system with configuration data received from the PC.
- The FPGA aligns the start of the neural data recording with the start trigger signal and recording of the 16-bit synchronization data.
- The FPGA verifies the values of the SYNC and COUNTER signal, which are generated by the neural probe and transmitted via the HS and BSC board. In case a COUNTER or SYNC value is missing or corrupted, the neural data stream is patched with zeros. In addition, the FPGA monitors transmission error status bits on the deserializer chip.
- The FPGA performs demultiplexing of neural data into AP and LFP channels.
- The neural and synchronization data are stored in the data memory buffer and read back by the FPGA for transmission to the PC.
- The fan speed is controlled by the FPGA code. The fan operation should not be obstructed.

Data memory (2):

- A 1GB SDRAM buffer is available on the board. The buffer is used as a FIFO. The neural data is moved to the buffer after processing by the FPGA. The buffer can store about 50 seconds of data. The user needs to regularly read out the data from the buffer via the TCP/IP port. In case the FIFO buffer is full, the buffer overflow LED (GPIO LED 3) will switch on. The user can still read out the remaining data in the buffer, but the buffer and probe need to be reset via the API functions before neural recording can be continued.

Configuration memory (3):

- A 128MB Flash memory is used for boot code storage.

Jumpers:

- The jumpers are not used and should remain in default configuration. The following figure and table indicate the default jumper configuration (extracted from KC705 User Guide).

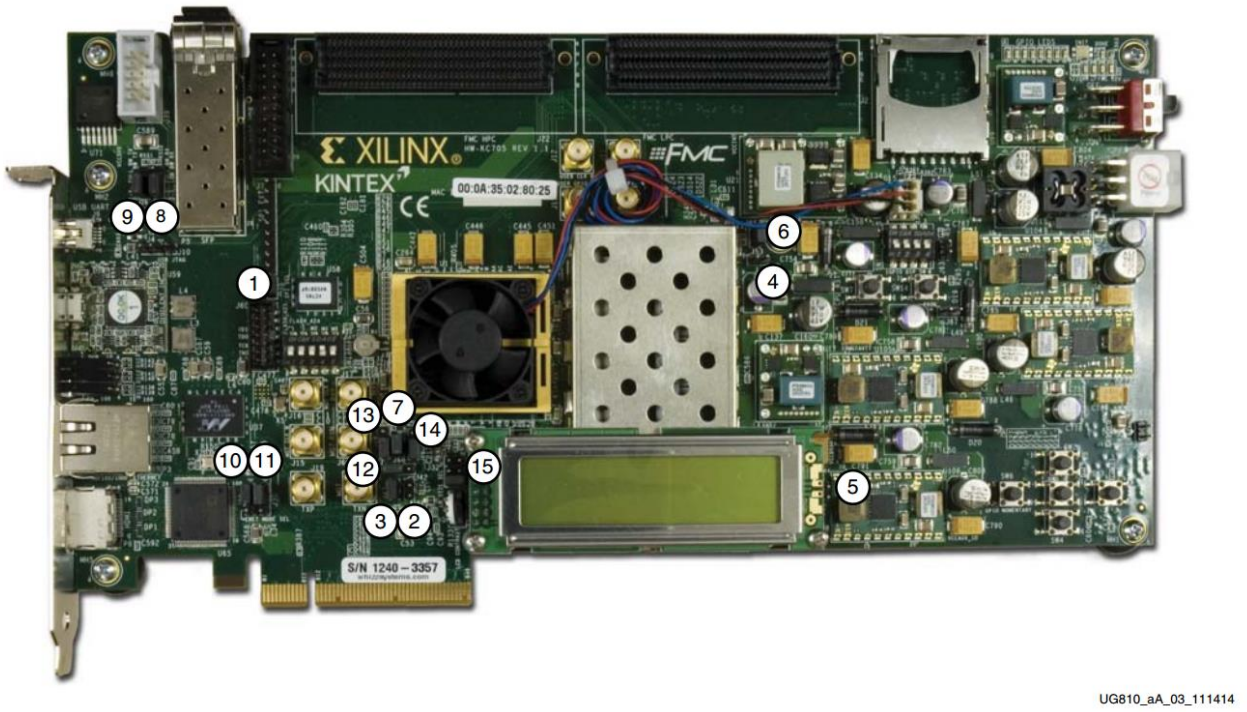

Figure 18: KC705 jumper positions.

Table 2: Default jumper positions.

Callout	Header Reference Designator	Default Jumper Position
1	J3	1-2

2	J42	None
3	J43	1-2
4	J53	None
5	J56	None
6	J65	1-2
7	J68	1-2
8	J27	2-3
9	J28	2-3
10	J29	1-2
11	J30	1-2
12	J47	1-2
13	J48	2-3
14	J69	1-2
15	J32	5-6

3 Hardware setup

3.1 Headstage

The HSs are delivered preconfigured by imec. No further configuration actions are required.

3.2 Basestation connect and FPGA development boards

The basestation connect board contains a version number, which can be read out via an API function.

The FPGA dev board needs to be configured with boot code. This is a one-time step. The code remains stored on the flash memory at power-off. The boot code is version controlled, and the version number can be read out via an API function.

In order to program the dev kit, the user needs to install Xilinx software, which can be downloaded from the Xilinx website. With the purchase of the KC705 FPGA dev board, a license for using the Xilinx Vivado is included. However, this toolkit is very extensive, most of the functionality is not required for programming the FPGA dev board. Therefore, it is also possible to use the lab edition of the Vivado toolkit, for which no license is needed.

For downloading the software from the Xilinx website, a user profile is required. When installing Vivado Lab Edition, make sure that the checkbox for cable driver is also selected (default ok).

1. After the Xilinx software is installed, plug the USB cable in the micro USB connector labelled as JTAG on the FPGA dev board. Verify that the switch SW13 is set to the default value (ref. chapter 2.4). Switch on the FPGA dev board.
2. Start the Vivado software. Select ‘Open Hardware Manager’.

Figure 19: Vivado ‘Open Hardware Manager’.

3. A green information bar is shown below the menu bar. Click “Open Target’. Select ‘Autoconnect’.

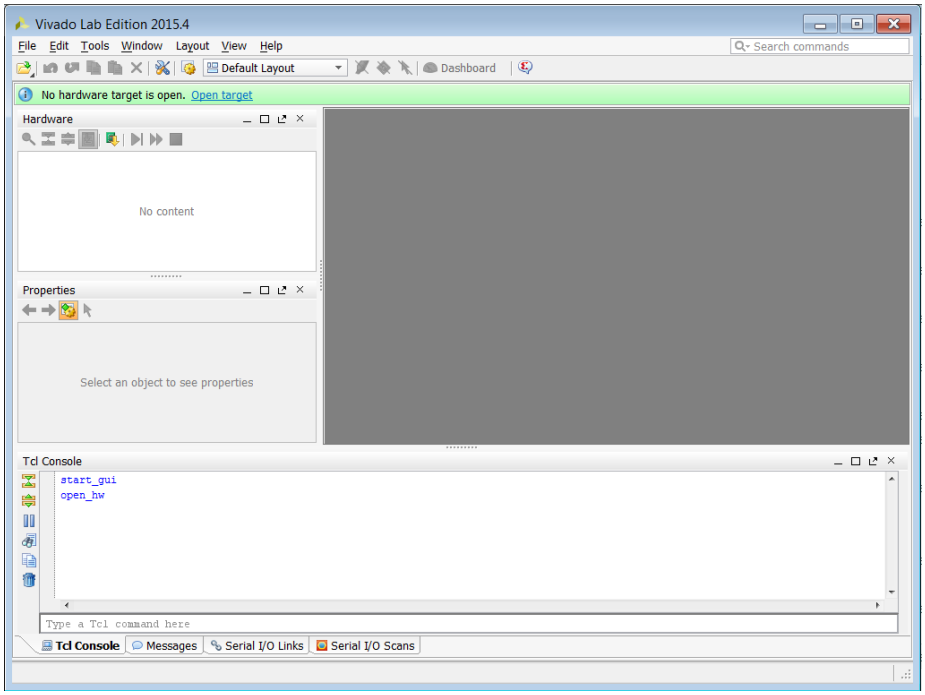

Figure 20: Vivado 'Open target'.

4. The FPGA dev board is detected. Right click 'xc7k325t_0'. Select 'Add Configuration Memory Device...'.

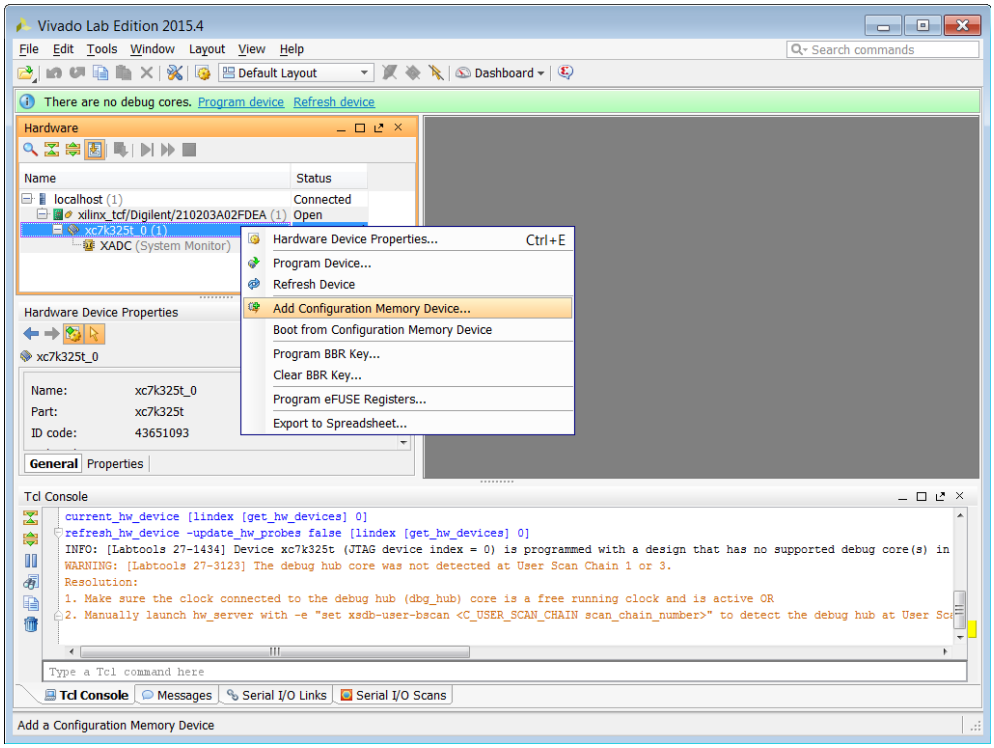

Figure 21: Vivado 'Add configuration memory device'.

5. Select 28f00ap30t-bpi-x16. Click 'OK'.

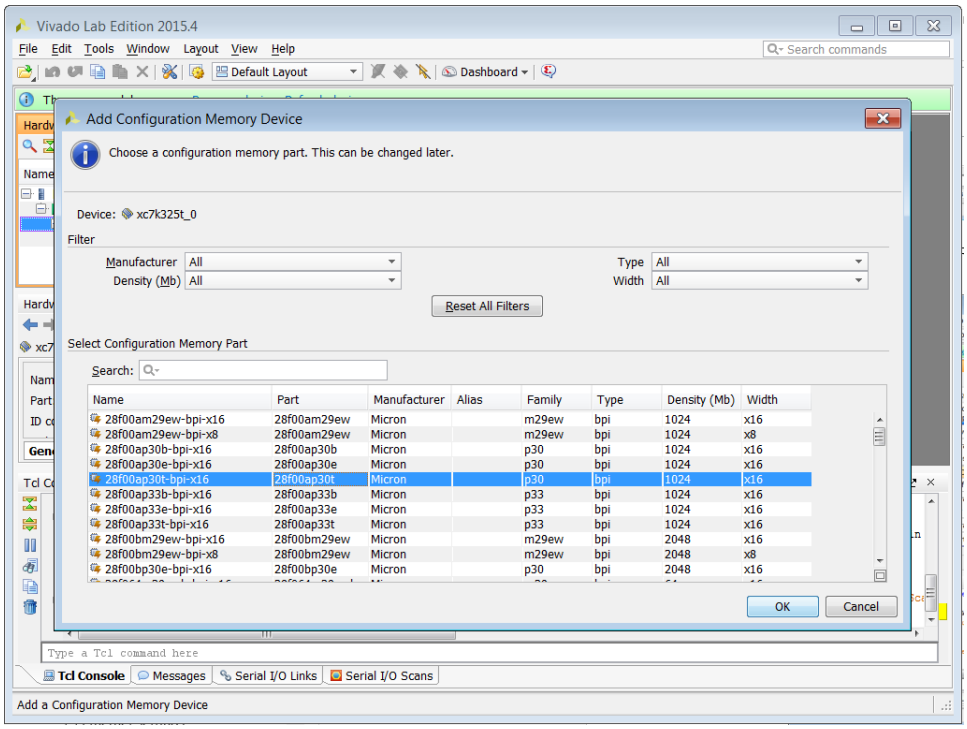

Figure 22: Vivado: select flash memory.

6. A pop-up window appears: ‘Do you want to program the configuration memory device now?’ Click ‘OK’.

7. In the field ‘Configuration file’, select the neuropix_basestation.mcs as provided by imec. Click ‘OK’.

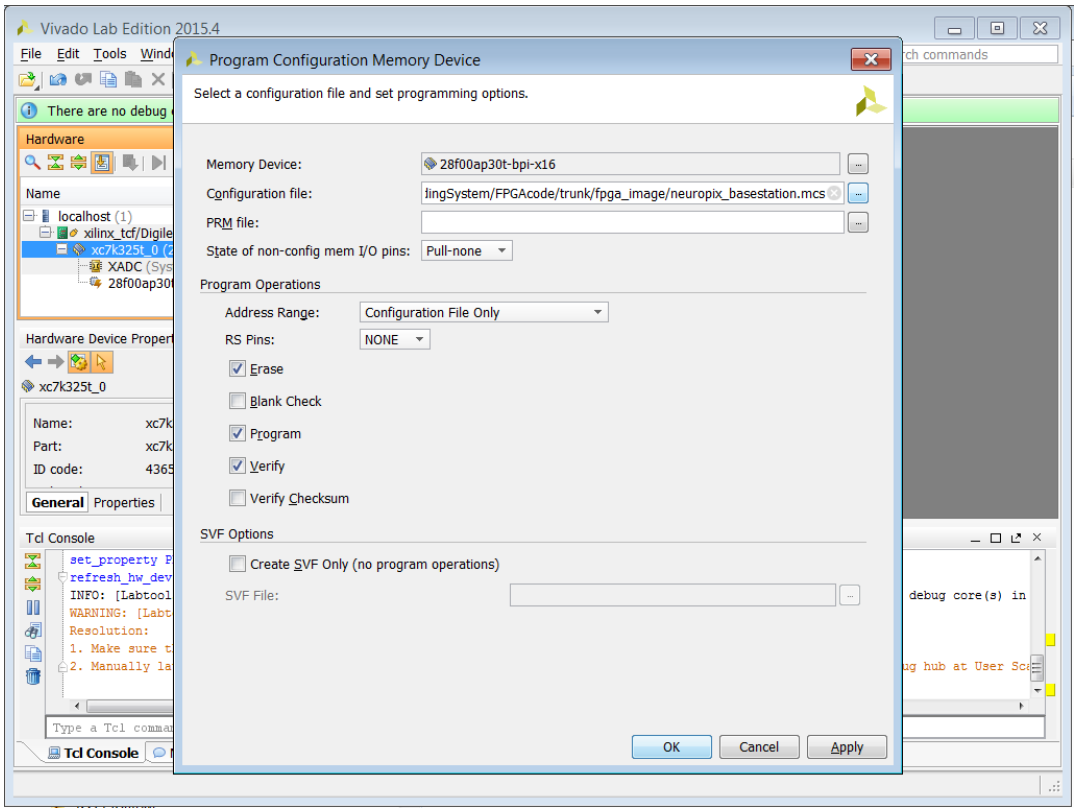

Figure 23: Vivado: selection of configuration file.

8. The device is then being programmed. This takes a few minutes. When successful, a message appears ‘Flash programming completed successfully’. The Vivado toolkit can be closed. At the next switch-on of the FPGA dev board, the configuration code is loaded onto the FPGA.

4 PC

4.1 Ethernet

The PC used for controlling the Neuropix system needs to be equipped with a 1000base-T Ethernet card.

The IP address needs to be set to 10.2.0.123, as indicated in the following picture.

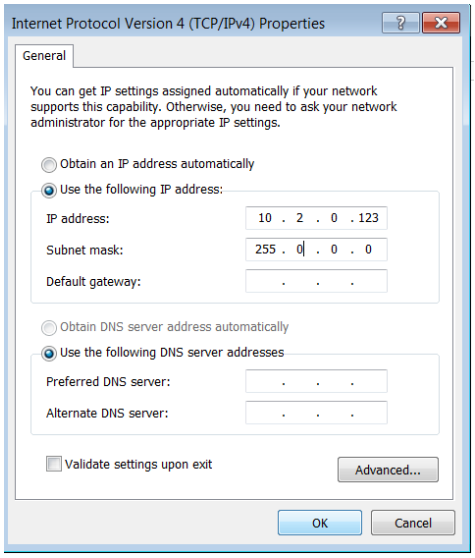

Figure 24: PC Ethernet settings.

4.2 Driver

The following DLL files are provided for building a user application:

- A DLL file for Windows, compiled with MingW, with .lib file.
- A DLL file for Windows, compiled with Visual Studio, with .lib file.

Also a folder with header files is included.

The API is version controlled. The version number can be read out using an API function.

5 Using the system

When handling the electronic boards (probe on flex, HS, BSC board, and FPGA dev board) it is advised to wear ESD protection (such as a wrist strap), in order to prevent ESD damage to the hardware. General ESD guidelines can be found under: http://www.esda.org/about-esd/esd-fundamentals/part-3-basic-esd-control-procedures-and-materials/.

5.1 Connections and initialization

1. Plug the flex in the HS. Follow the proper orientation (up/down) as shown in the picture below. Make sure that the flex is well aligned in the ZIF connector before closing the connector.

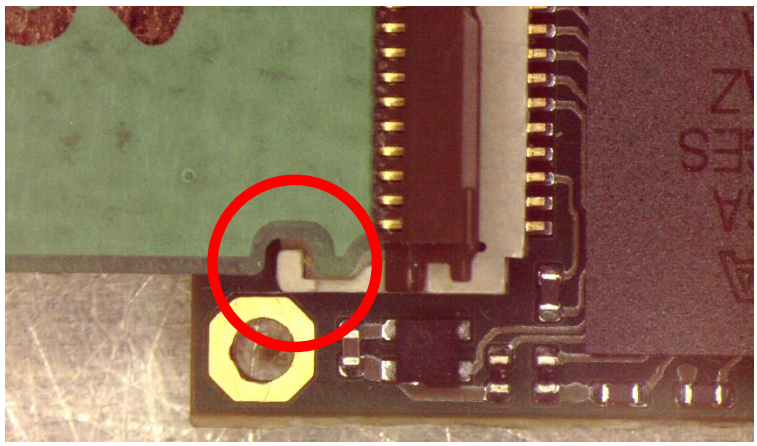

Figure 25: Alignment of flex in ZIF.

2. Plug the microcoax cable in the Omnetics connector on the HS. Pay attention to the orientation of the cable (one end is tagged as ‘headstage’, and is a 10-pin connector).

Figure 26: Connection of microcoax to HS.

3. Verify that the power supply configuration jumper of the BSC board is set to the default position (ref. chapter 2.4.3).
4. Plug the BSC board in the FMC connector (J22) on the FPGA dev board.
5. Plug the microcoax cable in the Omnetics connector on the BSC board. Pay attention to the orientation of the cable (one end is tagged as ‘basestation’ and is an 8-pin connector).
6. Make sure that all jumpers and switches of the FPGA dev board are in their default positions (ref. chapter 2.5).
7. Connect the power supply cable to the FPGA dev board. Connect the Ethernet cable between PC and FPGA dev board.
8. Switch on the FPGA dev board via the power switch. The FPGA will load its boot code. The fan will shortly operate at full speed and switch to quiet mode. The FPGA dev board LEDs ‘init’, ‘done’ and the 4 power LEDs will switch-on. GPIO LED2 will switch-on, indicating a successful SDRAM initialization. BIST #1, #2 and #3 will automatically start. If successful, the blue and green-yellow LEDs on the HS and blue LED on the BSC board are blinking simultaneously.
9. Transmit the API command neuropix_open() via the user application. If successful, the GPIO LED0 and GPIO LED1 on the FPGA board will turn on, indicating a data and configuration link between PC and BSC/FPGA dev board. The BIST #3 test is terminated, and the green-yellow LED on the HS turns on. The green NEUR_ENABLE LED on the BSC

turns on.
The recording system and probe are now configured to a predefined state. Specifically for the probe, the NRST and PR_NRST signal are set to logic high. TE is set to logic low (inactive). The shank is configured with all electrode sites disconnected.

The recording system is now ready for use.

5.2 Probe calibration

After power-up of a probe & system, the data for calibration of ADC comparator, slope, and offset stored on the EEPROM need to be loaded and applied to the probe. This is a mandatory step in the system initialization procedure. The ADC calibration ensures the reliable operation of the probe ADCs. The loading of ADC calibration parameters to the probe is done by calling the API function neuropix_applyAdcCalibrationFromEeprom.

The EEPROM also contains gain correction data that can optionally be loaded and applied to the FPGA dev board. Basically, the gain correction multiplies the signal on each channel by a constant factor in order to match the actual with the nominal gain and to reduce the channel-to-channel variability. This matching is optimized for a probe gain of x50. The loading of the gain correction parameters to the FPGA dev board done by calling the API function neuropix_applyGainCalibrationFromEeprom.

In Table 3 we provide expected average gain values for the different probe Options without and with gain correction applied. The values were obtained in PBS using signals of 100 Hz or 1.8 KHz for LFP and AP channels, respectively; with input amplitudes of 10 mVpp, 500 uVpp, 10 mVpp, 500 uVpp and 200 uVpp for gain settings 50, 1000 and 2500. Probes were always fully immersed in PBS.

Table 3: Indicative gain values for different probe types with and without gain correction.

		AP			LFP		
Nominal		50	1000	2500	50	1000	2500
Option 1	Uncorrected[1]						
	Corrected						
Option 2	Uncorrected						
	Corrected[2]	-	-	-	-	-	-
Option 3	Uncorrected	33-37					
	Corrected	49-50	980-1000	2550-2650	55-56	1100-1150	2850-2950
Option 4	Uncorrected	~40	800-810	2100-2200			
	Corrected	49-50	980-1000	2550-2650	55-56	1100-1150	2850-2950

[1] Uncorrected = Measured without gain correction, but with ADC calibration.

[2] No gain correction available for Option 2 probes.

5.2.1 *ADC calibration from csv file*

In case the EEPROM on the flex has detached or is no longer functional, the ADC calibration parameters and gain correction parameters can be read from csv files provided by imec and applied to the probe or FPGA dev board by using a few API functions.

The following procedure is an example code for MSVC. Prior to using this sequence of functions, the user must already have set the probe type using the function neuropix_writeId().

The filescontaining the ADC calibration info as provided by imec are named “Comparator calibration.csv”, “Offset calibration.csv” and “Slope calibration.csv”.

```
Neuropix_basestation_api api;
std::vector<adcComp> adcCompC;
std::vector<adcPairCommon> adcPairCommonC;

//Read from csv and apply to API and read from API
api.neuropix_readComparatorCalibrationFromCsv("Comparator calibration.csv");
api.neuropix_readADCOffsetCalibrationFromCsv("Offset calibration.csv");
api.neuropix_readADCSlopeCalibrationFromCsv("Slope calibration.csv");
//Read parameters from API
api.neuropix_getADCCompCalibration(adcCompC);
api.neuropix_getADCPairCommonCalibration(adcPairCommonC);
//Write parameters to probe

for (size_t i = 0; i < 15; i=i+2)
{
	api.neuropix_ADCCalibration(i, adcCompC[2 * i].compP, adcCompC[2 * i].compN,
adcCompC[2 * i +2].compP, adcCompC[2 * i+2].compN, adcPairCommonC[i].slope,
adcPairCommonC[i].fine, adcPairCommonC[i].coarse, adcPairCommonC[i].cfix);

	api.neuropix_ADCCalibration(i+1, adcCompC[2 * i+1].compP, adcCompC[2 *
i+1].compN, adcCompC[2 * i + 3].compP, adcCompC[2 * i + 3].compN,
adcPairCommonC[i+1].slope, adcPairCommonC[i+1].fine, adcPairCommonC[i+1].coarse,
adcPairCommonC[i+1].cfix);
}
```

5.2.2 *Gain correction from csv file*

The following procedure is an example code for MSVC for reading the gain correction parameters from the csv file and applying these to the FPGA dev board. Prior to using this sequence of functions, the user must already have set the probe type using the function neuropix_writeId().

The file containing the gain correction info as provided by imec is named “Gain correction,csv”.

```
Neuropix_basestation_api api;
std::vector<unsigned short> gainCorrectionData_;
```

```
//Read gain correction from csv and apply to API member
api.neuropix_readGainCalibrationFromCsv("Gain correction.csv");
//Read gain correction from API member
api.neuropix_getGainCorrectionCalibration(gainCorrectionData_);
//resize according to probe type
unsigned int option = api.neuropix_getOption();
if (option < 2)
	gainCorrectionData_.resize(384);
else if (option == 2)
	gainCorrectionData_.resize(960);
else if (option == 3)
	gainCorrectionData_.resize(966);
//Write to basestation FPGA
api.neuropix_gainCorrection(gainCorrectionData_);
```

5.3 Probe configuration

The probe can be configured for electrode selectivity, reference selection, gain and bandwidth setting. Refer to the API documentation for more information.

5.3.1 *Electrode selectivity*

Shank programmability is only provided for Option 3 and 4 probes. For Option 3 probes, the shank contains 960 electrodes, while 384 channels are available on the base. For Option 4 probes, the shank contains 966 electrodes, and the base contains 276 channels. Each channel can connect to only a limited number of electrodes and only to one electrode at the same time. The mapping between electrodes and channels is different between Option 3 and Option 4 probes.

A complete table of channels mapping to electrodes is provided in document '2016-01-20_Electrode_mapping.xlsx'. The following table shows an extract of the complete mapping:

Table 4: Electrode to channel mapping for Option 3.

	Bank number 0	**Bank number 1**	**Bank number 2**
Channel	**Electrode**	**Electrode**	**Electrode**
0	1	385	769
1	2	386	770
…	…	…	…
383	384	768	

The following figures show the grouping of electrodes into banks:

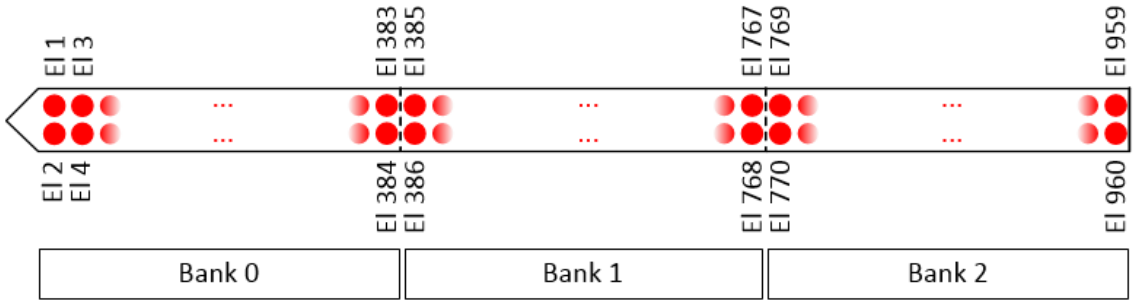

Figure 27: Electrode bank distribution over the shank for Option 3 probes.

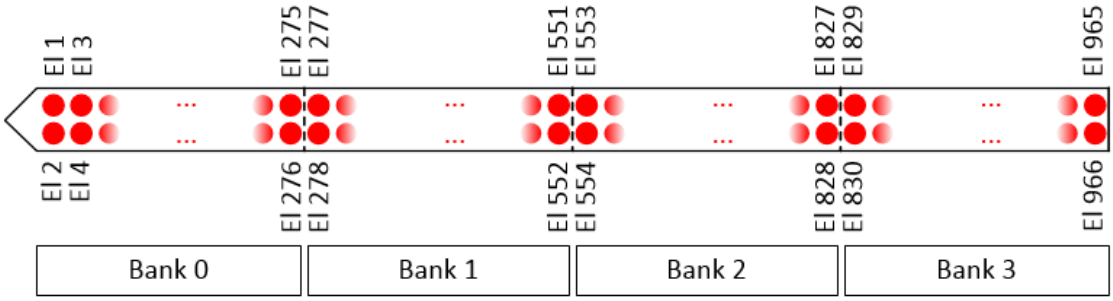

Figure 28: Electrode bank distribution over the shank for Option 4 probes.

The API function neuropix_selectElectrode() sets to which electrode a channel is connected. By using this function, the user can only connect one electrode at a time to a channel. Furthermore, the function checks the probe type, and returns an error in case the user tries to make an illegal electrode to channel connection.

Also the internal reference electrodes are connected/disconnected to the channel / reference mux via the aforementioned API function. Two important notes:

1. Channels mapped to internal reference electrodes cannot be used for recording, even if these electrodes are not used as a reference.
2. If the internal reference electrodes are not used as a reference, they need to be disconnected (via the API function neuropix_selectElectrode).

5.3.2 *Reference selection*

The user has the option to select the reference input for each channel on the ASIC. This reference selection differs for the different probe Options.

For Option 1, 2 and 3 probes, for each of the 384 channels on the base, 1 of 11 possible reference inputs can be selected. The first reference input connects to the external reference input pin on the probe. The other 10 reference input lines connect to electrodes on the shank (internal references).

The figure below shows a drawing of the reference selection switch for Option 1, 2 and 3 probes:

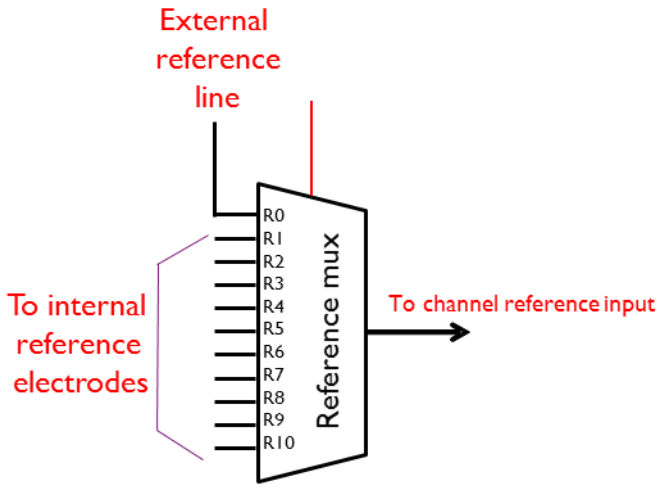

Figure 29: Reference mux for Option 3 probes.

For Option 4 probes, for each of the 276 channels on the base, 1 of 8 possible reference input lines can be selected. The first reference input connects to the external reference input pin on the probe. The other 7 reference input lines connect to electrodes on the shank (internal references).

The figure below shows a drawing of the reference selection switch for Option 4 probes:

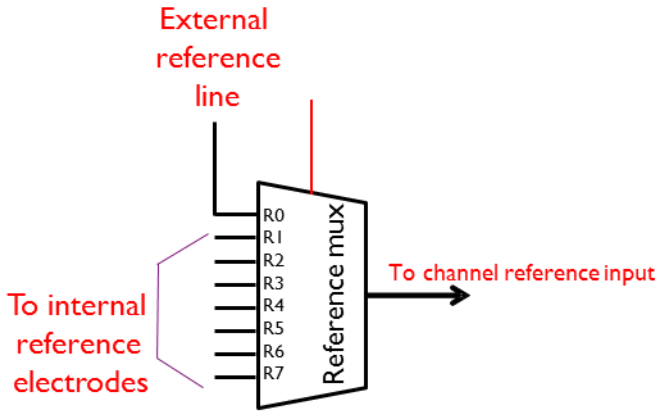

Figure 30: Reference mux for Option 4 probes.

The reference line which is selected as reference input to a channel is set by the API function neuropix_setReference(). This function allows the user to set the reference input for each channel individually.

Because of the shank programmability for Option 3 and Option 4 probes, the internal reference lines can connect to several reference electrodes along the shank. Therefore, which internal reference electrode is used as a reference input for the channel is defined by the API function neuropix_setReference() and neuropix_selectElectrode(). Please refer to the picture below.

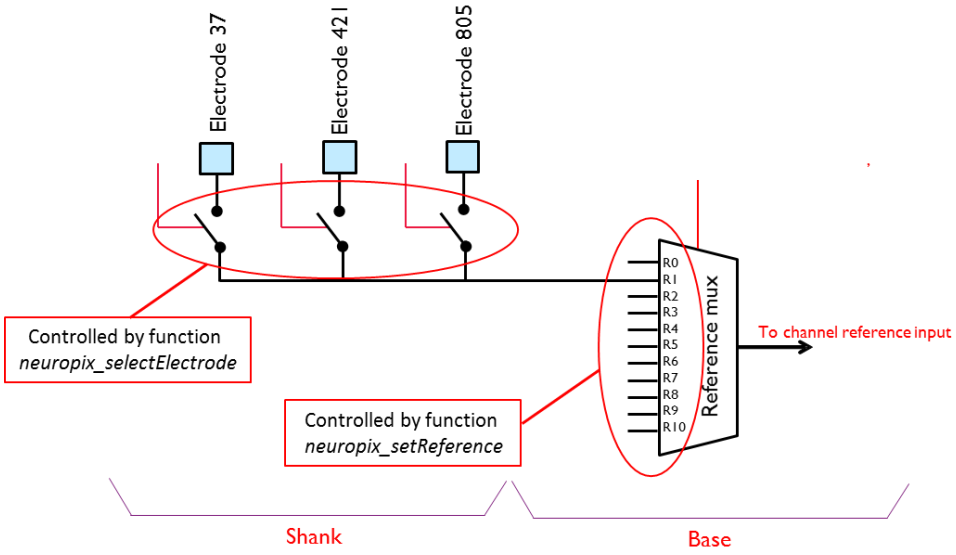

Figure 31: Setting the reference channel input via API functions (example for Option 3 probe).

For Option 3 probes, an overview of which electrodes can connect to the internal reference lines is given in the table below:

Table 5: Electrode mapping to internal reference lines for Option 3.

Ref input	**Electrode**	**Electrode**	**Electrode**
R0 (External)	External reference input		
R1 (Internal 1)	37	421	805
R2 (Internal 2)	76	460	844
R3 (Internal 3)	113	497	881
R4 (Internal 4)	152	536	920
R5 (Internal 5)	189	573	957
R6 (Internal 6)	228	612	
R7 (Internal 7)	265	649	
R8 (Internal 8)	304	688	
R9 (Internal 9)	341	725	
R10 (Internal 10)	380	764	

For Option 4 probes, an overview of which electrodes can connect to the internal reference lines is given in the table below:

Table 6: Electrode mapping to internal reference lines for Option 4.

Ref input	**Electrode**	**Electrode**	**Electrode**	**Electrode**
R0 (External)	External reference input			
R1 (Internal 1)	37	313	589	865

R2 (Internal 2)	76	352	628	904
R3 (Internal 3)	113	389	665	941
R4 (Internal 4)	152	428	704	
R5 (Internal 5)	189	465	741	
R6 (Internal 6)	228	504	780	
R7 (Internal 7)	265	541	817	

There are a few important restrictions to follow:

1. All channels which connect to a reference line Rx need to connect to the same electrode. Example: All channels which connect to reference R1 need to connect to the same electrode: 37, or 421 or 805. This restriction is guaranteed by API design.
2. Each reference line can connect to only one internal reference electrode. This restriction is guaranteed by API design.
3. Each channel can connect to only one reference line. This restriction is guaranteed by API design.
4. All reference electrodes which are not connected to by any of the 384 channels need to be disconnected. Example: if none of the 384 channels connects to R2, then electrode 76, 460 and 844 need to be disconnected. This can be done via the API function neuropix_selectElectrode().
5. In case none of the channels has selected the external reference line as a reference input signal, the external reference line needs to be disconnected. This can be done via the API function neuropix_setExtRef().
   The figure below shows the functionality of the API function neuropix_setExtRef():

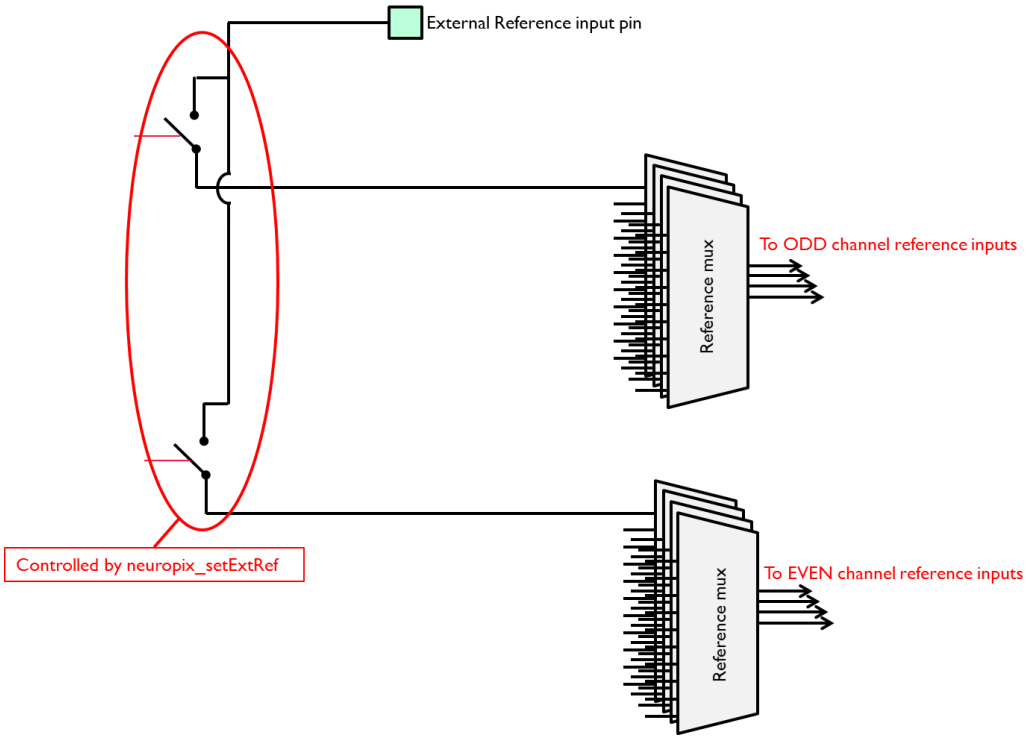

Figure 32: Control of External Reference input via API function.

5.4 Start trigger and synchronization

The start trigger and synchronization port can be used for synchronization of the neural recording experiment. For the time being, the start trigger can only be used in output mode.

During system initialization, the user should configure the system in start trigger output mode, using the API function neuropix_triggerMode().

The neural data acquisition by the probe is started by calling the API function neuropix_setNeuralStart(). This also generates a start trigger available for the user. This command replaces the previous procedure of actions required to be taken by the user:

1. Set probe in reset mode (neuropix_nrst('false'))
2. Reset FPGA dev board datapath (neuropix_resetDatapath())
3. Set probe in active mode (neuropix_nrst('true'))

The start trigger is available on the Ext. Start connector on the BSC board. When the user calls the API function neuropix_setNeuralStart, an active low pulse is generated on the Ext. Start connector. The rising edge of the Ext. Start pulse occurs simultaneous with the start of sampling of the neural data by the probe within 1 sample period (1/30 KHz) accuracy.

The 16 bit signal applied on the Synch. connector on the basestation connect board is recorded simultaneous with the neural data recording to 1 sample period (1/30KHz) accuracy.

5.5 Recording data

The probe transmits data when the recording mode is set to 'Impedance' or 'Recording' and NRST set to logic 'high'.

Follow the subsequent order of steps to start a recording session:

1. Set the probe in the selected mode ('Recording' or 'Impedance'), using API function neuropix_mode().
2. Optional: Set NRST to logic 'high' using API function neuropix_nrst(). The probe is now set to generate data. The NRST is already set in this mode by the neuropix_open() procedure.
3. Set the start trigger mode to input or output, using API function neuropix_triggerMode().
4. If required to stream the data to disc, call API function neuropix_startRecording to select the file to which all data read via the function neuropix_readElectrodeData() will be streamed to.
5. If the start trigger is set to output, generate a start trigger via the function neuropix_setNeuralStart().
   ~~If the start trigger is set to input, no data will be available in the FPGA dev board memory buffer until an external trigger pulse arrives on the BSC.~~
6. Transfer the incoming data from FPGA dev board to PC: call the API function neuropix_readElectrodeData() to read one packet of electrode data from the FPGA dev board memory buffer. Each packet contains 12 samples for each of the 384 AP channels and 1 sample for each of the 384 LFP channels.
7. To terminate the disc streaming and closing the recorded file, call the API function neuropix_stopRecording().

In case the data is not read out in time, the SDRAM buffer will overflow and FPGA dev board GPIO LED3 switches on. In this case a partial probe and buffer reset is required. This can be done manually or via the start trigger:

a. Manually: call the following API functions in order:
   1. Set NRST to logic 'low' using API function neuropix_nrst();
   2. Reset the FPGA SDRAM buffer via API function neuropix_resetDatapath();
   3. Set NRST to logic 'high' using API function neuropix_nrst();

b. If the start trigger mode is set to output: call API function neuropix_setNeuralStart();

~~c. If the start trigger mode is set to input: apply a new external start trigger to the BSC board.~~

The recorded .npx file can be read using the function neuropix_readElectrodeData(). Also a tool is provided to convert the .npx file to a .csv file (ref. chapter 5.13).

5.6 Probe signal offset

The digitized output signals of the probe have an offset which is dependent on the gain setting. Furthermore, there's a certain channel-to-channel variability in the offset value.

For a gain setting of 50 the offset is around 0.6V. For other gain settings the offset is close to 0.7V.

5.7 Probe signal inversion

The recorded signal is inverted by the amplifier structure on the probe.

5.8 Switch-off

To turn off the system:

1. Call the API function neuropix_close(). The GPIO LED0 and GPIO LED1 on the FPGA board will turn off.
2. Power off the recording system using the power switch located on the FPGA dev board.

The separate components of the neural recording system can now be disconnected.

5.9 Probe handling & cleaning

The probe must be manipulated with utmost care to prevent damage to shank and electrodes. In the following we provide an overview of some guidelines:

- Do not touch the shank with your fingers or other objects.
- Do not subject the probe to vibrations (potential damage of TiN electrodes and shank). Such vibrations could arise e.g. when blow-drying the probe with compressed air or N_2 or when subjecting the probe to ultrasound.
- Always work in a clean environment.
- Regularly inspect the probes with an optical microscope to verify that shank and electrodes are undamaged and clean. Color changes/darkening of the TiN can indicate corrosion or other damage. Please report such observations.
- Strong oxidizing agents (acids, bases, H_2O_2, etc) and/or elevated temperatures (>100 degC) should be avoided when handling TiN electrodes as these may irreversibly deteriorate the impedance.
- Store the probes in the transparent membrane boxes they were shipped in.
- Each probe flex has a unique ID. Please refer to this unique ID in your data and communication.
- The following solvents and solutions have been tested and are safe to use with Neuropix probes (no other organic solvents should be used):

- Deionized water (DIW)[3] at room temperature
- Iso-propyl alcohol (IPA)
- Phosphate buffered saline (PBS) @ pH 7.4[4] (e.g. https://www.sigmaaldrich.com/content/dam/sigma-aldrich/docs/Sigma/Datasheet/pbs1dat.pdf)
- 25 mM MOPS[5] (http://www.sigmaaldrich.com/catalog/product/sigma/m1254?lang=en&region=BE), 150 mM NaCl, DI water, pH adjusted w/ NaOH and HCl
- 100 mM MES[6] (http://www.sigmaaldrich.com/catalog/product/fluka/69889?lang=en&region=BE), 150 mM NaCl, DI water, pH adjusted w/ NaOH and HCl
- Tergazyme (http://www.sigmaaldrich.com/catalog/product/aldrich/z273287?lang=en&region=BE)

- Following these user/cleaning guidelines:
  - When performing experiments in PBS, avoid immersing the electric components on the flex backside into the PBS. This can cause shorts and may damage the components and probe. Immediately rinse with DIW and IPA and re-test the probe.
  - After use in PBS, always rinse the probes (including the electrical components) under a gentle DIW stream or in a beaker (1-2') followed by IPA rinsing (1-2'). Do not use a high-pressure water or IPA stream. After IPA rinsing, you can let the probes dry in air. The use of IPA is not mandatory, but advised.
  - Avoid leaving the probes dry out after removal from PBS. Salt crystals may form on the shank and TiN electrodes that are difficult to remove.
  - Always use fresh solutions, i.e. don't use beakers with DIW, IPA, or PBS after prolonged exposure to air (~1/2 day) since dust particles on the liquid surface may

[3] Rinsing or prolonged soaking of Neuropix probes in DIW (pH ≈ 5.7) does not deteriorate the TiN impedance; if anything, the impedance slightly decreases after prolonged soaking (likely due to improved wetting). DIW degassing is not needed.

[4] In Neuropix Phase 2, IMEC has tested the impact of PBS pH (6.6-7.6) on TiN impedance. No negative effect could be observed.

[5] In Neuropix Phase 2, IMEC has tested the impact of MOPS pH (4.5-9.2) on TiN impedance. No negative effect could be observed.

[6] In Neuropix Phase 2, IMEC has tested the impact of MES pH (3.5-9.3) on TiN impedance. No negative effect could be observed.

irreversible attach to the shank surface when immersed or rinsed with these solutions.

- After in-vivo use, soak the dirty probes in PBS until ready to clean them. Letting the shank dry out makes the cleaning less effective. Follow these cleaning steps:
    - Prepare 1% Tergazyme solution.
    - Soak explanted probes in standard PBS (pH 7.4) until cleaning w/ Tergazyme (to prevent drying out).
    - Immerse probes in Tergazyme solution @ RT (2h soaking on shelf).
    - Thoroughly rinse with DIW (~5 min under gentle DIW stream); if soaking in DIW is used, a brief rinse with DIW should be applied at the end.
    - Rinse the probes with IPA and let dry in air.
    - Place probes back in their membrane box.

5.10 Gain & noise measurements in saline

Two tests that should be performed with any newly received probe are gain (x2500) and noise measurements in saline. These measurements should be performed after loading calibration settings and gain correction factors[7]. Note that for Option 2 probes, no gain correction factor could be determined due to the attenuation effect described further below. For Option 2 probes, the gain correction stored on the EEPROM is 1.

5.10.1 *Gain measurement*

Gain measurements must be performed using the external REF shorted to GND. This can be achieved by soldering two wires to the respective pads on the probe flex and shorting them to the signal generator's ground (Figure 33, Figure 34). The probe should be configured for a nominal gain setting of 2500 both for AP and LFP channels and immersed in PBS. A sinusoidal test signal of e.g. 200 uVpp and 1.8 kHz or 100 Hz, respectively, should be applied to the PBS for AP and LFP gains, respectively, using a Pt, stainless steel, or Ag|AgCl electrode. The actual gain value should be around 2600 for AP and 2900 for LFP channels for Options 1, 3, and 4 probes.

[7] Guidelines on how to load calibration and gain correction values will be provided by the developers of the user software.

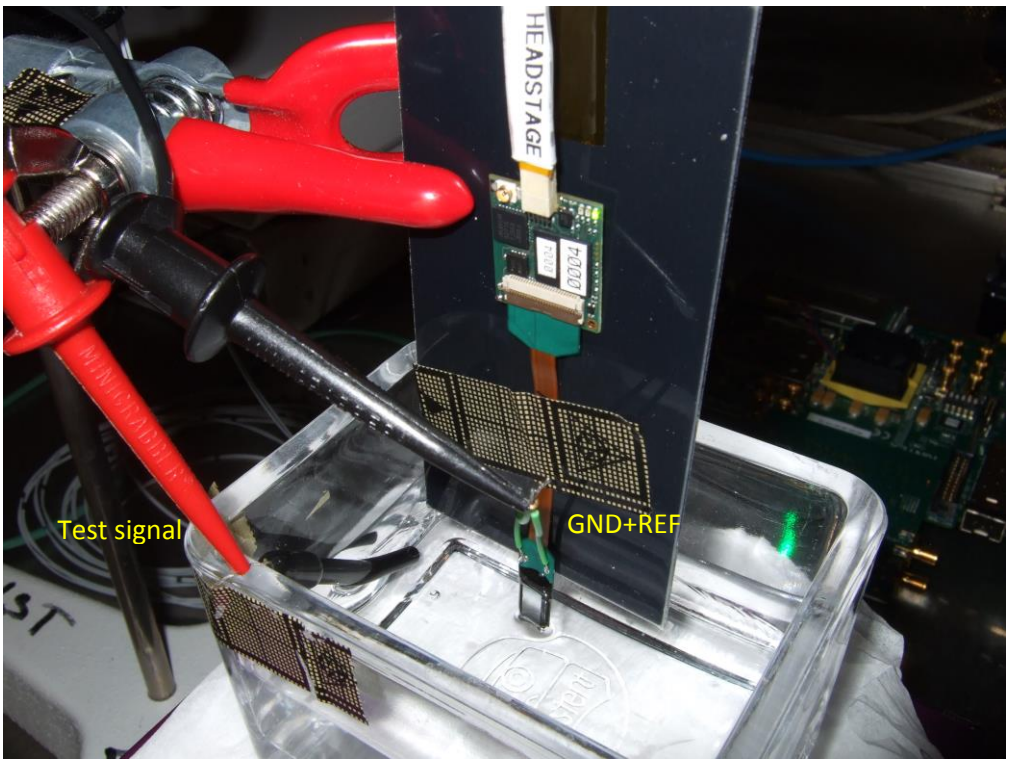

Figure 33: Signal, GND, and external REF connectivity during gain measurements.

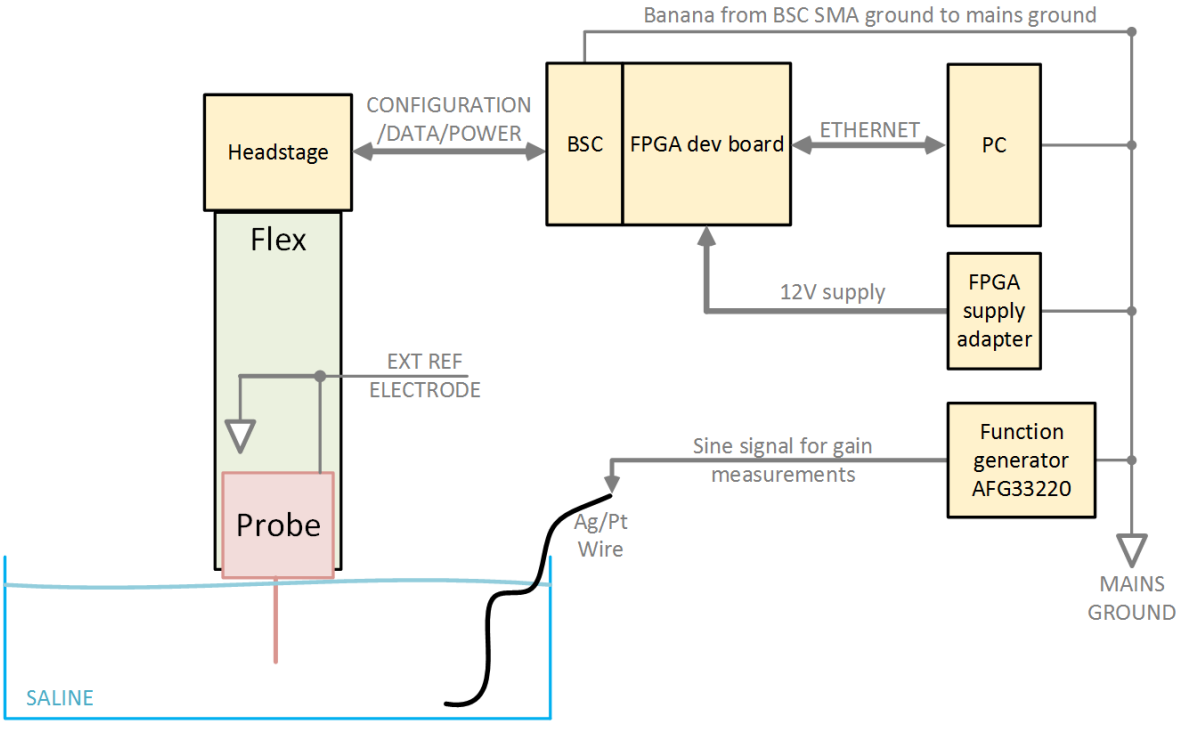

Figure 34: Cartoon showing connectivity for gain measurements.

For Option 2, the gain cannot be determined using the method above. Due to the insufficiently large OFF resistance of the REF switches in the probe base, the measured (common) signal will be

attenuated by a factor of 2.5-3. To get some indication of the channel gain of Option 2 probes, one can immerse only the proximal electrodes (1-36) in PBS thus and avoid signal pick-up by the internal REF electrodes (37, etc). In this case, the measured gain will be ~2300 and ~2400 for AP and LFP channels, respectively.

Further explanations of the attenuation effect are provided in Figure 35. Due to the attenuation effect, common signals such as test signals in PBS, LFP signals in the brain, and AP signals in the brain close to the internal REF electrodes will be attenuated (even when the external REF is used).

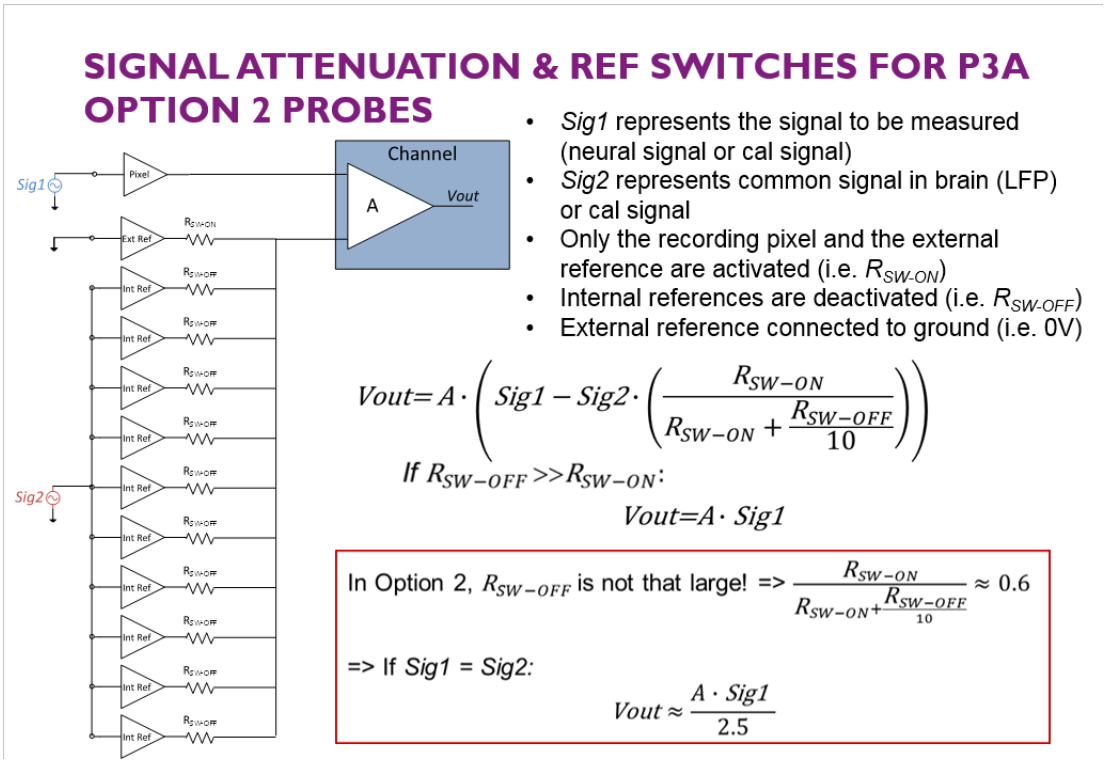

Figure 35: Common signal attenuation for Option 2 probes.

5.10.2 *Noise measurement*

Noise measurements should always be performed in a grounded Faraday cage. One can use either an internal REF electrode or the external REF shorted to GND. The PBS solution must be grounded (Figure 36). During noise measurements, all internal REF electrodes must be immersed in PBS to avoid unwanted noise pick-up and cross-coupling.

- When using the external REF, the internal REF electrodes must be disconnected.
- When using an internal REF electrode, the external REF electrode must be disconnected and grounded. Internal REF electrodes are grounded through the PBS solution.

The nominal gain for both AP and LFP channels should be set to 2500. To obtain the actual input-referred noise, one must divide the measured noise by the actual gain measured in 5.10.1 above.

Dividing by 2500 results in slightly overestimated noise values (in particular for LFP channels) for Option 1, 3 and 4 probes.

In order to determine the noise for Option 2 probes, one should divide by the gain measured with only the first 36 electrodes immersed in saline. Dividing by 2500 provides slightly underestimated but nonetheless close estimates.

One can expect a noise of <7 uVrms for Option 1 and 3 probes and <10 uVrms for Option 2 and 4 probes.

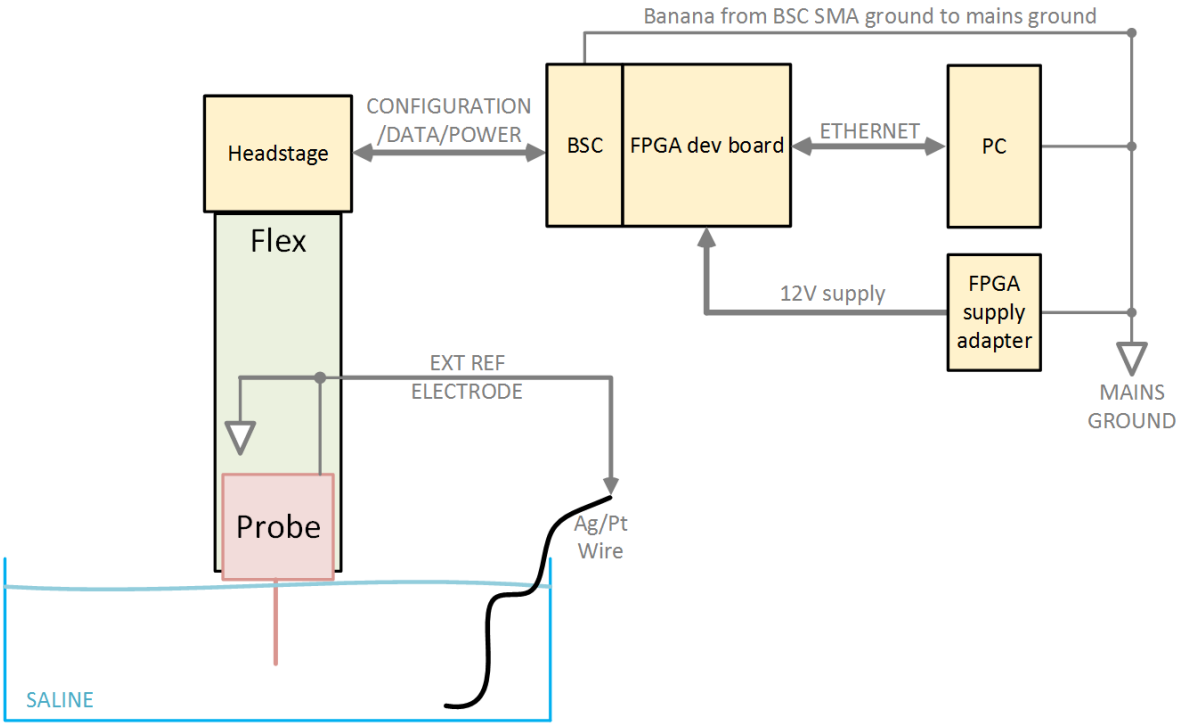

Figure 36: Cartoon showing the connectivity for noise measurements.

5.11 Start trigger & synchronization port measurements

The goal of this experiment is to verify the functionality and timing accuracy of the start trigger and synchronization port.

When the user calls the API function neuropix_setNeuralStart, the system performs a reset of the probe and the datapath through HS and BS FPGA. The BS FPGA generates a start trigger on the Ext. Start SMA connector of which the rising edge coincides with the start of neural data recording by the probe. From this point on, the 16-bit signal on the Synch. port is recorded simultaneous with the neural data.

The system is configured in start trigger output mode. Signals between HS and probe are measured on the ZIF connector using general purpose Cascade probes. The probe is immersed in PBS solution. An arbitrary waveform generator is used to generate a test pattern to the probe and to the Sync port. The electrical signals are visualized on an oscilloscope. The probe data is recorded with the recording system and analysed with Matlab.

The picture below shows a scope screenshot of electrical signals related to the start trigger. The NRST signal going high, a signal from HS to probe, brings the probe out of reset mode, and triggers the probe to start acquisition. The SPI_NCS signal is an output signal from the probe, toggling at a rate equal to the probes ADC sampling rate. The SPI_NCS signal toggles high and low right after the ADCs on the probe have completed one conversion of a channel. The SPI_NCS signal is encoded in the data stream from the HS to the BS FPGA development board, where it is used to generate the start trigger (Ext. Start) and to sample the 16-bit Synch port. For this experiment the Ext. Start signal is used to trigger the arbitrary waveform generator to generate a 3-pulse pattern which is connected to the saline solution in which the probe is immersed, and to 1 input of the 16-bit Synch. port. The 3-pulse pattern occurs 6.32ms after the start trigger.

Figure 37: Start trigger oscilloscope screenshot.

The figure below shows the recorded probe data, and the 1-bit signal recorded on the Synch. port. Although the probe is still stabilizing after the reset, the three pulses are present at the expected time.

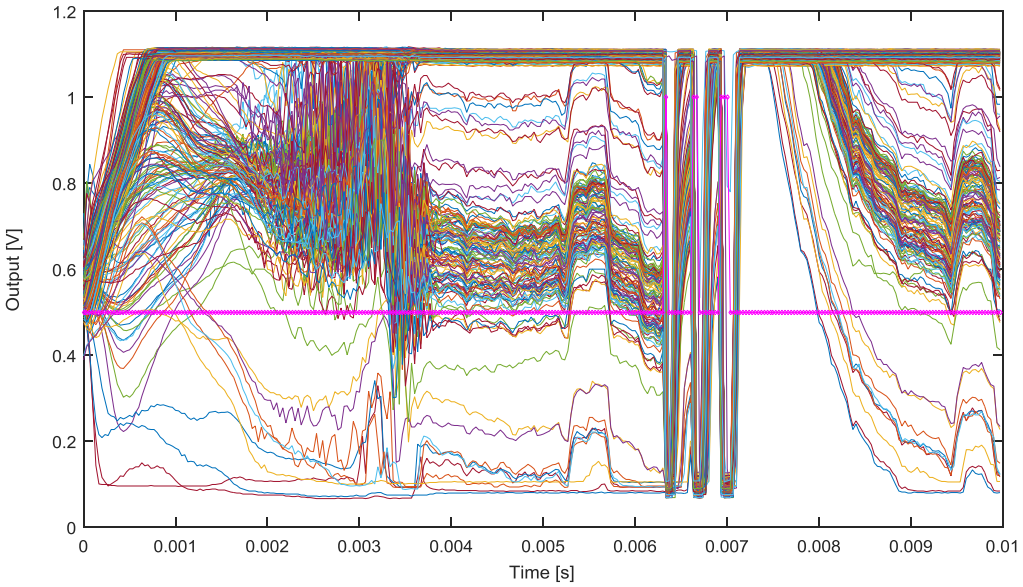

Figure 38: Probe data and Synch. port signal with input pulse at 6.32ms.

The 16-bit Synch port signal is scaled to 1. The 3-pulse input signal is connected to the MSB of the 16-bit Synch port. All other bits are pulled-up. This results in a pulse between 0.5 and 1V. The Synch port signal is shown on the graphs as a magenta line with diamond markers.

Zooming in on the figure shows the 3 pulses separately. The 1-bit signal on the Synch. port shows the expected pulse lengths of 1, 2 and 3 sample periods. For the neural signal channels this is not so obvious as the input amplitude causes the channels to go in saturation, which requires some time for the channels to recover.

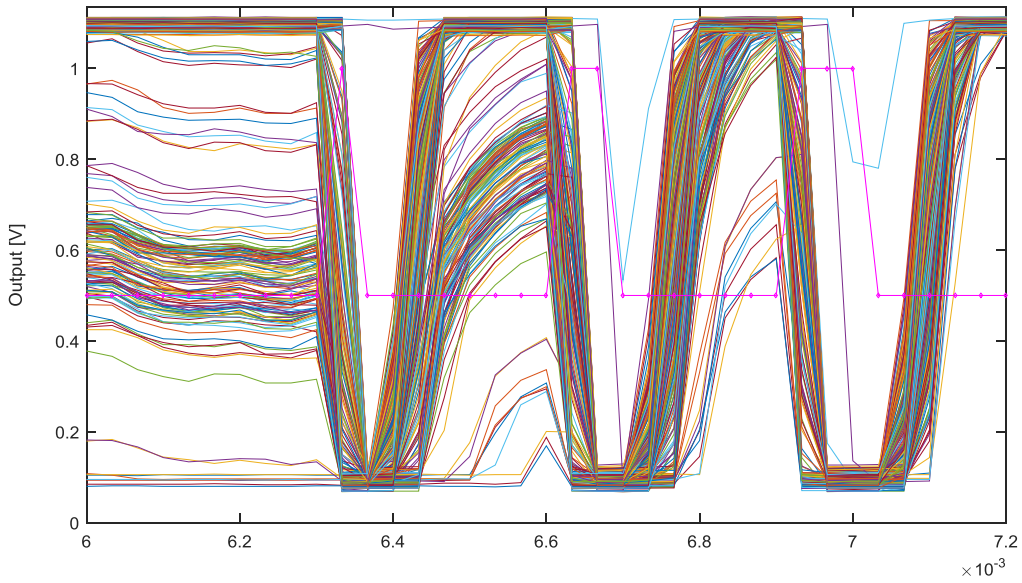

Figure 39: Probe data and Synch. port signal with input pulse at 6.32ms, detail.

Zooming in on the first pulse shows that the time location of the pulse corresponds with what was applied by the arbitrary waveform generator:

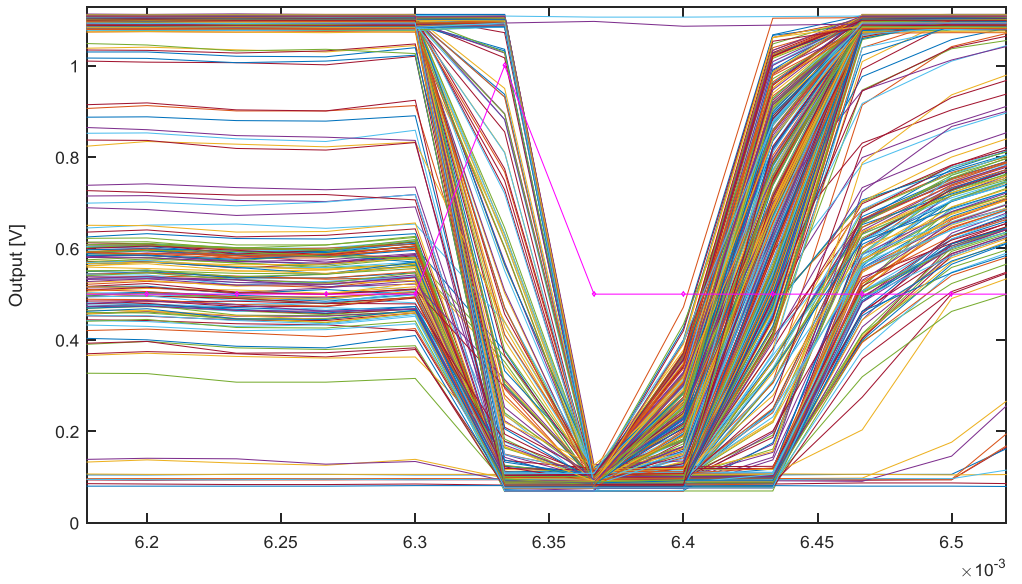

Figure 40: Probe data and Synch. port signal with input pulse at 6.32ms, first pulse only.

On the plots a difference of 1 sample period between different channels is observed: for some channels the falling edge of the pulse occurs one sample period after the other channels. This is due to the fact that not all the channels are sampled simultaneously: each

ADC on the probe samples 12 AP + 1 LFP channels sequentially. If the start of the input pulse does not coincide with the sampling of the first channel, there will be one sample period (1/30 KHz = 33us) difference between channels. This has been verified by changing the delay between the sync output and start of the input pulse in steps of +- 1/13 of 33 us. In this way it can be obtained that the falling edges of all channels are coinciding.

If the 3-pulse input signal is applied at a longer time after the start trigger, the probe has recovered from the reset, and the plots are more clear.

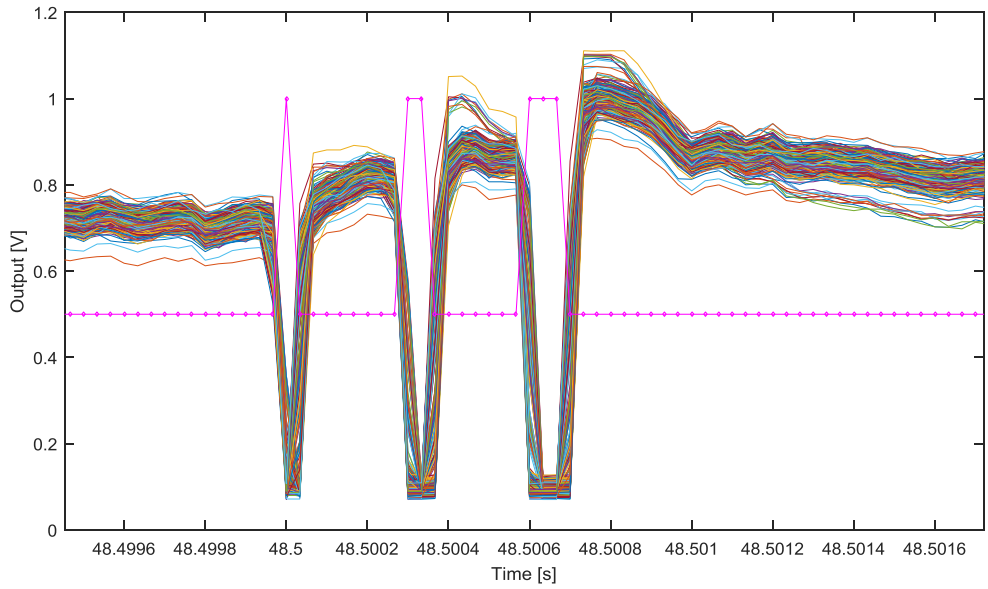

Figure 41: Probe data and Synch. port signal with input pulse at 48.5s.

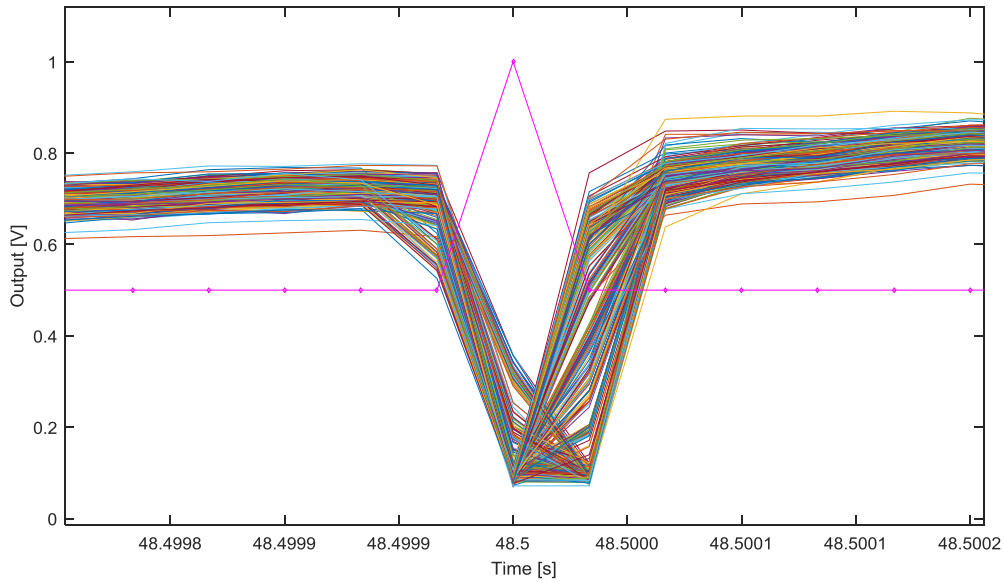

Figure 42: Probe data and Synch. port signal with input pulse at 48.5s, first pulse only.

5.12 BISTs

In order to debug and test the different hardware components of the recording system, a number of Built-In Self-Test (BIST) features are available (Figure 43). A subset of these tests can be performed during neural recording (except T4, T6, T8a and T9a).

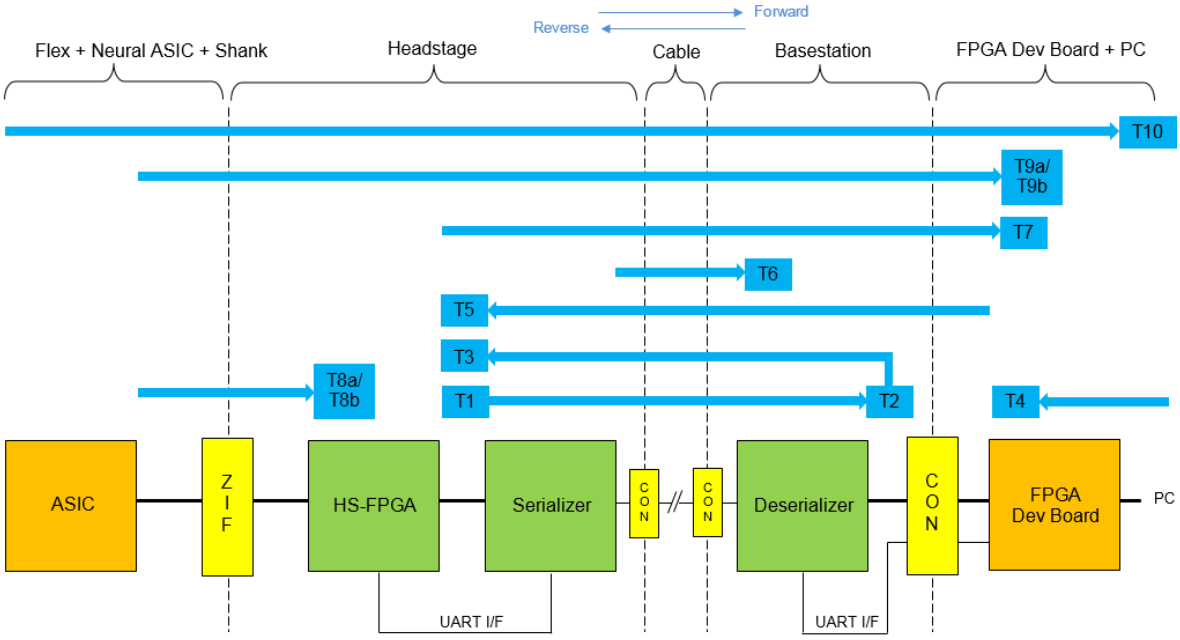

Figure 43: Overview of BISTs for the recording system.

The following Built-In Self-Tests are implemented in the recording system:

- Test T1: HW Heartbeat HS (LED blue)
- Test T2: HW Heartbeat BSC board forward (LED blue)
- Test T3: HW Heartbeat HS reverse (LED green)
- Test T4: PC ↔ FPGA dev board interconnection test
- Test T5: SW Heartbeat HS (LED orange)
- Test T6: SerDes PRBS test pattern
- Test T7: HS Neural Data test pattern
- Test T8a & T8b: Neural Data sync/count & test pattern check
- Test T9a & T9b: Full Neural Data sync/count & test pattern check
- Test 10 : Shank/probe → PC Neural Data path operational test

5.12.1 *BIST#1*

When the HS is powered on, the HS FPGA transmits a 1 Hz clock signal to the blue LED on the HS. This indicates that the HS is successfully powered over the microcoax cable and that the boot code was successfully loaded to the HS FPGA. The LED remains blinking as long as the neural recording system is switched on.

5.12.2 *BIST#2*

The clock signal driving the blue LED on the HS for BIST#1 is transmitted over the microcoax cable to the BSC board where it is triggering the blue LED labelled NEURAL_HBEAT. This test verifies the forward data link between HS and BSC board. It is successful if both blue LEDs are blinking at the same frequency. This test is running continuously when the recording system is switched on.

5.12.3 *BIST#3*

The BIST#3 test verifies the backward data interface between HS and BSC board. The signal that arrives on the BSC board driving the blue LED during BIST#2 is returned via the data interface over the microcoax to the HS where it drives the green-yellow LED.

The test is successful if the green-yellow LED on the HS is blinking simultaneously with the blue LED.

The BIST#3 test is automatically initiated when the recording system is powered on. It is terminated when the user opens a configuration and data link from the PC.

5.12.4 *BIST#4*

The system performs a loopback test of the TCP-IP connection to the FPGA dev board, an R/W test from FPGA to SDRAM memory, and an LCD test.

5.12.5 *BIST#5*

This test verifies the embedded datapath over the serializer/deserializer link between FPGA dev board and HS.

The FPGA dev board writes and reads registers on the HS FPGA. 2 Tests are performed:

1. The BSC board reads the HS board version and HS FPGA version, and returns the acquired values to the user.
2. The BSC board transmits a heartbeat signal (+-2Hz clock) to the HS, which drives the orange LED on the HS. The test is successful if the orange LED on the HS is blinking.

The test is started via an API function. The version register values are returned to the user.

5.12.6 *BIST#6*

The serializer/deserializer interface between HS and BSC board has an integrated PRBS generator, which allows bit error rate testing of the interface.

The test is started and stopped via API functions. The error counter value is returned to the user.

1. Start BIST #6 via API function neuropix_startTest6().
2. Stop the test and return result via neuropix_stopTest6().

5.12.7 *BIST#7*

BIST #7 verifies the connectivity between HS FPGA and serializer, between serializer and deserializer, and between deserializer and FPGA dev board. The HS FPGA uses idle times in the neural data to transmit a known test pattern. The FPGA dev board verifies the received test pattern and counts eventual errors.

The test is started and stopped via API functions. Separate API functions are available to return the number of errors as well as on which interface (SPI) line between the serializer/deserializer and the FPGA dev board the error occurred.

The test is started and stopped via API functions. Separate API functions are available to return the errors.

1. Start the BIST #7 via API function neuropix_startTest7().
2. Read results while the test is running via API functions
    a. neuropix_test7GetErrorCounter(): number of errors observed
    b. neuropix_test7GetTotalChecked(): number of SPI packets
    c. neuropix_test7GetErrorMask(): which SPI line between FPGA and serdes
3. Stop the BIST #7 via API function neuropix_stopTest7().

5.12.8 *BIST#8*

BIST #8 verifies the interface between the probe and the HS. There are 4 SPI lines between probe and HS, which are tested separately after selection by the user. The HS checks the data received from the probe. Two modes can be selected:

a. The probe is placed in a test mode in which the neural data is replaced by a fixed test pattern. The HS checks the SYNC and COUNTER values, as well as the test pattern, which is 100% of the data transmitted by the probe.
b. The HS checks the SYNC and COUNTER values as transmitted from the probe, which is about 30% of the data transmitted by the probe. Neural data itself is not checked.

The test is started and stopped via API functions. A separate API function is available to return the number of errors.

1. Set probe in 'Recording' mode using API function neuropix_mode().
2. Set NRST to logic 'high' using API function neuropix_nrst().
3. Start the BIST #8 via API function neuropix_startTest8().
4. Read results while the test is running via API function neuropix_test8GetErrorCounter().
5. Stop the BIST #8 via API function neuropix_stopTest8().

5.12.9 *BIST#9*

BIST #9 verifies the interface from probe to FPGA dev board. The test is similar to BIST#8, but now the data is checked by the FPGA dev board. All 4 interface lines from the probe are checked. Also here, 2 modes can be selected:

a. The probe is placed in a test mode in which the neural data is replaced by a fixed test pattern. The HS checks the SYNC and COUNTER values, as well as the test pattern, which is 100% of the data transmitted by the probe.
b. The FPGA dev board checks the SYNC and COUNTER values as transmitted from the probe, which is about 30% of the data transmitted by the probe. Neural data itself is not checked.

The test is started and stopped via API functions. A separate API function is available to return the number of errors. A specific procedure needs to be followed since it is necessary to have the probe transmit data:

1. Set probe in 'Recording' mode using API function neuropix_mode().
2. Set NRST to logic 'low' using API function neuropix_nrst().
3. Reset the datapath using API function neuropix_resetDatapath().
4. Start the BIST #9 via API function neuropix_startTest9().
5. Set NRST to logic 'high' using API function neuropix_nrst().
6. Read results while the test is running via API function neuropix_test9GetResults().
7. Stop the BIST #9 via API function neuropix_stopTest9().

5.13 Tools

The following executables can be called from the command line interface.

5.13.1 *Csv_gen_electrode.*

The tool converts the binary .npx format to an easily readable .csv file. Type csv_gen_electrode without any argument to get instructions on how to use the function.

5.13.2 *Record_to_npx_electrode*

The tool can be used to acquire data from the probe to a .npx file for a defined time. The tool configures the probe such that it transmits data. Type record_to_npx_electrode for instructions on how to use the function.