MEA2100 User Guide

Introduction

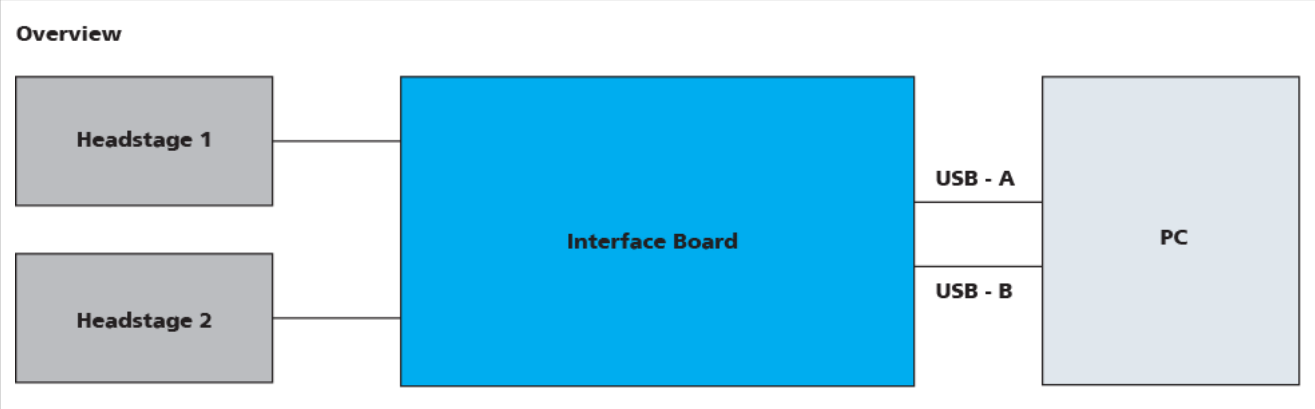
System Overview

The MEA2100 system consists of an interface board and one or two headstages.

The interface bord consists of:

- Two USB connectors for PC connection
- Two ports for one or two headstages
- 16 digital input and 16 output ports for triggering
- Eight additional analog inputs for monitoring
- DSP debug port
- Audio out
- A DSP
  - Who has access to the measured data stream for additional promt data calculation
  - Optional Stimulation control

The DSP can be programmed by the user, for example to do real-time signal analyses and programm feedback via the stimulation channels. A debug port is provided at the same connector as the analog inputs.

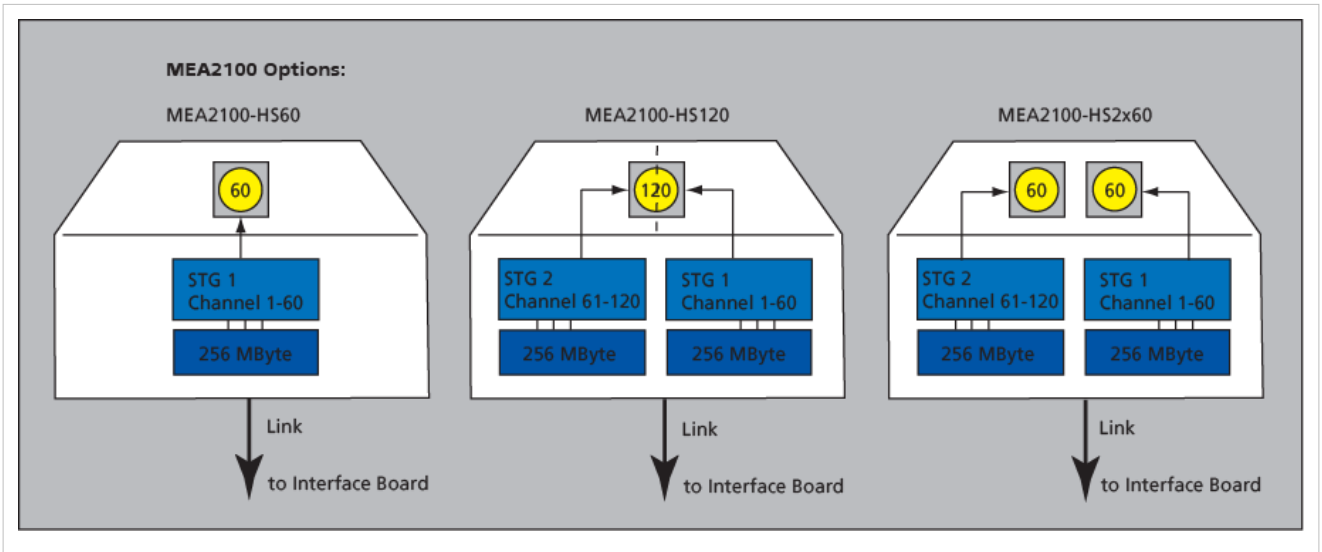


The headstage consists of:

- Adapters for different MEAs (see options below)
  - Electronic for measurement
  - Stimulation of up to 120 electrodes

The headstages connect to an interface board.

The overview about HS options:



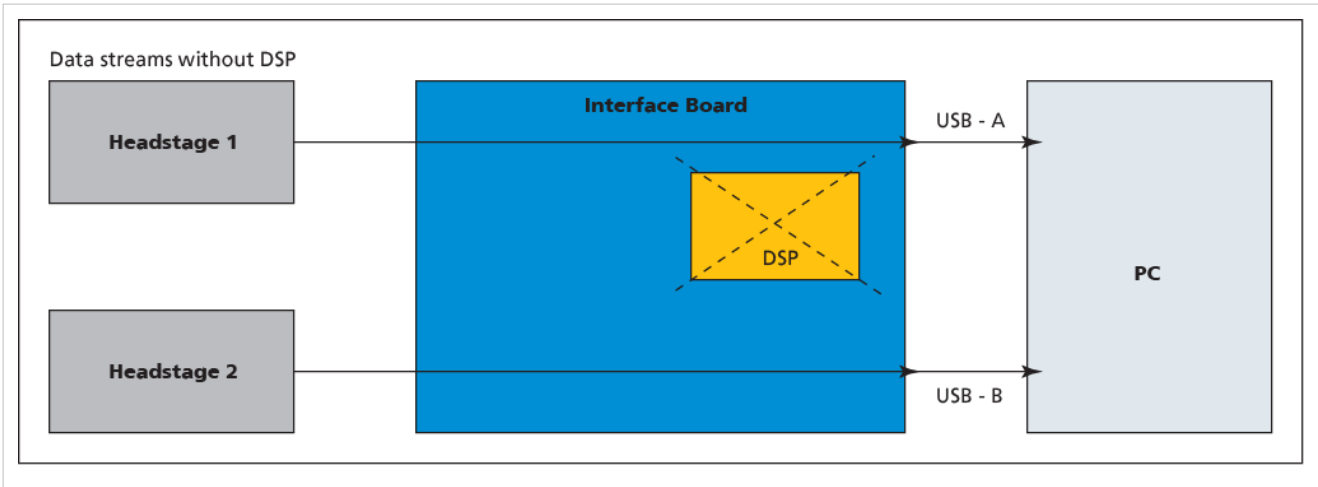
Getting Started

To write code for the DSP, the following tools are needed:

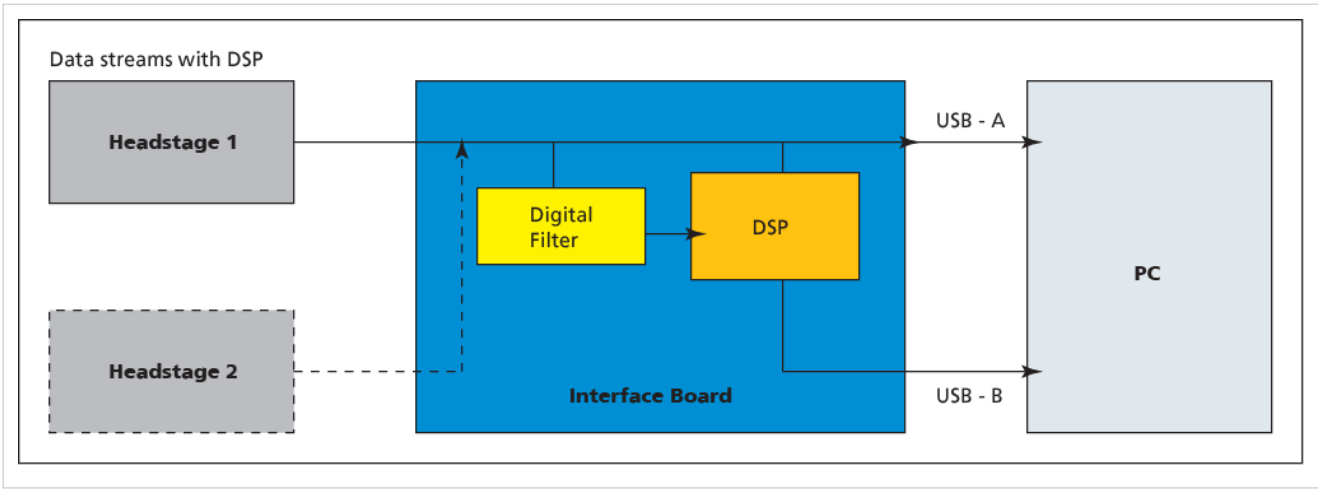
- Code Composer Studio IDE V8 from Texas Instruments http://processors.wiki.ti.com/index.php/Download_CCS
  - You need to install the version 7.4.x of the C6000 compiler: Menu -> Help -> Install Code Generation Compiler Tools...; TI Compiler Updates
    - C6000 compiler 7.4.x; Documentation: http://www.ti.com/lit/ug/spru187u/spru187u.pdf
    - compiler 8.2.x does not support the TMS320C6454
  - Driver_Bh560v2: v1.0.0.7 might be needed for Windows 10 due to signature problems
- ActivePython V2.7.x http://www.activestate.com/activepython/downloads
- Visual Studio 2017, there is a Community Edition available from Microsoft https://www.visualstudio.com/de/free-developer-offers
- A System Trace Emulator is of great help to debug DSP code
  - BH-XDS-560v2 https://www.blackhawk-dsp.com/store/12161.html
  - BH-USB-100v2D https://www.blackhawk-dsp.com/store/12139b.html

DSP

The default setup of the system routes the data without DSP interaction from HS 1 to USB A and from HS2 to USB B Port. In this case the DSP is only able to monitor the sweep data.



The DSP can be programmed with user code to access the data stream from the headstages and the onboard ADCs of the interface board. The datastream can be send to the USB ports towards the PC. The DSP can also configure all parts of the MEA2100-System, including stimulation and trigger, exept the USB data stream configuration to the PC. By this a realtime feedback with low latency can be build.



The DSP used in the MEA2100 system is a TMS320C6454 [1] from Texas Instruments running at 819.2 MHz. He has access to the MEA2100 system via its external memory interface (EMIF). Access to the system is via three parts.

- The Configuration of the MEA2100 is done with the register based interface. Therefore the DSP has the MEA addreses mapped into the range from 0xA0000000 to 0xAFFFFFFF. The different components of the MEA2100 use each a subrange of this address region.#Register Map In the FB_Example Project, `MEA21_init.h` defines Macros which can be useed to access these registers:

```
#define READ_REGISTER(reg)            MEA21_REGISTER(reg)
#define WRITE_REGISTER(reg, value)    MEA21_REGISTER(reg) = value
```

- Sweep and Digital data is transfered via a fifo-type interface, mapped into the DSP address range starting from 0xB0000000.
- One of two options of controling the stimulation is the direct streaming mode. This can be done via the streaming port using address 0xC0000000.

Data Aquisition

General Overview

The MEA2100 System can measure analog signals from 120 electrodes per Headstage at 50kHz with 24bit resolution.
On the interface board are 8 additional analog inputs for monitoring purposes(also 50kHz and 24bit resolution).
Additional to the analog signals are several digital signals(configurable DSP in data).

DSP Reception of Sweep and Digital data

Sweep and Digital Data is available to the DSP through a FIFO type interface mapped to the address 0xB0000000 in the DSP memory map.

Each time (once every 20 us), when the MEA2100 has a new sweep of data available from the Datasources, the signal line (GPIO4) to the DSP will be toggled (enabled in Interconnection Logic). After the DSP has received this signal, it can read the new sweep data either by multiple reads from address 0xB0000000, or the more convenient way is to set up a DMA transfer which automatically transfers all data from the sweep into DSP memory. After the transfer is complete, a DMA completion interrupt can be called.

The example code in the FB_Example project initializes and enables such a DMA transfer. In the example, each time after a new sweep of MEA2100 is avaible in DSP memory, the function `interrupt6()` is called. The new data is available to the user in the array `MeaData[]`. The data in this `MeaData[]` array automatically updates every 20 us when the DMA is running.

The setup and initialisation of the DMA is done in the `init_dma()` function in `MEA21_init.c`. Once it is up and running, it does the transfers automatic without CPU intervention, so no loose of CPU time in this mode.

One important Register for the readout of data is Register 0x400. This register controls which kind of data is transfered into the FIFO from MEA2100 to the DSP. It can enable data from Headstage 1, Headstage 2, or from the ADCs on the Interface board and some other helpfull data.

The line 94 of `main.c`

```
WRITE_REGISTER(DspIndataCtrl, DSPINDATACTRL_VALUE);  // Enable Irq and HS1 Data
```

enables the data transfer from the FPGA to the FIFO towards the DSP.

The ADC Data blocks have the format

```
HS1                122 datawords (Header, 120 Data, Counter)
HS2                122 datawords (Header, 120 Data, Counter)
IF                   9 datawords (Header,   8 Data)
HS1 Filtered Data  122 datawords (Header, 120 Data, Counter)
HS2 Filtered Data  122 datawords (Header, 120 Data, Counter)
```

which means, when you have enabled data from Headstage 1 (HS1), you will get 122 32-bit Datawords per sweep, where the first dataword is a header word, followed by 120 32-bit datawords of ADC data (Format is signed, the ADCs have 24 bit resolution), followed by a 32-bit counter which enumerates the sweep.

So, when data from Headstage 1 is enabled there will be at `MeaData[0]` the header, at `MeaData[1]` the data from the first ADC (MEA Hardware Channel Nr. 1) and so on.

The pointers in the `irq.c` example are intended to help to access the ADC data, HS1_DATA_OFFSET is defined to 1, because this is the location of the data from the first data channel.

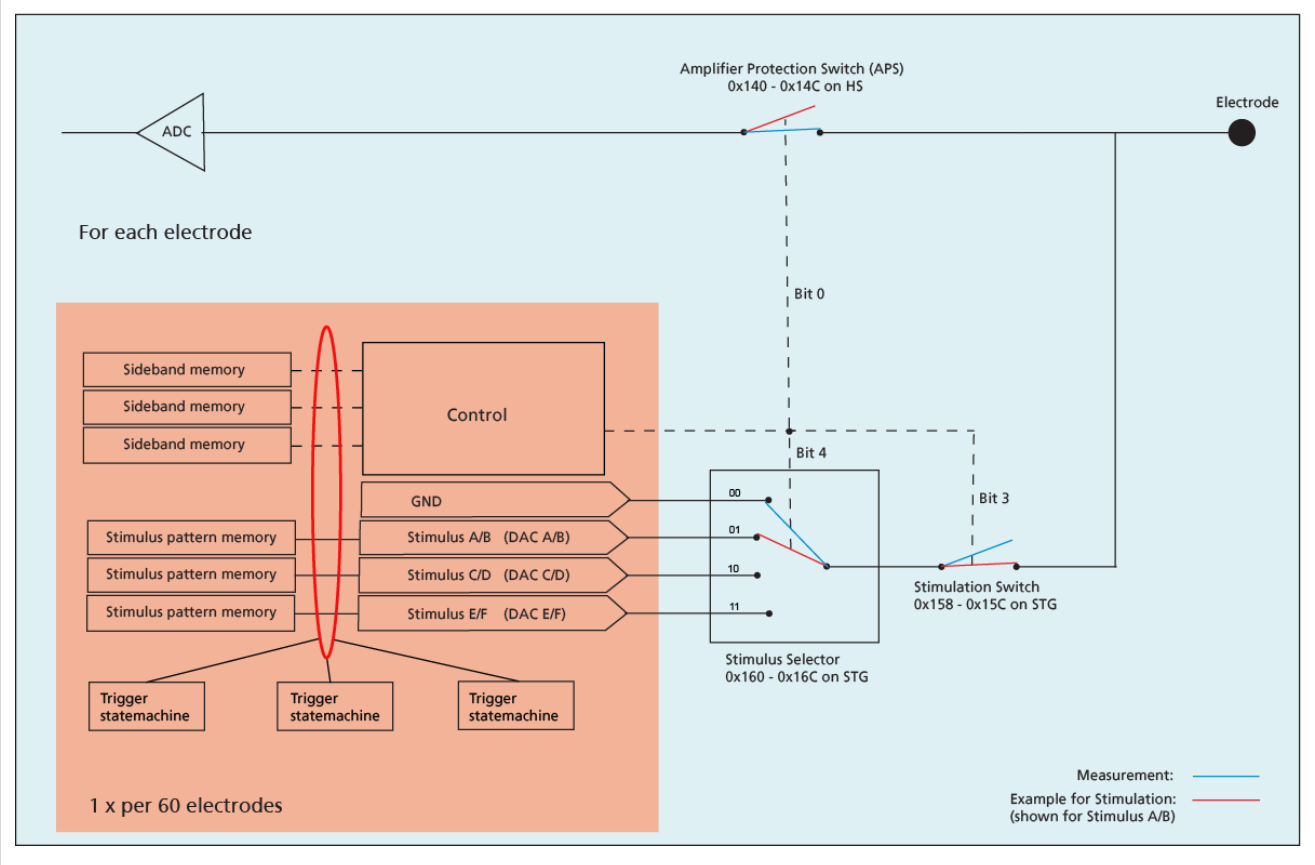
```
Int32* restrict HS1_Data_p = (Int32 *)&MeaData[HS1_DATA_OFFSET];
```

`HS1_Data_p` is a pointer to the location within `MeaData` where the ADC data is located. When the code in the DSP needs to do some calculation on the sweep data, the data of interest should be copyed to a memory location under its control. You always need to remember that `MeaData` will be changed as soon as new sweep data is transfered by the DMA.

The Digital Data blocks have the format

```
Digital Data 28 datawords (Header, 27 Data)
Timestamp     3  datawords (Header,  2 Data)
```

Stimulation



General Overview

Each headstage, which can measure from and stimulate up to 120 electrodes has two seperate modules (STG 1 and STG 2) for the stimulation. The first module is in control of electrode channels 1 to 60, the second module for electrode channels 61 to 120. Each of the two modules has access to three DAC pairs and three sideband channels.

There are two options of controling the stimulation in the modules:

- Configure Stimulus-Generator modules
- Direct streaming mode from DSP via streaming port

Within each module, the three DAC pairs and three sideband channels can be arbitrary controlled by three trigger statemachines.

These statemachines handle the stimulus pattern read out units (DAC/SBS) to whom they are assigned to (reg:0x104-0x108). The stimulus pattern is stored in onboard memory on the MEA2100 headstage where each module has its own independent memory. For the stimulation (DAC/SBS) the memory is organized in segments and blocks to support the stimulus with different stimulus pattern lists, to enable quick switching between lists of

prepared patterns. The DACs update their output value every 20 us, resulting in an output rate of 50 kHz. Corresponding for the analog patterns, additional information is stored in memory regions called sidebands. These sideband data contain digital information for controling the switches and muxes to enable stimulation. The update of this data is also every 20 us.

- The electrical parameters are:
  - Voltage Mode: Range: +- 12 V, Resolution: 0.571 mV / digit (571 uV/digit)
  - Current Mode: Range: +- 1.5 mA, Resolution: 0.05 uA / digit (50 nA/digit)

The configuration of the Stimulus Generator consists of three parts:

- The stimulus pattern upload
- The electrode configuration
- The trigger setup

The configuration of the Direct Streaming mode can be done in two ways:

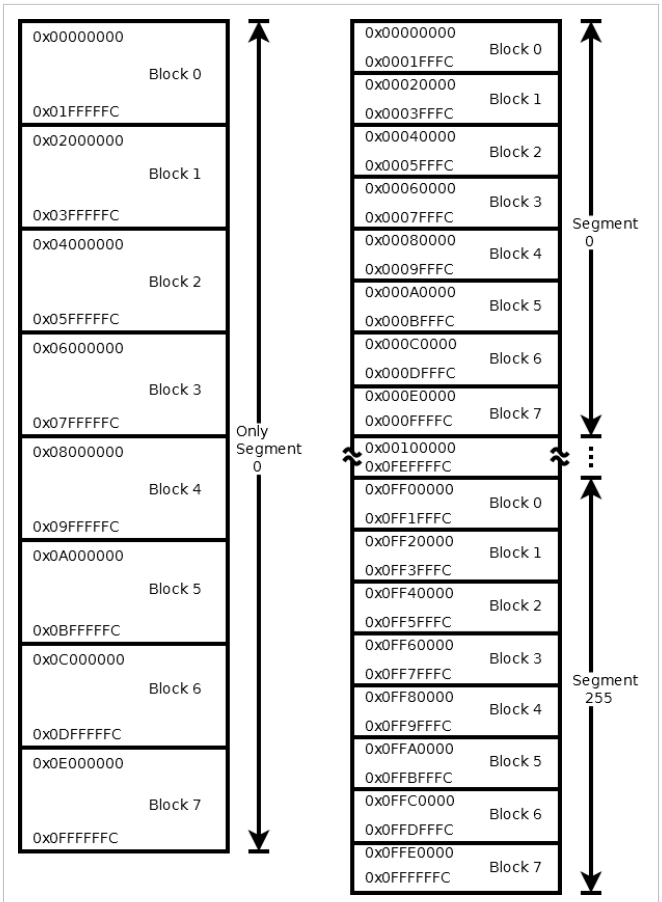
- When controled by Computer SW via DSP
- When fully controled by DSP

Stimulus Pattern Upload

There are eigth stimulus pattern read out units to read the stimulus pattern from the assigned memory block and support the data to the stimulation logic. In the default configuration, only six units are in use, where three support the data for the DAC pairs and three for the sideband signals.

Setup: Before the patterns can be uploaded into memory, the available 256 MByte of memory need to be devided in blocks and segments. These blocks and segments need to be assigned to the eigth units. One option is to divide all available memory into eigth blocks and ONE segment. A more advanced option is to subdivide the total memory into 256 segments, and then subdivide the memory within each of these segments into the eigth blocks. This mode, called the segmented mode, allows to predefine and upload up to 256 independant stimulation patterns and quickly select amoung them within microseconds. The mode will be selected by a write to Memory Control Register (0x200) on each of the STG subsystems. Setting bit 28 one segment will be set up automaticaly. Setting bit 29 the memory is subdivided into 256 segments with interal logic.

Alternatively the size of each block and each segment can be subdivided individual with the needed amount of memory. This is controlled by the registers 0x200 to 0x2f0 and need to be done manualy. There the start pointer for each of the memory blocks need to be inserted. For example the memory block for DAC A and B (default setting of Register 0x1D0) is controlled by the registers 0x200 to 0x210. Register 0x200 is used to select a segment, and then the start address for this block in the selected segment is set via registers 0x204. In default Segment 0 is used. When segmented mode is chosen another segment is selected via register 0x200. Again the start address of the selected block is set via register 0x204. So register 0x200 selects the segment to be defined/modified and then the other registers reflect the corresponding block start pointer of the segment.



Memory Layout options

After the intendet memory layout is set up, each of the DAC and sideband memory blocks can be filled with data. Data is written to the blocks in a FIFO type mode via the Write DAC and Write SBS Data Registers (0xF20-0xF3C). Each write to one of these registers appends one 32bit data word to the corresponding block in the selected segment. When a data word is written to the Data Registers at address from 0xF40 to 0xF54, the pointer of the corresponding block is first set to Start. The segment will be changed as well in Register 0x200 (0x220,0x240,...,0x2E0). If the number of vectors for a stimulus pattern is more than the memory space allocated for one block, no hardware will limit the write to the memory (overlapping pattern memory). So the data of an other block will be overwriten! This will lead to unpredictable behavior during operation!

The 32 bit data word is described in Data Vector decoding.

The data vector (000) is either a value with a 16 bit amplitude for DACs or 16 bit Sideband Signal value for Sideband Channels. Bit 26 defines the timebase of this vector. A '0' defines that the value will be valid for one 20 us frame until the number of repeats (bits 25-16) will be decreased by one. If bit 26 is '1' the value will be valid for 1000 times 20 us until the decrement will be done.

Two kind of loops can be used. A single vector loop (001) and a long loop vector (010 in combination with 011) with two vectors keeping the information. For both types of loops the "number of repeats" are one based. Whereby a 0x1 will lead to ignore the vector and a 0x2 will repeat once. A value of 0x0 means to loop forever! The address offset is also one based. A 0x1 will jump backward one vector before the loop vectors. So nested loops for repetition of (sub-)blocks of data can be created. This avoids to store one stimulus multiple times.

DAC Units

The stimulation module for the 60 electrode channels is subdivided in two halfs of 30 electrodes, where each half has its own set of three DACs. In default configuration these six DACs are combined into three DAC pairs, so that it looks like that there are three DACs available for each of the 60 electrodes.

For advanced applications, these DAC pairs can be broken up, allowing stimulation with three DACs available for each block of 30 electrodes. The DACs which are available to the lower 30 electrodes in a module are named DAC A, C and E, the DACs which are available to the upper 30 electrodes in a module are named DAC B, D and F.

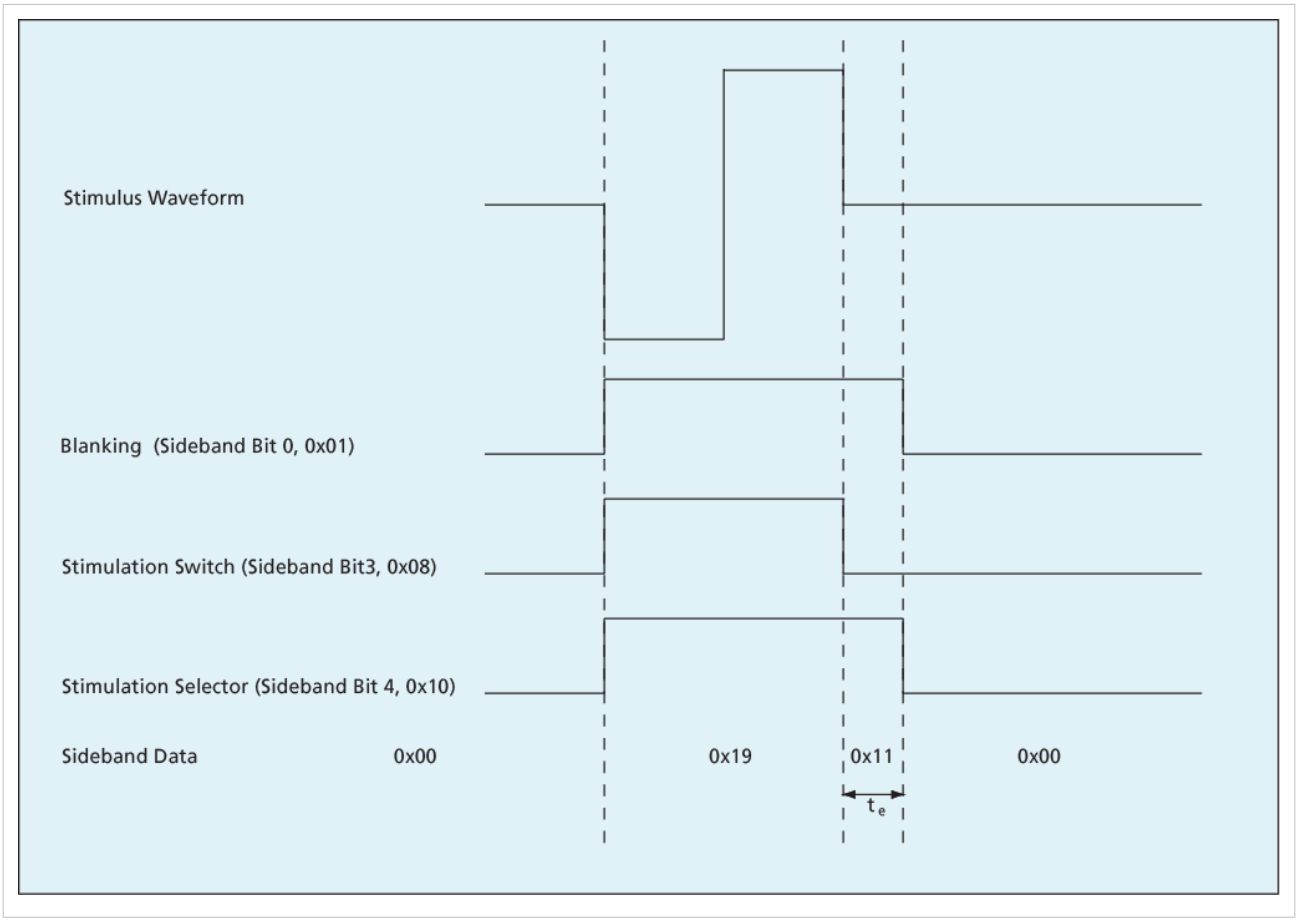
In default configuration, DAC A and B form the DAC AB pair, likewise DAC C and D form the DAC CD pair, and DAC E and F form the DAC EF pair, making it look like there are three DACs named AB, CD and EF. Both DACs out of a DAC pair take their data out of the same memory block, thus always have the same analog output.

The DAC pairs are broken up by assigning each DAC to its own stimulus pattern read out unit of the available eigth stimulus pattern supporting units. When all six DACS are to be used individually there are two units left to be shared among the three sidebands. The register which controls the DAC to stimulus pattern read out units is 0x1D0. For Sideband configuration it is register 0x1D4.

Sideband Data

A Sideband has two functions:

- These Sideband Signals (SBS) control the switches which connect the Stimulation DACs to the electrode and it controls the Amplifier Protection("Blanking") of ADC data while a Stimulus pulse is active. Each function is assigned to a bit in the 16 bit wide data stream:
  - Bit 0: Amplifier Protection Switch on Headstage/Blanking
  - Bit 3: Stimulation Switch Close
  - Bit 4: Stimulus Selector Enable
  - Bit 8: List Mode config ID increment on/to the Interface Board or Bit 15-8 List Mode config ID when source of ID is switched to SB bits
- It can be used to send data to the USB, to the DSP or to the digital outputs on the Interfaceboard which are synchronous to the running stimulation.



The diagram shows an example stimulus pattern together with the sideband datastream. As shown in the drawing, the Stimulation Switch can open with the end of the Stimulus. The Blanking signal should stay active for some additional time te, recommendations are 20 us for Voltage Stimulation and 20 us for Current Stimulation. Likewise, the Stimulation Selector should be kept for some additional time te, here the recommendation is 20 us for Voltage Stimulation and 20 us for Current Stimulation.

Electrode Configuration

To configure an eletrode for stimulation, each of the 120 electrodes has to be either assigned to one of the three DACs within its STG subsystem, or it can be in an "inactive" configuration. This is controlled by register 0x160-0x16C, the Electrode DAC Multiplexer assignment Register. These registers sets the Stimulus selectors to one of the four inputs. Two bits for each electrode either configure the electrode to be stimulated to DAC A/B, C/D or E/F. The bit pattern "00" configures the selector into an "inactive" mode, in which it is configured to fixed ground.

Behind the multiplexer, the Stimulation Enable switch connects the DAC selector output with the electrode. This will be configured in register 0x158-0x15C.

For Testing or for Measurement electrodes, each electrode can be set in "manual" mode. With this the DAC Multiplexer and the Stimulation Enable switch are set imidiatly in the state which is defined by register 0x160-0x16C for the DAC settings and 0x158-0x15C for the Stimulation Enable. During stimulation, when the system should be controled by the sidband signals, the corresponding stimulation electrode needs to be set into dynamic (automatic) mode (register 0x120-0x12C).

In default setup the sideband 1 is assigned to DAC A/B, sideband 2 is assigned to DAC C/D and sideband 3 is assigned to DAC E/F. For advanced setup these defaults can be changed in register 0x154. For example all DACs can have assigned sideband 1 to it.

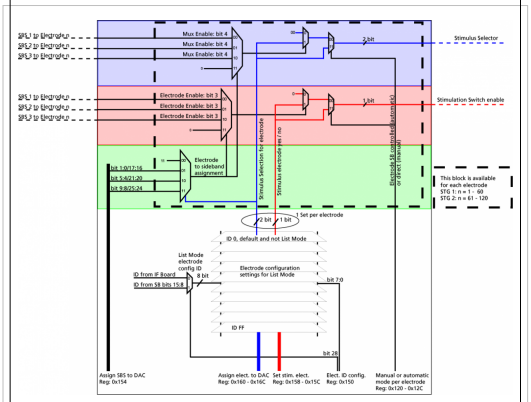
When an electrode is in automatic mode assigned to one of the three DACs, the DAC Select and Stimulation Enable register do not directly configure the state of the switches. Only when the Sideband datastream is running, the corresponding SBS vector to the current DAC vector determine the state of the switches. The default state, which is used when the sideband is not running or when the corresponding bit in the sideband datastream is zero, is for the Multiplexer to use its ground connected pin to be selected and the Elektrode Enable Switch is open. The default state of the switches, which is used when the signal from the electrodes is measured, is shown in blue in the schematic overview above. The sideband is a 16 bit wide data stream. Bit 0 of this data stream controls the Amplifier Protection Switch and the Blanking, Bit 3 controls the Stimulation Switch and Bit 4 is in control of the Stimulus Selector.

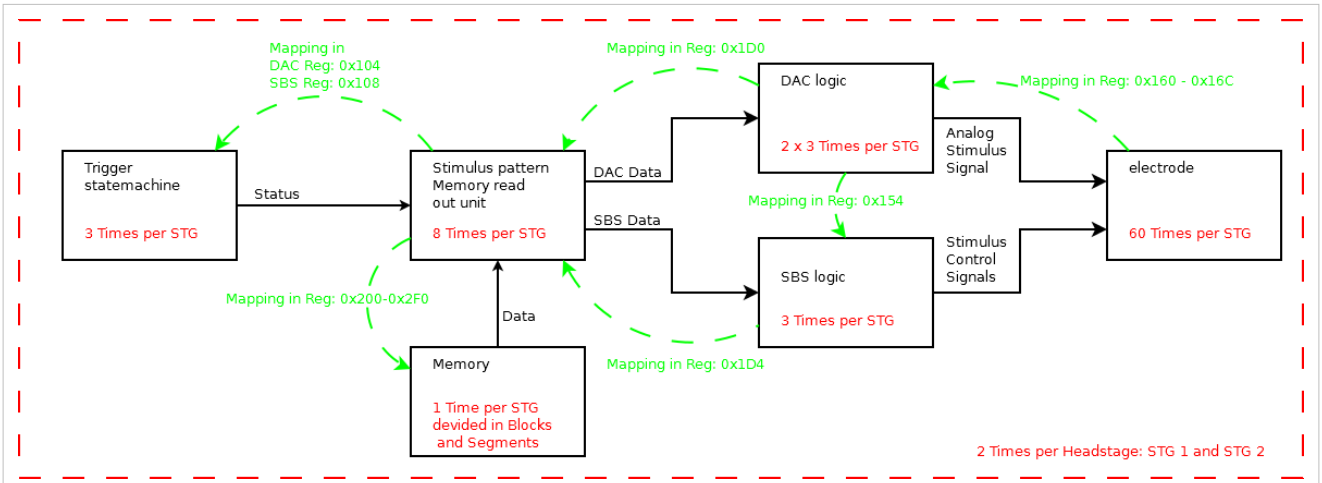
Electrode setup advice

- Decide wether your electrode is used as stimulation or measurement electrode (set corresponding bit in register 0x120-0x12C and 0x158-0x15C)
- Assign your stimulation electrode to a stimulation channel (DAC A to F) (set corresponding bits in register 0x160-0x16C)

Stimulus channel setup advice

- Assign a sideband Channel to a stimulation channel (set in register 0x154)

Control:	Decoding for stimulation channel (DAC) at stimulus selector (register 0x160-0x16C)
 Sideband Singnal Control for electrode n	00: GND 01: DAC A/B 10: DAC C/D 11: DAC E/F



STG Data Path and configuration dependencies

List mode for advanced stimulation

There is the possibilty to predefine and quickly switch the electrodes between up to 256 different stimulation configurations. The DAC select and Stimulation Enable registers are actually an array of 256 independent configurations. During setup the register 0x150, the Electrode configuration ID CTRL register controls which configuration is to be defined. Here bits 0-7 define which one of the 256 possible configrations is selected for setup editing.

Bit 28 in the Electrode configuration ID CTRL register control how the configuration is selected during stimulation. This bit can either be "0" for taking configuration ID from the Interface Board. When this bit is set to "1" the "internal" configuration ID is active. Then bits 8 to 15 in the sideband datasteam define the active electrode configuration. In "external mode" the List Mode configuration ID is generated in the Trigger Block of the Interface Board.

Standard Trigger

Standard Trigger Setup

For controling the stimulation three trigger statemachines are implemented. In the default configuration, each trigger has control of two stimulus pattern read out units. These units support data for one DAC pair and one sideband. So this controls one stimulus pattern and its corresponding digital information 0x104-0x108.

An example where more than one DAC is assigned to one trigger is, when multiple electrodes are to be stimulated, grouped into two blocks, where each of the two blocks has a different stimulation signal amplitude. To accomplish this, two DACs can be assigned to one trigger. As long as the timing of the two stimulation patterns for two stimulus groups are the same, they can share one sideband resulting in a setup where one trigger controls two DACS and one sideband.

When the two stimulation groups have differences in timing it is useful to use a seperate sidebands for each of these groups, thus resulting in a setup where one trigger controls two DACs and two sidebands.

The assignment of the three DAC pairs and three sidebands to the three triggers within each block is controlled by registers 0x104 for the DACs and by register 0x108 for the sideband channels. With this setup the controlling trigger for each DAC and sideband has to be choosen.

Start and Stop of the stimulation is always controlled at the level of the triggers, so that all DAC and sidebands which are grouped together to a specific trigger are started and keeped running in sync.

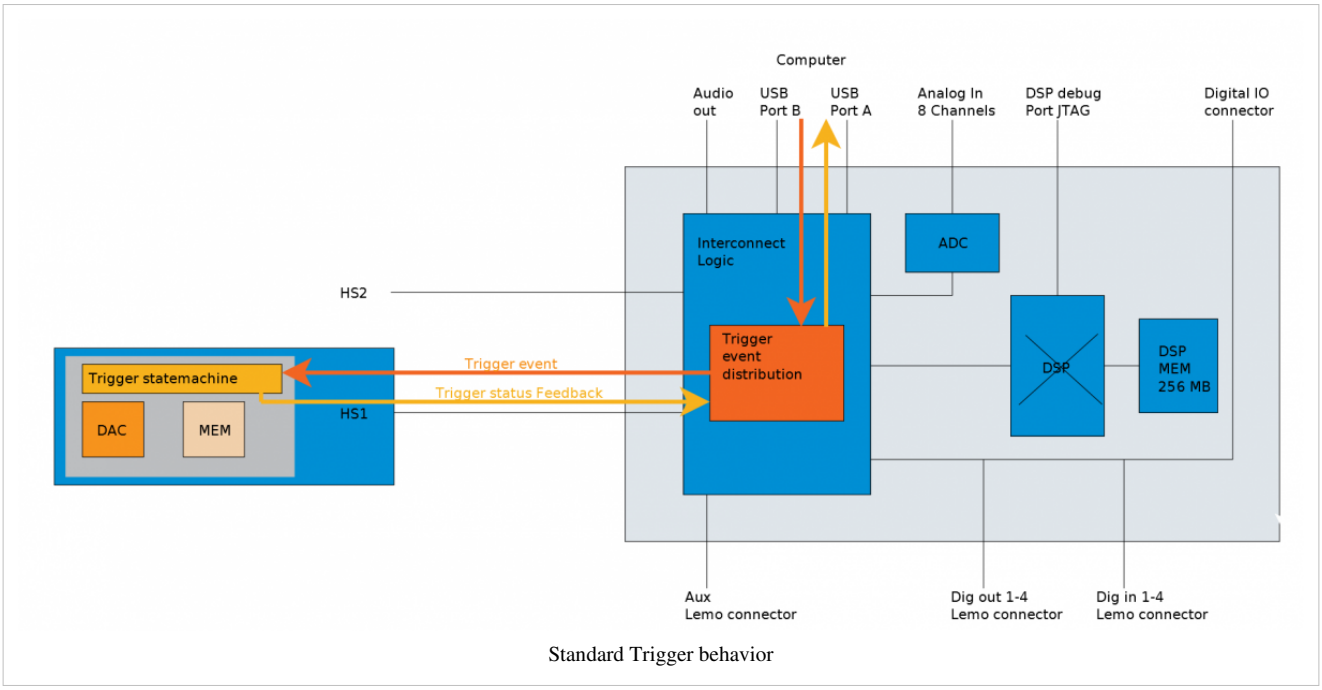
Start Trigger

For the trigger to work, the first bit(0) in the Trigger Control Register 1 (0x200) **on the Interface board** has to be set to 1. This enables the trigger event packet to propagate from the interface board to the stimulus generators.

The Trigger ID registers (0x218 to 0x244) define for each trigger, which Stimulus-Pattern Memory-Segment to use when the corresponding trigger is startet. For single segment mode leave this register at the default value of 0.

To actually start a trigger manually, write a "1" to the bit which corresponds the trigger number in the Trigger Event Status register 0x214. For example to manually start trigger 1, write a value of 0x00000001 to register 0x214.

To start a trigger with external signals or other sources a Digital Multiplexer for source select is implemented (0x280-0x2AC).



Standard Trigger behavior

Direct Streaming Mode triggered by SW

A further advanced mode for Stimulation is the DSP generated stimulus pattern direct streaming mode, which can be standard triggered and monitored by the computer SW. In this mode the trigger statemachine and memory on an STG will be bypassed and the DAC and SBS data will be send direct from the DSP to the stimulus logic. The setup for stimulus switch and selector still needs to be done as in standard mode.

Direct Steaming Trigger Setup

To detect a trigger event on the DSP the digital data stream to the DSP needs to be enabled reg 0x400 bit 12. In this digital data stream the 8th and 18th vector reflect the trigger event information

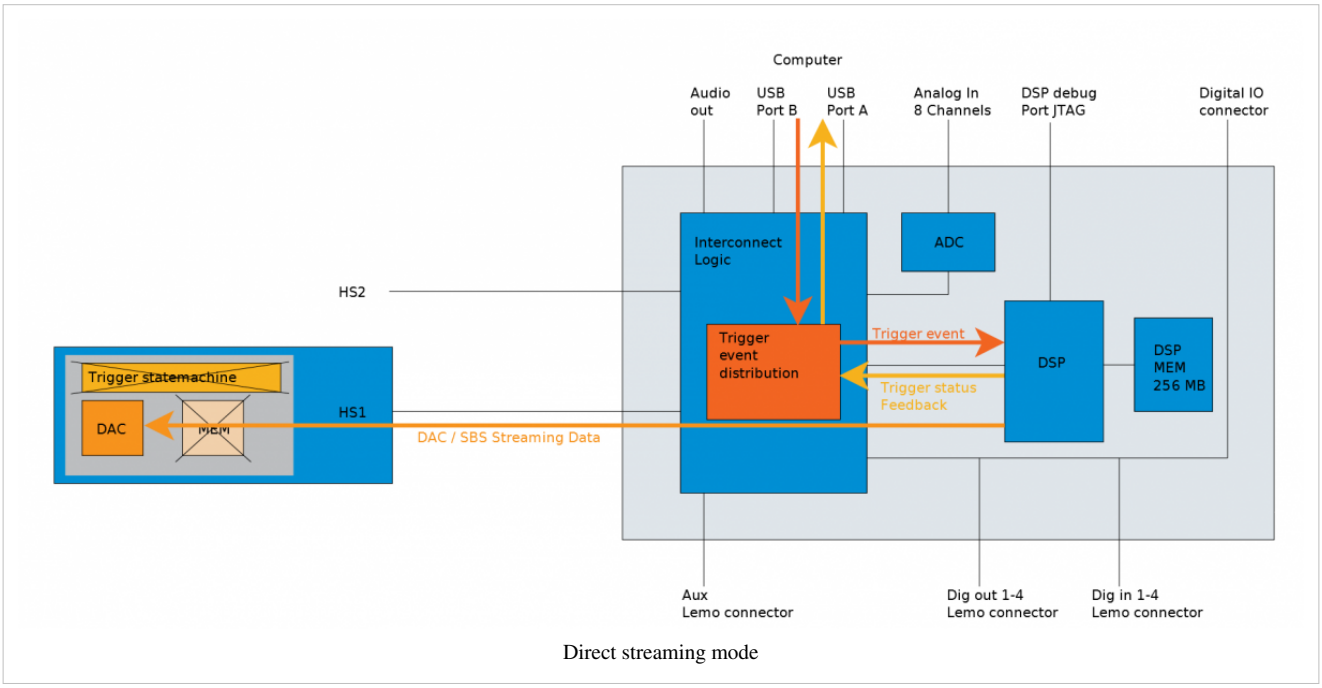
Direct Steaming Data path Setup

For controling the stimulation by direct streaming data to an STG, the data source for the DAC and SBS Data need to be switched to DSP. This needs to be done in register 0x100.

To start the distribution of the streaming data from the FIFO configure the register 0x430. After this the stimulus data can be send via the straming FIFO of the DSP. For this write 36 DWords to the address 0xC0000000 from the DSP.

- 24 DWords for DAC Data
  - 6 DWords to HS1 STG1 (DAC: A, C, E, B, D, F)
  - 6 DWords to HS1 STG2 (DAC: A, C, E, B, D, F)
  - 6 DWords to HS2 STG1 (DAC: A, C, E, B, D, F)
  - 6 DWords to HS2 STG2 (DAC: A, C, E, B, D, F)
- 12 DWords for SBS Data
  - 3 DWords to HS1 STG1 (SBS: 1, 2, 3)
  - 3 DWords to HS1 STG2 (SBS: 1, 2, 3)
  - 3 DWords to HS2 STG1 (SBS: 1, 2, 3)
  - 3 DWords to HS2 STG2 (SBS: 1, 2, 3)

Allways write 36 DWords, even when you switch only one STG to direct streaming mode, because the FIFO read logic expects 36 Data in the predefined order.



Direct streaming mode

Digital Multiplexer

There are multiple digital signals within the MEA2100 system, which represent internal states of the systems. These digital signals can be mapped

- To the digital outputs of the Interface board
- Into the digital datastream via the USB connection to the PC
- Used by the DSP
- With some restrictions used as triggers conditions for Stimulation
- With some restrictions used as list mode ID increment
- Trigger gated data start mode

The following digital signals are available for Digital Output and Digital Datastream to the computer and DSP

- Digital In bit 0 to 31; these are 32 bit taken from the rear side Digital Connector
- Digital Pulse Register bit 0 to 31; taken from Digital Pulse Register 0x700, valid for the duration defined in Register 0x704-0x710, started by a write to Register 0x700.
- Feedback Register bit 0 to 31; taken from Feedback Register 0x780
- Aux In bit 0 and 1; taken from the two lemo connectors on the Interface board.
- A fixed value of "0"
- A fixed value of "1"
- Trigger Status of the Stimulation Boards
- Any bit from all the Sideband Channels

There is a multiplexer for each bit of the digital datastream at register 0x880-0x8BC for USB connector A and register 0x8C0-0x8FC for USB connector B. For the digital output there is for each bit a register at 0x840-0x87C. There you can select its source from all selectable sources as descibed in the table above.

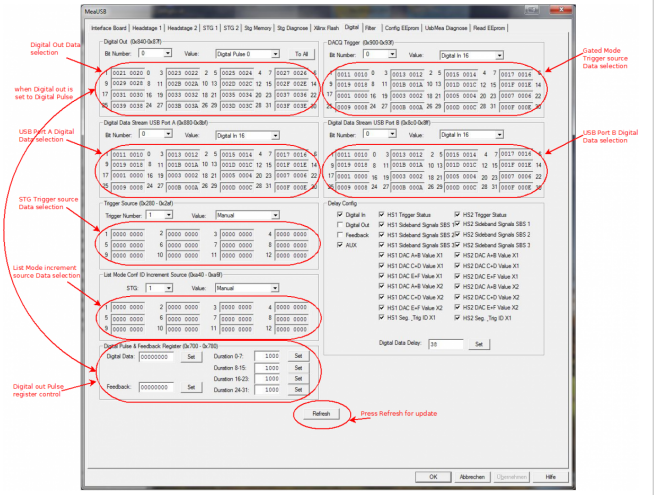
The multiplexer for the following digital signals are available to trigger the Stimulators

- A fixed value of "0"
- Digital In bit 0 to 31; these are 32 bit taken from the rear side Digital Connector
- Feedback Register bit 0 to 31; taken from Feedback Register 0x780
- Aux In bit 0 and 1; taken from the two lemo connectors on the Interface board.

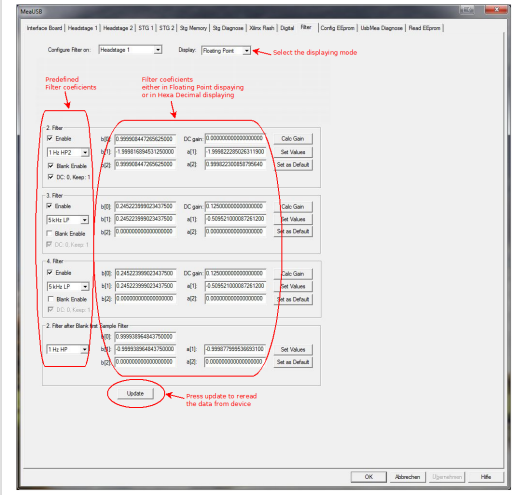
The multiplxer for each trigger are located in register 0x280 to 0x2AC.

Using McsUsbUpdate to review configuration

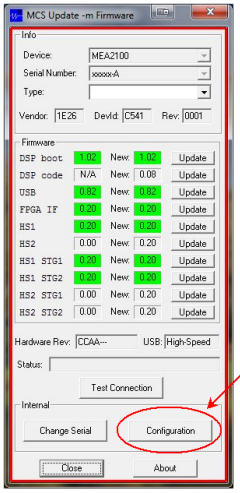
For reviewing the setup done with the DSP there is a program called McsUsbUpdate. With this program all relevant registers are reflected and according to the use case RO, RW, WO.



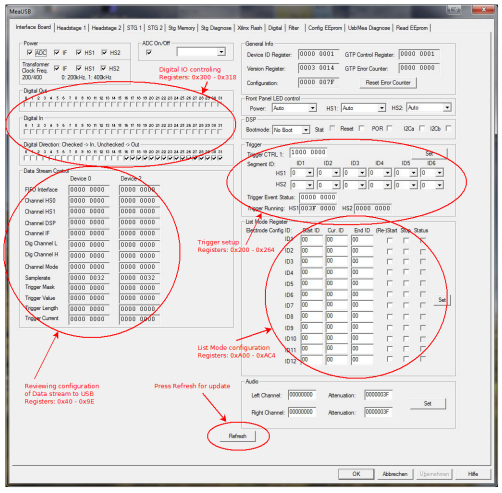
McsUsbUpdate Digital signal configuration screen



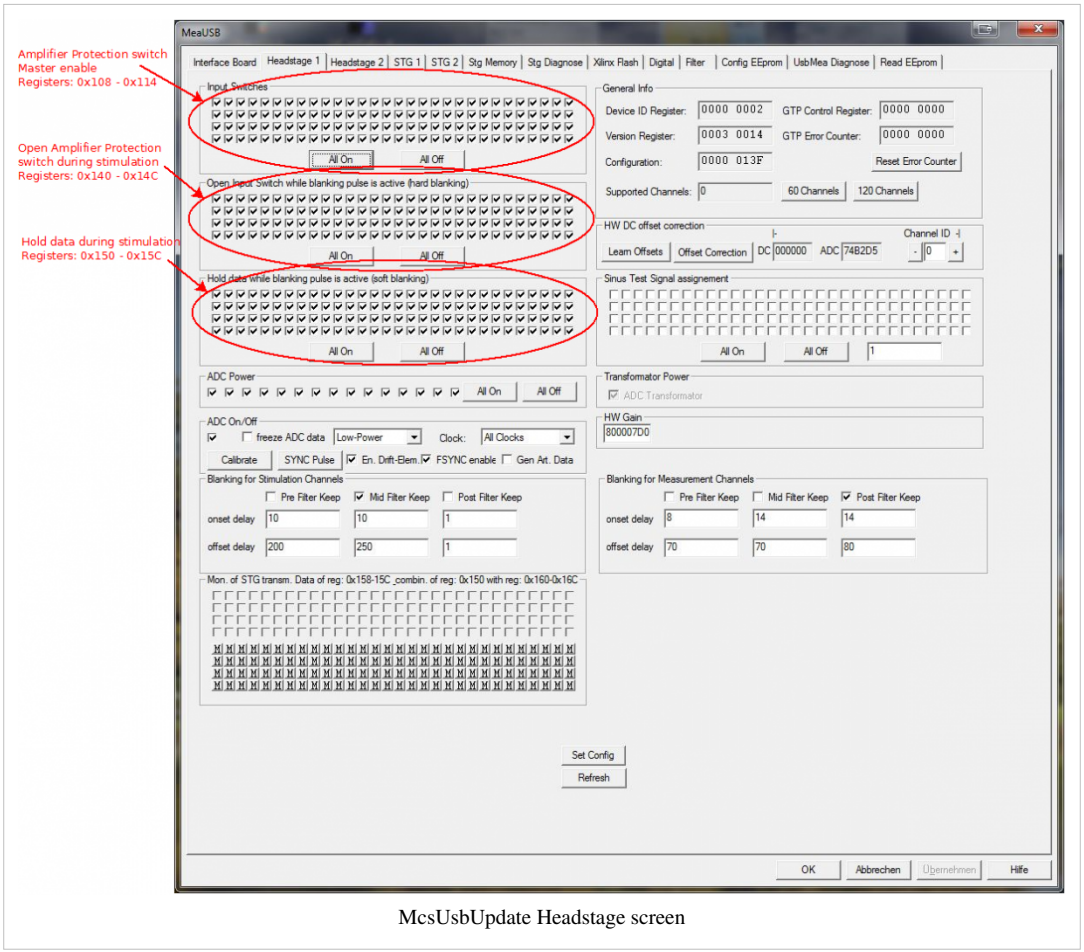
McsUsbUpdate filter settings screen



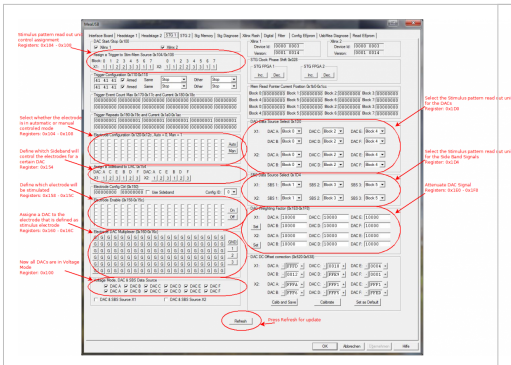
McsUsbUpdate



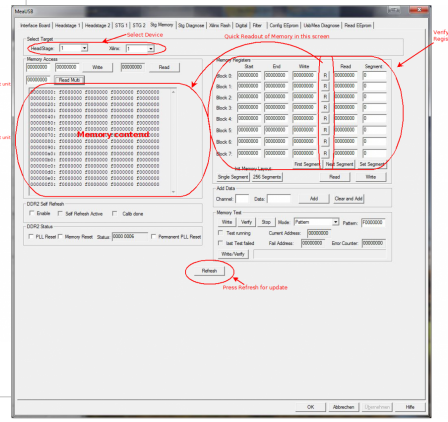
McsUsbUpdate Interface Board screen



McsUsbUpdate Headstage screen



McsUsbUpdate STG configuration screen



McsUsbUpdate STG Memory readout pane screen

Data Format for > 16 bit

24 bit Format, Little endian byteorder

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte 1								Byte 2								Byte 3								Byte 4								Byte 5								Byte 6							
LSB 0								MSB 0								HSB 0								LSB 1								MSB 1								HSB 1							

24 bit Format, MC_Rack freundlich, eventuell alle LSBs als Block nach allen 16 bit Kanälen

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte 1								Byte 2								Byte 3								Byte 4								Byte 5								Byte 6							
MSB 0								HSB 0								MSB 1								HSB 1								LSB 0								LSB 1							

32 bit Format, Little endian byteorder

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte 1								Byte 2								Byte 3								Byte 4							
LSB 0								MSB 0								HSB 0								sign extension							

Register Interface

Address Map (Address bits 15-12 determine the subcomponet)

When accessed from the DSP, all FPGA Registers are memory mapped to the address region from `0xA0000000` to `0xA000FFFF`.

```
0x0000 to 0x0FFF: Interfaceboard
0x1000 to 0x1FFF: Mailbox Registers on Interfaceboard
0x2000 to 0x2FFF: RAM Registers on Interfaceboard
0x8000 to 0x8FFF: Headstage 0
0x9000 to 0x9FFF: STG1 on HS 0
0xA000 to 0xAFFF: STG2 on HS 0
0xC000 to 0xCFFF: Headstage 1
0xD000 to 0xDFFF: STG1 on HS 1
0xE000 to 0xEFFF: STG2 on HS 1
```

Interface Board Address Map (Address bits 11-0), Base Address: 0x0000:

Access time is 0,85 us for writes and 0,85 us for reads.

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000: Device ID Register	0x00000001 = MEA 21 Interface FPGA																															
0x004: HW/FPGA Version Register	HW/Board Version																FPGA Version: 0x0101 = Initial Version															
0x008 : Configuration Register	Reserved																							Cy2notCy1	Reserved		Additional Boards attached, What kind of board?, ...					

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x010: GTP CTRL Register	Reserved																						HS 1 Link enable	HS 0 Link enable	Reserved						HS 1 Link up	HS 0 Link up
0x014: (RO)GTP Error counter Register	HS 1																HS 0															
0x014: (WO)GTP Error counter Reset	Any Value Resets the counters																															

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x020: FPGA Reset	Reserved																															FPGA Reset
0x024: Power Enable	Reserved																													HS2 PWR Enable	HS1 PWR Enable	IF Trafo
0x028: Trafo startup delay	Reserved								HS2 Delay								HS1 Delay								IF Delay							
0x02C: LED config Register	Reserved																						LED HS 1/2 and IF register mode enable		Reserved							LED HS 1/2 and IF output register

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x030: (WO)Flash Instruction Code Register	Reserved																						Stop instruction	Fifo Reset	Instruction Code							
0x030: (RO)Flash Status Register	Reserved																					FIFO_empty	FIFO_full	Statemachine busy	Flash Status Register							
0x034 : Flash Memory Address Register	Reserved								Flash Address																							
0x038 : Flash Data FIFO Register	256 Data Bytes in 64 DWords to/from Flash																															
0x03C : Flash Clock Divider Register	Multiples of 2 divide 38,4 MHz																															

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x040: FIFO Interface CTRL Dev0	Reserved																											incl. EOF	incl. Timestamp	incl. Packet cnt.	Reserved	incl. SOF
0x042: FIFO Interface CTRL Dev2	Reserved																											incl. EOF	incl. Timestamp	incl. Packet cnt.	Reserved	incl. SOF
0x044: (W1Set) FIFO Interface CTRL Dev0	Reserved																											incl. EOF	incl. Timestamp	incl. Packet cnt.	Reserved	incl. SOF
0x046: (W1Set) FIFO Interface CTRL Dev2	Reserved																											incl. EOF	incl. Timestamp	incl. Packet cnt.	Reserved	incl. SOF
0x048: (W1Clear) FIFO Interface CTRL Dev0	Reserved																											incl. EOF	incl. Timestamp	incl. Packet cnt.	Reserved	incl. SOF
0x04A: (W1Clear) FIFO Interface CTRL Dev2	Reserved																											incl. EOF	incl. Timestamp	incl. Packet cnt.	Reserved	incl. SOF

0x04C: (RO)Enabled Annalog Channels Dev0	Reserved				IF	HS2	HS1
0x04E: (RO)Enabled Annalog Channels Dev2	Reserved				IF	HS2	HS1
0x050: (RO)Enabled Digital Channels Dev0	Reserved				IF	HS2	HS1
0x052: (RO)Enabled Digital Channels Dev2	Reserved				IF	HS2	HS1
0x054: Channel config. HS1 Dev0	Reserved	Channeloffset	Reserved	# of Channels			
0x056: Channel config. HS1 Dev2	Reserved	Channeloffset	Reserved	# of Channels			
0x058: Channel config. HS2 Dev0	Reserved	Channeloffset	Reserved	# of Channels			
0x05A: Channel config. HS2 Dev2	Reserved	Channeloffset	Reserved	# of Channels			
0x05C: Channel config. DSP Dev0	Reserved	Channeloffset	Reserved	# of Channels			
0x05E: Channel config. DSP Dev2	Reserved	Channeloffset	Reserved	# of Channels			
0x060: Channel config. IF Dev0	Reserved	Channeloffset	Reserved	# of Channels			
0x062: Channel config. IF Dev2	Reserved	Channeloffset	Reserved	# of Channels			

0x064 - 0x6F: Reserved	Reserved																		
0x06c: digital configuration Dev0	Reserved	HS2 Sideband	HS2 Trigger	Reserved	HS1 Sideband	HS1 Trigger	Reserved	AUX	FB up	FB low	Reg up	Reg low	DO up	DO low	DI up	DI low	MUX up	MUX low	
0x06e: digital configuration Dev1	Reserved	HS2 Sideband	HS2 Trigger	Reserved	HS1 Sideband	HS1 Trigger	Reserved	AUX	FB up	FB low	Reg up	Reg low	DO up	DO low	DI up	DI low	MUX low	MUX up	
0x070: Digital Data 1 low half Enable Dev0	Reserved					HS2 segm. ID X2	HS2 segm. ID X1	Reserved									HS1 segm. ID X2	HS1 segm. ID X1	
0x072: Digital Data 1 low half Enable Dev2	Reserved					HS2 segm. ID X2	HS2 segm. ID X1	Reserved									HS1 segm. ID X2	HS1 segm. ID X1	
0x074: Digital Data 1 up half Enable Dev0	Reserved					HS2 elec. ID X2	HS2 elec. ID X1	Reserved									HS1 elec. ID X2	HS1 elec. ID X1	
0x076: Digital Data 1 up half Enable Dev2	Reserved					HS2 elec. ID X2	HS2 elec. ID X1	Reserved									HS1 elec. ID X2	HS1 elec. ID X1	
0x080: Digital Data 2 low half Enable Dev0	Reserved	HS2 DAC Data		HS2 Sideband	HS2 Trigger Status	Reserved	HS1 DAC Data			HS1 Sideband	HS1 Trigger Status	Reserved	Aux In	Feedback	Register	Digital Out	Digital In	Mux Data	
0x082: Digital Data 2 low half Enable Dev2	Reserved	HS2 DAC Data		HS2 Sideband	HS2 Trigger Status	Reserved	HS1 DAC Data			HS1 Sideband	HS1 Trigger Status	Reserved	Aux In	Feedback	Register	Digital Out	Digital In	Mux Data	
0x084: Digital Data 2 up half Enable Dev0	Reserved	HS2 DAC Data		HS2 Sideband	HS2 Trigger Status	Reserved	HS1 DAC Data			HS1 Sideband	HS1 Trigger Status	Reserved	Aux In	Feedback	Register	Digital Out	Digital In	Mux Data	
0x086: Digital Data 2 up half Enable Dev2	Reserved	HS2 DAC Data		HS2 Sideband	HS2 Trigger Status	Reserved	HS1 DAC Data			HS1 Sideband	HS1 Trigger Status	Reserved	Aux In	Feedback	Register	Digital Out	Digital In	Mux Data	
0x088: FIFO/Channel2 Mode CTRL Dev0	Reserved							16bit, 24bit, 32 bit mode CTRL*			Reserved				Cy. distinction	Reserved	Triggered Mode	Dev0 Start	
0x08A: FIFO/Channel Mode CTRL Dev2	Reserved							16bit, 24bit, 32 bit mode CTRL*			Reserved				Cy. distinction	Reserved	Triggered Mode	Dev2 Start	
0x08C: Sampling Freq. Dev0	Reserved																Sampling Rate		
0x08E: Sampling Freq. Dev2	Reserved																Sampling Rate		

0x090: Gate Mask Dev0	Gate Mask of Trigged Mode	
0x092: Gate Mask Dev2	Gate Mask of Trigged Mode	
0x094: Compare Value of Gate Mask Dev0	Compare this Value against the Multiplexed Data configured in Register 0x900	
0x096: Compare Value of Gate Mask Dev2	Compare this Value against the Multiplexed Data configured in Register 0x900	
0x098: Amount of sweeps in Trigged Mode Dev0	Count Start Value	
0x09A: Amount of sweeps in Trigged Mode Dev2	Count Start Value	
0x09C: Current Count Value Dev0	Current remaining sweeps until Stop	
0x0B0: Endpoint FIFO Reset	Reserved	0x2: Reset Fifo EP2, 0x6: Reset Fifo EP 6
0x0F0: I2C Blocking for other Cypress	Reserved	Block other Cypress

*Decoding Table:

```
            bit 15  14  13  12  11  10  09  08     Description:
16 bit mode      0   0   0   0  S/U  0   0   0  Bits 23 & 14 downto 0
16 bit mode      0   0   0   1  S/U  0   0   0  Bits 23 & 15 downto 1
16 bit mode      0   0   1   0  S/U  0   0   0  Bits 23 & 16 downto 2
16 bit mode      0   0   1   1  S/U  0   0   0  Bits 23 & 17 downto 3
16 bit mode      0   1   0   0  S/U  0   0   0  Bits 23 & 18 downto 4
16 bit mode      0   1   0   1  S/U  0   0   0  Bits 23 & 19 downto 5
16 bit mode      0   1   1   0  S/U  0   0   0  Bits 23 & 20 downto 6
16 bit mode      0   1   1   1  S/U  0   0   0  Bits 23 & 21 downto 7
16 bit mode      1   0   0   0  S/U  0   0   0  Bits 23      downto 8
reserved mode    x   x   x   x  S/U  0   0   1
24 bit real mode x   x   x   x  S/U  0   1   0  all 24 bit in a chain
32 bit mode      x   x   x   x  S/U  0   1   1  all data 32 bit aligned
                                S/U : Signed or unsigned mode - 0 = unsigend, 1 = signed
```

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x100: ADC Control Register	Reserved																															Start Stop
0x104: Enable ADCs	Reserved																															ADC 1
0x108 - 0x1FC:	Reserved																															

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x200: Trigger CTRL Register 1	Reserved				Trigger status feedback source				Reserved				send Rise/Fall seperate per Trigger												Reserved							Enable Trigger
0x214: Trigger Event Status Register	Reserved																				Trigger Event											
0x218: Trigger Segment ID Register	Reserved																								Segment ID 1, to HS1 STG 1 Trigger 1							
0x21C: Trigger Segment ID Register	Reserved																								Segment ID 2, to HS1 STG 1 Trigger 2							
0x220: Trigger Segment ID Register	Reserved																								Segment ID 3, to HS1 STG 1 Trigger 3							
0x224: Trigger Segment ID Register	Reserved																								Segment ID 4, to HS1 STG 2 Trigger 1							
0x228: Trigger Segment ID Register	Reserved																								Segment ID 5, to HS1 STG 2 Trigger 2							
0x22C: Trigger Segment ID Register	Reserved																								Segment ID 6, to HS1 STG 2 Trigger 3							
0x230: Trigger Segment ID Register	Reserved																								Segment ID 7, to HS2 STG 1 Trigger 1							
0x234: Trigger Segment ID Register	Reserved																								Segment ID 8, to HS2 STG 1 Trigger 2							

0x238: Trigger Segment ID Register	Reserved			Segment ID 9, to HS2 STG 1 Trigger 3
0x23C: Trigger Segment ID Register	Reserved			Segment ID 10, to HS2 STG 2 Trigger 1
0x240: Trigger Segment ID Register	Reserved			Segment ID 11, to HS2 STG 2 Trigger 2
0x244: Trigger Segment ID Register	Reserved			Segment ID 12, to HS2 STG 2 Trigger 3
0x260: Trigger status feedback HS1 Register (RO/RW)	Reserved	Armed 6 ... 1	Reserved	Running 6 ... 1
0x264: Trigger status feedback HS2 Register (RO/RW)	Reserved	Armed 6 ... 1	Reserved	Running 6 ... 1
0x280 - 0x294: External trigger source	Data value from 0 to 99 for Trigger 1 to 6 (Headstage 1)***			
0x298 - 0x2AC: External trigger source	Data value from 0 to 99 for Trigger 7 to 12 (Headstage 2)***			

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x300: Digital Out Register	Data (RO)																															
0x304: Digital In Register	Data (RO)																															
0x308: Direction Register	'1': Input, '0' Output																															
0x30C: Interrupt Enable	'1': Interrupt enabled, only for Inputs, '0' Interrupt disabled																															
0x310: Aux Data Out Register	Reserved																														Aux Data	
0x314: Aux Data In Register	Reserved																														Aux Data (RO)	
0x318: Aux Data Dir Register	Reserved																														'0': Input, '1' Output	

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x400: DSP Indata CTRL Register	Reserved																Filtered HS2 Data en	Filtered HS1 Data en	Timestamp Data en	Digital Data en	IF Data en	HS2 Data en	HS1 Data en	Int en	Reserved							Clr Fifo Flags	Reset Fifo
0x404: DSP Outdata CTRL Register	Reserved																							Int en	Reserved							Clr Fifo Flags	Reset Fifo
0x408: DSP In Fifo Status Flags (RO)	Reserved																								overflow occured	overflow	full	progr full	prog empty	empty	underflow	underflow occured
0x40C: DSP Out Fifo Status Flags (RO)	Reserved																								overflow occured	overflow	full	progr full	prog empty	empty	underflow	underflow occured
0x410: DSP Indata Threshold Register	Reserved														Fifo full TH																	
	Reserved																Reserved														Fifo empty TH		

0x414: DSP Outdata Threshold Register	Reserved	Fifo full TH	Reserved	Fifo empty TH					
0x420: DSP Boot conf. Register	Reserved	Bootmode	Reserved	Reset	POR	Reserved	I2C Bus enable		
0x424: DSP Mailbox CTRL Register	Reserved	Mailbox to DSP int enable (GP 6)	Reserved						
0x428: DSP Mailbox Last used Addr. Register	Reserved	Last used Address for write to Mailbox							
0x430: DSP Streamdata CTRL Register	Reserved	Keep last Streamdata	Enable Streaming	Reserved	Int en	Reserved	Clr Fifo Flags	Reset Fifo	
0x434: DSP Stream Fifo Status Flags (RO)	Reserved	overflow occured	overflow	full	progr full	prog empty	empty	underflow	underflow occured
Filter coefficents									
0x600: Filter 1 coefficent b[0]	Filter coefficent b[0] as Q1.30 value								
0x608: Filter 1 coefficent b[1]	Filter coefficent b[1] as Q1.30 value								
0x60C: Filter 1 coefficent a[1]	Filter coefficent a[1] as Q1.30 value								
0x610: Filter 1 coefficent b[2]	Filter coefficent b[2] as Q1.30 value								
0x614: Filter 1 coefficent a[2]	Filter coefficent a[2] as Q1.30 value								

0x61C: Filter 1 CTRL	Reserved	Filter Enable HS 2	Filter Enable HS 1
0x620: Filter 2 coefficent b[0]	Filter coefficent b[0] as Q1.30 value		
0x628: Filter 2 coefficent b[1]	Filter coefficent b[1] as Q1.30 value		
0x62C: Filter 2 coefficent a[1]	Filter coefficent a[1] as Q1.30 value		
0x630: Filter 2 coefficent b[2]	Filter coefficent b[2] as Q1.30 value		
0x634: Filter 2 coefficent a[2]	Filter coefficent a[2] as Q1.30 value		
0x63C: Filter 2 CTRL	Reserved	Filter Enable HS 2	Filter Enable HS 1
0x640: Filter 3 coefficent b[0]	Filter coefficent b[0] as Q1.30 value		
0x648: Filter 3 coefficent b[1]	Filter coefficent b[1] as Q1.30 value		
0x64C: Filter 3 coefficent a[1]	Filter coefficent a[1] as Q1.30 value		
0x650: Filter 3 coefficent b[2]	Filter coefficent b[2] as Q1.30 value		
0x654: Filter 3 coefficent a[2]	Filter coefficent a[2] as Q1.30 value		
0x65C: Filter 3 CTRL	Reserved	Filter Enable HS 2	Filter Enable HS 1

0x660: Filter 4 coefficent b[0]	Filter coefficent b[0] as Q1.30 value		
0x668: Filter 4 coefficent b[1]	Filter coefficent b[1] as Q1.30 value		
0x66C: Filter 4 coefficent a[1]	Filter coefficent a[1] as Q1.30 value		
0x670: Filter 4 coefficent b[2]	Filter coefficent b[2] as Q1.30 value		
0x674: Filter 4 coefficent a[2]	Filter coefficent a[2] as Q1.30 value		
0x67C: Filter 4 CTRL	Reserved	Filter Enable HS 2	Filter Enable HS 1

Configurable DSP in Data overview:

```
HS1           122 Datawords (Header, 120 Data, Counter)

HS2           122 Datawords (Header, 120 Data, Counter)

IF              9 Datawords (Header, 8 Data)

Filtered HS1  122 Datawords (Header, 120 Data, Counter)

Filtered HS2  122 Datawords (Header, 120 Data, Counter)

Digital        32 Datawords (Header, 31x Digital Data)

   - Header                              (32 bit)

   - Digital Mux Data for USB-A          (32 bit)

   - Digital Mux Data for USB-B          (32 bit)

   - Digital In Data                     (32 bit)

   - Digital Out Data                    (32 bit)

   - Digital Register Data (0x800)       (32 bit)

   - Feedback                            (32 bit)

   - Aux In:

       Bit 0: Input Level of Aux 1

       Bit 1: Input Level of Aux 2

   - Trigger Status from HS1:

       Bits  0 - 11 Trigger Statemachine status (2 bit per trigger: 00: not Armed, 01: Armed, 10: Triggerd (running), 11: Reserved)

       Bits 16 - 21 Trigger Event (1: Trigger event occured)

       Bits 24 - 29 Trigger Event Type (1: begin of Trigger event, 0: end of Trigger event)

   - SBS1 from HS1:

       Bits  0 - 15: Headstage 1: Sideband 1 from STG 1

       Bits 16 - 31: Headstage 1: Sideband 1 from STG 2

   - SBS2 from HS1:
```

```
    Bits  0 - 15: Headstage 1: Sideband 2 from STG 1

    Bits 16 - 31: Headstage 1: Sideband 2 from STG 2

- SBS3 from HS1:

    Bits  0 - 15: Headstage 1: Sideband 3 from STG 1

    Bits 16 - 31: Headstage 1: Sideband 3 from STG 2

- DAC A+B from HS1 of STG1:

    Bits  0 - 15: Headstage 1: DAC A from STG 1

    Bits 16 - 31: Headstage 1: DAC B from STG 1

- DAC C+D from HS1 of STG1:

    Bits  0 - 15: Headstage 1: DAC C from STG 1

    Bits 16 - 31: Headstage 1: DAC D from STG 1

- DAC E+F from HS1 of STG1:

    Bits  0 - 15: Headstage 1: DAC E from STG 1

    Bits 16 - 31: Headstage 1: DAC F from STG 1

- DAC A+B from HS1 of STG2:

    Bits  0 - 15: Headstage 1: DAC A from STG 2

    Bits 16 - 31: Headstage 1: DAC B from STG 2

- DAC C+D from HS1 of STG2:

    Bits  0 - 15: Headstage 1: DAC C from STG 2

    Bits 16 - 31: Headstage 1: DAC D from STG 2

- DAC E+F from HS1 of STG2:

    Bits  0 - 15: Headstage 1: DAC E from STG 2

    Bits 16 - 31: Headstage 1: DAC F from STG 2

- Seg. & Elec. ID from HS1 of STG1:

    Bits  0 -  7: current Segment ID

    Bits 16 - 23: Electrode Config ID

- Seg. & Elec. ID from HS1 of STG2:

    Bits  0 -  7: current Segment ID

    Bits 16 - 23: Electrode Config ID

- Trigger Status from HS2:

    Bits  0 - 11 Trigger Statemachine status (2 bit per trigger: 00: not Armed, 01: Armed, 10: Triggerd (running), 11: Reserved)

    Bits 16 - 21 Trigger Event (1: Trigger event occured)

    Bits 24 - 29 Trigger Event Type (1: begin of Trigger event, 0: end of Trigger event)

- SBS1 from HS2:

    Bits  0 - 15: Headstage 2: Sideband 1 from STG 1

    Bits 16 - 31: Headstage 2: Sideband 1 from STG 2

- SBS2 from HS2:

    Bits  0 - 15: Headstage 2: Sideband 2 from STG 1

    Bits 16 - 31: Headstage 2: Sideband 2 from STG 2

- SBS3 from HS2:

    Bits  0 - 15: Headstage 2: Sideband 3 from STG 1

    Bits 16 - 31: Headstage 2: Sideband 3 from STG 2

- DAC A+B from HS2 of STG1:

    Bits  0 - 15: Headstage 2: DAC A from STG 1

    Bits 16 - 31: Headstage 2: DAC B from STG 1

- DAC C+D from HS2 of STG1:

    Bits  0 - 15: Headstage 2: DAC C from STG 1
```

```
      Bits 16 - 31: Headstage 2: DAC D from STG 1

   - DAC E+F from HS2 of STG1:

      Bits  0 - 15: Headstage 2: DAC E from STG 1

      Bits 16 - 31: Headstage 2: DAC F from STG 1

   - DAC A+B from HS2 of STG2:

      Bits  0 - 15: Headstage 2: DAC A from STG 2

      Bits 16 - 31: Headstage 2: DAC B from STG 2

   - DAC C+D from HS2 of STG2:

      Bits  0 - 15: Headstage 2: DAC C from STG 2

      Bits 16 - 31: Headstage 2: DAC D from STG 2

   - DAC E+F from HS2 of STG2:

      Bits  0 - 15: Headstage 2: DAC E from STG 2

      Bits 16 - 31: Headstage 2: DAC F from STG 2

   - Seg. & Elec. ID from HS2 of STG1:

      Bits  0 -  7: current Segment ID

      Bits 16 - 23: Electrode Config ID

   - Seg. & Elec. ID from HS2 of STG2:

      Bits  0 -  7: current Segment ID

      Bits 16 - 23: Electrode Config ID
Timestamp       3 Datawords (Header, 2x Counter == 64 bit Counter)
```

```
The header has the format
 Bit 31       :  1, when data from a headstage is enabled and the HS is not connected, otherwise 0
 Bit 24 to 30 :  Data Source enumeration (see below)
 Bit 9 to 23  :  Reserved (Always Zero)
 Bit 0 to 7   :  The number of datapoints + counter values following this header, 0x79 for Headstage data (120 + 1)
```

```
Data Source enumeration:
 1: Headstage 1
 2: Headstage 2
 3: Analog Data from Interfaceboard
 4: Headstage 1 filtered Data
 5: Headstage 2 filtered Data
 6: Digital Data
 7: Timestamp counter
```

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x500: Right Audio Channel	Reserved												Source (DSP, IF, HS2, HS1)				Reserved								Channel							
0x504: Left Audio Channel	Reserved												Source (DSP, IF, HS2, HS1)				Reserved								Channel							
0x510: Right Audio Attenuation	Reserved																										Attenuation					
0x514: Left Audio Attenuation	Reserved																										Attenuation					

```
Source decoding:
 0: No source
 1: HS1
 2: HS2
 3: IF
 4: DSP(bits 23 - 0)
```

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x700: Digital Pulse Register	bits can only be set here, and will be cleared after the time defined in reg 0x704 to 0x710																															
0x704: Digital Pulse Duration 1	Duration of Digital Data in multiple of 20 us for Pulse Register bits 0 to 7																															
0x708: Digital Pulse Duration 2	Duration of Digital Data in multiple of 20 us for Pulse Register bits 8 to 15																															
0x70c: Digital Pulse Duration 3	Duration of Digital Data in multiple of 20 us for Pulse Register bits 16 to 23																															
0x710: Digital Pulse Duration 4	Duration of Digital Data in multiple of 20 us for Pulse Register bits 24 to 31																															
0x780: Feedback Data	Data																															
0x800: Data to Data Stream	'Digital Data-Stream' face in data																															
0x804: Dig. Data Delay	Reserved																											Number of sweeps to delay the digital data				
0x808: delay configuration	Switch of datawords if delayed or direct*																															
0x840 - 0x87C: Select mux for Dig out	Data value from 0 to 315 **** for Digital out: 2 to 32(even bits)															Data value from 0 to 315 **** for Digital out: 1 to 31(odd bits)													
0x880 - 0x8BC: Select mux for Dig Data to USB-A	Data value from 0 to 315 **** for Digital Data stream to Host: 2 to 32(even bits)															Data value from 0 to 315 **** for Digital Data stream to Host: 1 to 31(odd bits)													
0x8C0 - 0x8FC: Select mux for Dig Data to USB-B	Data value from 0 to 315 **** for Digital Data stream to Host: 2 to 32(even bits)															Data value from 0 to 315 **** for Digital Data stream to Host: 1 to 31(odd bits)													
0x900 - 0x93C: Select mux for Trigger of Gated Mode to USB	Data value from 0 to 315 **** for Trigger of Gate to Host: 2 to 32(even bits)															Data value from 0 to 315 **** for Trigger of Gate to Host: 1 to 31(odd bits)													

*Decoding Table:

```
   Digital In Data                    (bit 1)
   Digital Out Data                   (bit 2)
   Feedback                           (bit 4)
   Aux In                             (bit 5)
```

```
Trigger Statemashine Status from HS1 (bit 8)
SBS from HS1 Channel 1 of X1 and X2  (bit 9)
SBS from HS1 Channel 2 of X1 and X2  (bit 10)
SBS from HS1 Channel 3 of X1 and X2  (bit 11)
DAC A+B from HS1 of X1               (bit 12)
DAC C+D from HS1 of X1               (bit 13)
DAC E+F from HS1 of X1               (bit 14)
DAC A+B from HS1 of X2               (bit 15)
DAC C+D from HS1 of X2               (bit 16)
DAC E+F from HS1 of X2               (bit 17)
Trigger Statemashine Status from HS2 (bit 20)
SBS from HS2 Channel 1 of X1 and X2  (bit 21)
SBS from HS2 Channel 2 of X1 and X2  (bit 22)
SBS from HS2 Channel 3 of X1 and X2  (bit 23)
DAC A+B from HS2 of X1               (bit 24)
DAC C+D from HS2 of X1               (bit 25)
DAC E+F from HS2 of X1               (bit 26)
DAC A+B from HS2 of X2               (bit 27)
DAC C+D from HS2 of X2               (bit 28)
DAC E+F from HS2 of X2               (bit 29)
```

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA00 - 0xA2C: Config ID boundarys	set first config ID to end	Reserved							Config Start ID								Reserved								Config End ID							
0xA40: Config ID Manual/current for SBS 1	Reserved																								Electrode Config ID 0x00 to 0xFF							
0xA4C: Config ID Manual/current for SBS 4	Reserved																								Electrode Config ID 0x00 to 0xFF							
0xA58: Config ID Manual/current for SBS 7	Reserved																								Electrode Config ID 0x00 to 0xFF							
0xA64: Config ID Manual/current for SBS 10	Reserved																								Electrode Config ID 0x00 to 0xFF							
0xA80: Config ID trigger source	Reserved																														Data value from 0 to 100 for Config ID 1***	

0xA8C: Config ID trigger source	Reserved	Data value from 0 to 100 for Config ID 4***
0xA98: Config ID trigger source	Reserved	Data value from 0 to 100 for Config ID 7***
0xAA4: Config ID trigger source	Reserved	Data value from 0 to 100 for Config ID 10***
0xAC0: (Re-)Start List mode increment	Reserved	1 bit per Sideband
0xAC4: Stop List mode increment	Reserved	1 bit per Sideband

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xE00: DMA CTRL Register	Reserved																															DMA Enable
0xE04 : EEPRom Address Ptr. Register	Reserved																															
0xE08 : EEPRom Data Ptr. Register	Reserved																Reserved															

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xF00: (WO)EEPRom Instruction Code Register	Reserved																							Fifo Reset	Instruction Code							
0xF00: (RO)EEPRom Status Register	Reserved																					FIFO_empty	FIFO_full	Statemachine busy	Flash Status Register							
0xF04 : EEPRom Memory Address Register	Reserved								Flash Address																							
0xF08 : EEPRom Data FIFO Register	256 Data Bytes in 64 DWords to/from Flash																															

0xF0C : EEPROM HW configuration Register	Address length (1,2,3 Bytes)	Clock Divider Register (Multiples of 2 divide 38,4 MHz)
0xF10 : EEPROM Offset Register	Offset for reads and writes to the EEprom	
0xF14 : EEPROM Size Register	Size of the EEprom Block available for this Image	

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xF80: FPGA GENERAL2 Register	Reserved																GENERAL2 data															
0xF84: FPGA GENERAL5 Register	Reserved																GENERAL5 data															
0xF90: direct connection to FPGA SPI Flash	Reserved																														Cypress number (0 = none)	
0xF94: Config EEprom Base	Reserved																base address of config eeprom															
0xF98: FX3-USB Bootstrap Firmware	reserved																Firmware Version of FX3 Bootstrap code this USB Port															
0xF9C: FX3-USB Bootstrap Firmware other port	reserved																Firmware Version of FX3 Bootstrap code on other USB Port															
0xFB0: Message to other Cypress	A write will cause an interrupt on the other Cypress, with a read you can check is it is read already																															
0xFB4: Read Only: Message from other Cypress	Each bit which is set will be reset on read																															

Register	from	to	in Revision	USB Rev
Electrode config/segment ID register	0x0208 - 0x0234	0x0218 - 0x0244	0.05	0.27
Event Status Register	0x0238	0x0214	0.05	0.27
ADC Filter	0x0120 - 0x0138	0x0600 - 0x0674	0.05	
FIFO/Channel Mode CTRL DevX	0x0068 - 0x006B	0x006C - 0x006F	0.05	
Sampling Freq. DevX	0x006C - 0x006F	0x0070 - 0x0073	0.05	
DIGITAL_CHANNELS_L	0x0064 - 0x0067	0x0080 - 0x0083	0.07	0.42
DIGITAL_CHANNELS_H	0x0068 - 0x006B	0x0084 - 0x0087	0.07	0.42
FIFO/Channel Mode CTRL DevX	0x006C - 0x006F	0x0088 - 0x008B	0.07	0.42
Sampling Freq. DevX	0x0070 - 0x0073	0x008C - 0x008F	0.07	0.42
Select mux for Dig Data	0x0340 - 0x03FC	0x0840 - 0x08FC	0.09	0.51

***Decoding Table:

```
Value     Source
0         : '0' (default for all bits)
32  -   1 : Digital In bit 31 downto 0
64  -  33 : Feedback bit 31 downto 0
66  -  65 : Aux In bit 1 downto 0
98  -  67 : Digital Pulse Register bit 31 downto 0
99        : '1'
100       : SBS bit 8 of each Trigger is all STGs (only for Config ID)
101       : DACQ Cypress 1 Virtual Device 1 is started
102       : DACQ Cypress 1 Virtual Device 2 is started
103       : DACQ Cypress 2 Virtual Device 1 is started
104       : DACQ Cypress 2 Virtual Device 2 is started
```

****Decoding Table:

```
Value       Source
31  -   0 : Digital In bit 31 downto 0
63  -  32 : Digital Pulse bit 31 downto 0
95  -  64 : Feedback bit 31 downto 0
97  -  96 : Aux In bit 1 downto 0
99  -  98 : "10" fix Values
101 - 100 : Trigger Status HS1 STG X1 Trigger 1
103 - 102 : Trigger Status HS1 STG X1 Trigger 2
105 - 104 : Trigger Status HS1 STG X1 Trigger 3
107 - 106 : Trigger Status HS1 STG X2 Trigger 1
109 - 108 : Trigger Status HS1 STG X2 Trigger 2
111 - 110 : Trigger Status HS1 STG X2 Trigger 3
127 - 112 : Sideband signals HS1 STG X1 Trigger 1
143 - 128 : Sideband signals HS1 STG X1 Trigger 2
159 - 144 : Sideband signals HS1 STG X1 Trigger 3
175 - 160 : Sideband signals HS1 STG X2 Trigger 1
191 - 176 : Sideband signals HS1 STG X2 Trigger 2
207 - 192 : Sideband signals HS1 STG X2 Trigger 3
```

```
209 – 208 : Trigger Status HS2 STG X1 Trigger 1
211 – 210 : Trigger Status HS2 STG X1 Trigger 2
213 – 212 : Trigger Status HS2 STG X1 Trigger 3
215 – 214 : Trigger Status HS2 STG X2 Trigger 1
217 – 216 : Trigger Status HS2 STG X2 Trigger 2
219 – 218 : Trigger Status HS2 STG X2 Trigger 3
235 – 220 : Sideband signals HS2 STG X1 Trigger 1
251 – 236 : Sideband signals HS2 STG X1 Trigger 2
267 – 252 : Sideband signals HS2 STG X1 Trigger 3
283 – 268 : Sideband signals HS2 STG X2 Trigger 1
299 – 284 : Sideband signals HS2 STG X2 Trigger 2
315 – 300 : Sideband signals HS2 STG X2 Trigger 3
```

Mailbox Register Address Map (Address bits 11-0) Base Address: 0x1000:

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000 - 0xFFC: Mailbox Registers	write will generate an Interrupt on GP 6 Line																															

RAM Register Address Map (Address bits 11-0) Base Address: 0x2000:

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000 - 0xFFC: RAM Registers																																

Headstage Board Address Map (Address bits 11-0) Base Address: HS1: 0x8000 HS2: 0xC000:

Access time is 2,2 us for writes and 2,4 us for reads.

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000: Device ID Register	0x00000002 = MEA 21 Headstage FPGA																															
0x004: HW/FPGA Version Register	HW/Board Version																FPGA Version: 0x0101 = Initial Version															
0x008 : Configuration Register	Reserved																							STG Presend	Reserved		Additional Boards attached, What kind of board?, ... *					

0x00c : Channel Limit	Reserved	Number of supported analog channels

*Decoding Table:

```
0: 2 x 60 MEA
1: 1 x 60 MEA
2: 1 x 120 MEA
3: special
```

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x010: GTP CTRL Register	Reserved																															
0x014: (RO)GTP Error counter Register	Reserved																															
0x014: (WO)GTP Error counter Reset	Reserved																															

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x020: FPGA Reset	Reserved																													STG X2 Reset	STG X1 Reset	FPGA Reset
0x024: Power Enable	Reserved																															Trafo

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x030: (WO)Flash Instruction Code Register	Reserved																						Stop instruction	Fifo Reset	Instruction Code							
0x030: (RO)Flash Status Register	Reserved																					FIFO_empty	FIFO_full	Statemachine busy	Flash Status Register							
0x034 : Flash Memory Address Register	Reserved								Flash Address																							
0x038 : Flash Data FIFO Register	256 Data Bytes in 64 DWords to/from Flash																															

0x03C : Flash HW configuration register	Address length (1,2,3 Bytes)	Clock Divider Register (Multiples of 2 divide 38,4 MHz)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x100: ADC Control Register	Reset done?	Artif. Data	Force data keep	Reserved		Enable Data keep on Measurement channels, post filter	Enable Data keep on Measurement channels, mid filter	Enable Data keep on Measurement channels, pre filter	Reserved					Enable Data keep on Stimulation channels, post filter	Enable Data keep on Stimulation channels, mid filter	Enable Data keep on Stimulation channels, pre filter	Offset correction Enable	Learn DC offsets	Reserved											Reserved (do not use, see Changed Register Information)		Start Stop
0x104: Enable ADCs	Reserved																	ADC 15 - 1														
0x108: Enable input switches	Reserved		Electrode 30 - 1																													
0x10C: Enable input switches	Reserved		Electrode 60 - 31																													
0x110: Enable input switches	Reserved		Electrode 90 - 61																													
0x114: Enable input switches	Reserved		Electrode 119 - 91																													
0x118: ADC settings	Reserved														ADC Power MODE		Reserved													25.6 CLK Enable		SCLK Enable
0x11C: Filter and ADC Gain	Gain of Filter coefficient + ADC Gain for GUI SW configuration, default 0x8000 07D0; 2000 meaning 2.0																															
0x140: Open Amplifier Protection Switch while Blanking	Reserved		Electrode 30 - 1																													
0x144: Open Amplifier Protection Switch while Blanking	Reserved		Electrode 60 - 31																													
0x148: Open Amplifier Protection Switch while Blanking	Reserved		Electrode 90 - 61																													

0x14C: Open Amplifier Protection Switch while Blanking	Reserved	Electrode 119 - 91		
0x150: Enable Blanking	Reserved	Electrode 30 - 1		
0x154: Enable Blanking	Reserved	Electrode 60 - 31		
0x158: Enable Blanking	Reserved	Electrode 90 - 61		
0x15C: Enable Blanking	Reserved	Electrode 119 - 91		
0x160: Stimulus Data keep pre filter	Reserved	onset delay	Reserved	offset delay
0x164: Stimulus Data keep mid filter	Reserved	onset delay	Reserved	offset delay
0x168: Stimulus Data keep post filter	Reserved	onset delay	Reserved	offset delay
0x170: Non-Stimulus Data keep pre filter	Reserved	onset delay	Reserved	offset delay
0x174: Non-Stimulus Data keep mid filter	Reserved	onset delay	Reserved	offset delay
0x178: Non-Stimulus Data keep post filter	Reserved	onset delay	Reserved	offset delay
0x180: Channel Index	Reserved			Index used for Reg.: 0x184 - 0x188
0x184: Current ADC Value	Reserved	Reflects the ADC Value of the channel selected in Reg. 0x180		
0x188: Current DC Channel offset	Reserved	Reflects the DC offset correction Value of the channel selected in Reg. 0x180		

0x1F0: Stimulation Channel copy	Reserved	Electrode 30 - 1 Reflects the STG Register 0x158 and 0x15C for monitoring if information is on HS				
0x1F4: Stimulation Channel copy	Reserved	Electrode 60 - 31				
0x1F8: Stimulation Channel copy	Reserved	Electrode 90 - 61				
0x1FC: Stimulation Channel copy	Reserved	Electrode 119 - 91				
0x600: Filter 1 coefficent b[0]	Filter coefficent b[0] as Q1.16 value		"00000000000000" to extend coefficient to Q1.30			
0x604: Reserved (DC)	Reserved					
0x608: Filter 1 coefficent b[1]	Filter coefficent b[1] as Q1.16 value		"00000000000000" to extend coefficient to Q1.30			
0x60C: Filter 1 coefficent a[1]	Filter coefficent a[1] as Q1.30 value					
0x610: Filter 1 coefficent b[2]	Filter coefficent b[2] as Q1.16 value		"00000000000000" to extend coefficient to Q1.30			
0x614: Filter 1 coefficent a[2]	Filter coefficent a[2] as Q1.30 value					
0x618: Filter 1 coefficent a[1]ext. and a[2]ext.	Filter Coef. A2 lower bits -31 downto -46		Filter Coef. A1 lower bits -31 downto -46			
0x61C: Filter 1 CTRL	Reserved			Reserved Mode	Filter Blank Enable	Filter Enable
0x620: Filter 2 coefficent b[0]	Filter coefficent b[0] as Q1.16 value		"00000000000000" to extend coefficient to Q1.30			
0x624: Reserved (DC)	Reserved					
0x628: Filter 2 coefficent b[1]	Filter coefficent b[1] as Q1.16 value		"00000000000000" to extend coefficient to Q1.30			

0x62C: Filter 2 coefficent a[1]	Filter coefficent a[1] as Q1.30 value															
0x630: Filter 2 coefficent b[2]	Filter coefficent b[2] as Q1.16 value														"000000000000000" to extend coefficient to Q1.30	
0x634: Filter 2 coefficent a[2]	Filter coefficent a[2] as Q1.30 value															
0x638: Filter 2 coefficent a[1]ext. and a[2]ext.	Filter Coef. A2 lower bits -31 downto -46														Filter Coef. A1 lower bits -31 downto -46	
0x63C: Filter 2 CTRL	Reserved														Reserved Mode	Filter Blank Enable	Filter Enable
0x640: Filter 3 coefficent b[0]	Filter coefficent b[0] as Q1.16 value														"000000000000000" to extend coefficient to Q1.30	
0x644: Reserved (DC)	Reserved															
0x648: Filter 3 coefficent b[1]	Filter coefficent b[1] as Q1.16 value														"000000000000000" to extend coefficient to Q1.30	
0x64C: Filter 3 coefficent a[1]	Filter coefficent a[1] as Q1.30 value															
0x650: Filter 3 coefficent b[2]	Filter coefficent b[2] as Q1.16 value														"000000000000000" to extend coefficient to Q1.30	
0x654: Filter 3 coefficent a[2]	Filter coefficent a[2] as Q1.30 value															
0x658: Filter 3 coefficent a[1]ext. and a[2]ext.	Filter Coef. A2 lower bits -31 downto -46														Filter Coef. A1 lower bits -31 downto -46	
0x65C: Filter 3 CTRL	Reserved														Reserved Mode	Filter Blank Enable	Filter Enable
0x660: Filter 4 coefficent b[0]	Filter coefficent b[0] as Q1.16 value														"000000000000000" to extend coefficient to Q1.30	
0x664: Reserved (DC)	Reserved															
0x668: Filter 4 coefficent b[1]	Filter coefficent b[1] as Q1.16 value														"000000000000000" to extend coefficient to Q1.30	

0x66C: Filter 4 coefficent a[1]	Filter coefficent a[1] as Q1.30 value					
0x670: Filter 4 coefficent b[2]	Filter coefficent b[2] as Q1.16 value			"00000000000000" to extend coefficient to Q1.30		
0x674: Filter 4 coefficent a[2]	Filter coefficent a[2] as Q1.30 value					
0x678: Filter 4 coefficent a[1]ext. and a[2]ext.	Filter Coef. A2 lower bits -31 downto -46		Filter Coef. A1 lower bits -31 downto -46			
0x67C: Filter 4 CTRL	Reserved			Reserved	Mode	Filter Blank Enable	Filter Enable
0x6A0: Filter 2 coefficent b2[0]	Filter coefficent b2[0] as Q1.16 value for one clock after blank			"00000000000000" to extend coefficient to Q1.30		
0x6A8: Filter 2 coefficent b2[1]	Filter coefficent b2[1] as Q1.16 value for one clock after blank			"00000000000000" to extend coefficient to Q1.30		
0x6AC: Filter 2 coefficent a2[1]	Filter coefficent a2[1] as Q1.30 value for one clock after blank					
0x6B0: Filter 2 coefficent b2[2]	Filter coefficent b2[2] as Q1.16 value for one clock after blank			"00000000000000" to extend coefficient to Q1.30		
0x6B4: Filter 2 coefficent a2[2]	Filter coefficent a2[2] as Q1.30 value for one clock after blank					
0x6c0: Filter Info	Corner frequency of Hardware Filter in mHz					
0x6c4: Filter Info	Hardware Filter order	Band: Lowpass	Family: Butterworth/Bessel	Reserved	0 = Hardware Filter	Filter Active
0x6d0: Filter Info	Corner frequency of Highpass Filter in mHz					
0x6d4: Filter Info	Highpass Filter order	Band: Highpass	Family: Butterworth/Bessel	Reserved	1 = Software Filter	Filter Active
0x6e0: Filter Info	Corner frequency of Lowpass Filter in mHz					
0x6e4: Filter Info	Lowpass Filter order	Band: Lowpass	Family: Butterworth/Bessel	Reserved	1 = Software Filter	Filter Active

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x300: (WO)EEPROM Instruction Code Register	Reserved																						Stop instruction	Fifo Reset	Instruction Code								0x300: (WO)EEPRom Instruction Code Register	Reserved	Fifo Reset	Instruction Code
0x300: (RO)EEPRom Status Register	Reserved																					FIFO_empty	FIFO_full	Statemachine busy	Flash Status Register											
0x304 : EEPRom Memory Address Register	Reserved								Flash Address																											
0x308 : EEPRom Data FIFO Register	256 Data Bytes in 64 DWords to/from Flash																																			
0x30C : EEPROM HW configuration register	Address length (1,2,3 Bytes)		Clock Divider Register (Multiples of 2 divide 38,4 MHz)																																	

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xE00: DMA CTRL Register	Reserved																															DMA Enable
0xE04 : EEPRom Address Ptr. Register	Reserved																															
0xE08 : EEPRom Data Ptr. Register	Reserved																Reserved															

Command	Bit 31 to 24	Bit 23 to 0	Description
STP	0xFF	0xXXXXXX	Stop DMA
SDTA	0x01	0x008RegAddr.	Store Next Data to Register Address
DATA	0xData	0xData	Data expected after SDTA command

Register	from	to	in Revision
ADC Filter	0x0120 - 0x0138	0x0600 - 0x0674	0.04
ADC Filter enable	0x100 bits 4:1	0x61C, 0x63C, 0x65C, 0x67C each bit 0	1.15

Stimulus Board Address Map (Address bits 11-0) Base Address: HS1: 0x9000 and 0xA000 HS2: 0xD000 and 0xE000:

Access time is 4,3 us for writes and 6,4 us for reads.

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000: Device ID Register	0x00000003 = MEA 21 Stimulus FPGA																															
0x004: HW/FPGA Version Register	Reserved																HW/Board Version															FPGA Version: 0x0101 = Initial Version										

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x010: SPI CTRL Register	Reserved																															

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x020: FPGA Reset (WO)	Reserved																													DDR Block Reset	DDR clk Reset	FPGA Reset
0x024: FPGA Reset Status (RO)	Reserved																													DDR Block PLL Locked	PLL Locked	System Reset
0x028: System Pll CTRL (WO)	Reserved																														Decrement Phase	Increment Phase

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x030: (WO)Flash Instruction Code Register	Reserved																						Stop instruction	Fifo Reset	Instruction Code								
0x030: (RO)Flash Status Register	Reserved																					FIFO_empty	FIFO_full	Statemachine busy	Flash Status Register							
0x034 : Flash Memory Address Register	Reserved								Flash Address																							
0x038 : Flash Data FIFO Register	256 Data Bytes in 64 DWords to/from Flash																															
0x03C : Flash Clock Divider Register	Multiples of 2 divide 38,4 MHz																															

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x100: DAC Control Register	Reserved																		0: Current 1: Voltage DAC: F	0: Current 1: Voltage DAC: E	0: Current 1: Voltage DAC: D	0: Current 1: Voltage DAC: C	0: Current 1: Voltage DAC: B	0: Current 1: Voltage DAC: A	Reserved			DAC Test	Reserved		DAC data source (0 intern, 1 DSP stream)	Start Stop
0x104: Trigger select* for Stim MEM FSM	Reserved						Mem 7	Reserved						Mem 5	Reserved								Mem 3		Reserved						Mem 1	
0x108: Trigger select* for Stim MEM FSM	Reserved						Mem 8	Reserved						Mem 6	Reserved								Mem 4		Reserved						Mem 2	
0x10C: Stop Trigger Register	Reserved																														Stop Trigger 3-2-1	

0x110: Trigger conf. Register**	Reserved			Trigger 1
0x114: Trigger conf. Register**	Reserved			Trigger 2
0x118: Trigger conf. Register**	Reserved			Trigger 3
0x11C: Reserved	Reserved			
0x120: Electrode Mode	Reserved	Electrode 15 - 1 (Manual mode: 11 / Automatic mode: 00 , other bit combinations are reserved)		
0x124: Electrode Mode	Reserved	Electrode 30 - 16		
0x128: Electrode Mode	Reserved	Electrode 45 - 31		
0x12C: Electrode Mode	Reserved	Electrode 60 - 46		
0x130: Reserved	Reserved			
0x134: Reserved	Reserved			
0x138: Reserved	Reserved			
0x13C: Reserved	Reserved			
0x140: Reserved	Reserved			
0x144: Reserved	Reserved			
0x148: Reserved	Reserved			
0x14C: Reserved	Reserved			
0x150: Electr. conf. ID CTRL f. Auto	Reserved	Electrode configuration source 0:int/1:ext	Reserved	Electrode Config ID

0x154: DAC Sideband Select***	Reserved	DAC F	Reserved	DAC D	Reserved	DAC B	Reserved	DAC E	Reserved	DAC C	Reserved	DAC A	
0x158: Stimulation Enable	Reserved	Electr. 30 - 1 (Segment ID depending on bits 7:0 of reg. 0x150)											
0x15C: Stimulation Enable	Reserved	Electr. 60 - 31 (Segment ID depending on bits 7:0 of reg. 0x150)											
0x160: Electrode DAC select****	Reserved	Electr. 15 - 1 (Segment ID depending on bits 7:0 of reg. 0x150)											
0x164: Electrode DAC select****	Reserved	Electr. 30 - 16 (Segment ID depending on bits 7:0 of reg. 0x150)											
0x168: Electrode DAC select****	Reserved	Electr. 45 - 31 (Segment ID depending on bits 7:0 of reg. 0x150)											
0x16C: Electrode DAC select****	Reserved	Electr. 60 - 46 (Segment ID depending on bits 7:0 of reg. 0x150)											
0x170: Event Cnt. Max val	Trigger 1												
0x174: Event Cnt. Max val	Trigger 2												
0x178: Event Cnt. Max val	Trigger 3												
0x17C: Reserved	Reserved												
0x180: Event Cnt. Cur. val	Trigger 1												
0x184: Event Cnt. Cur. val	Trigger 2												
0x188: Event Cnt. Cur. val	Trigger 3												
0x18C: Reserved	Reserved												

0x190: Repeat Trigger # times	Trigger 1
0x194: Repeat Trigger # times	Trigger 2
0x198: Repeat Trigger # times	Trigger 3
0x19C: Reserved	Reserved
0x1A0: Repeat times counter	Trigger 1
0x1A4: Repeat times counter	Trigger 2
0x1A8: Repeat times counter	Trigger 3
0x1AC: Reserved	Reserved
0x1B0: Read Ptr. Cur. pos.	Stim 1
0x1B4: Read Ptr. Cur. pos.	Stim 2
0x1B8: Read Ptr. Cur. pos.	Stim 3
0x1BC: Read Ptr. Cur. pos.	Stim 4
0x1C0: Read Ptr. Cur. pos.	Stim 5
0x1C4: Read Ptr. Cur. pos.	Stim 6
0x1C8: Read Ptr. Cur. pos.	Stim 7

0x1CC: Read Ptr. Cur. pos.	Stim 8							
0x1D0: DAC Data Source select*****	Reserved	DAC F	DAC D	DAC B	Reserved	DAC E	DAC C	DAC A
0x1D4: SBS Data Source select*****	Reserved					SBS 3	SBS 2	SBS 1
0x1E0: DAC Weighting Factor	Reserved				DAC A			
0x1E4: DAC Weighting Factor	Reserved				DAC C			
0x1E8: DAC Weighting Factor	Reserved				DAC E			
0x1F0: DAC Weighting Factor	Reserved				DAC B			
0x1F4: DAC Weighting Factor	Reserved				DAC D			
0x1F8: DAC Weighting Factor	Reserved				DAC F			

SBS: Side Band Signal *Decoding Table:

```
00: Trigger 1
01: Trigger 2
10: Trigger 3
11: Reserved
```

**Decoding Table:

```
Bit 0: Enable Trigger: not Armed -> Armed
Bit 3:1:
      0: Stop stimulus sequence at recuring of same trigger event
      1: Restart stimulus sequence at recuring of same trigger event
      2: Ignore same trigger and continue processing
      3: Gate stimulus sequence at trigger event
Bit 5:4:
      0: Stop stimulus sequence at occuring of other trigger event
```

```
      1: Restart stimulus sequence at occuring of other trigger event
      2: Ignore other trigger and continue processing
bit 7-6: Status of Trigger statemachine (00: not Armed, 01: Armed, 10: Triggerd (running), 11: Reserved)
```

***Decoding Table:

```
00: SBS 1
01: SBS 2
10: SBS 3
11: Reserved, not Valid
```

****Decoding Table:

```
00: GND
01: DAC A/B
10: DAC C/D
11: DAC E/F
```

*****Decoding Table:

```
0000: Stimmulus 1 Data Stream
0001: Stimmulus 2 Data Stream
0010: Stimmulus 3 Data Stream
0011: Stimmulus 4 Data Stream
0100: Stimmulus 5 Data Stream
0101: Stimmulus 6 Data Stream
0110: Stimmulus 7 Data Stream
0111: Stimmulus 8 Data Stream
```

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x200: MEM Control Stim 1	Reserved		Init Ptr all**	Init Ptr Seg.0**	Reserved																				Segment Selector*							
0x204: Start Pointer Stim 1	Memory Pointer (DAC A and B select this Data source in reg. 0x1D0 by default)																															
0x208: End Pointer Stim 1	Memory Pointer																															
0x20C: Write Pointer Stim 1	Memory Pointer, write will clear Channel																															
0x210: Read Pointer Stim 1	Memory Pointer																															

0x220: MEM Control Stim 2	Reserved	Segment Selector*
0x224: Start Pointer Stim 2	Memory Pointer (SBS 1 select this Data source in reg. 0x1D4 by default)	
0x228: End Pointer Stim 2	Memory Pointer	
0x22C: Write Pointer Stim 2	Memory Pointer, write will clear Channel	
0x230: Read Pointer Stim 2	Memory Pointer	
0x240: MEM Control Stim 3	Reserved	Segment Selector*
0x244: Start Pointer Stim 3	Memory Pointer (DAC C and D select this Data source in reg. 0x1D0 by default)	
0x248: End Pointer Stim 3	Memory Pointer	
0x24C: Write Pointer Stim 3	Memory Pointer, write will clear Channel	
0x250: Read Pointer Stim 3	Memory Pointer	
0x260: MEM Control Stim 4	Reserved	Segment Selector*
0x264: Start Pointer Stim 4	Memory Pointer (SBS 2 select this Data source in reg. 0x1D4 by default)	
0x268: End Pointer Stim 4	Memory Pointer	

0x26C: Write Pointer Stim 4	Memory Pointer, write will clear Channel	
0x270: Read Pointer Stim 4	Memory Pointer	
0x280: MEM Control Stim 5	Reserved	Segment Selector*
0x284: Start Pointer Stim 5	Memory Pointer (DAC E and F select this Data source in reg. 0x1D0 by default)	
0x288: End Pointer Stim 5	Memory Pointer	
0x28C: Write Pointer Stim 5	Memory Pointer, write will clear Channel	
0x290: Read Pointer Stim 5	Memory Pointer	
0x2A0: MEM Control Stim 6	Reserved	Segment Selector*
0x2A4: Start Pointer Stim 6	Memory Pointer (SBS 3 select this Data source in reg. 0x1D4 by default)	
0x2A8: End Pointer Stim 6	Memory Pointer	
0x2AC: Write Pointer Stim 6	Memory Pointer, write will clear Channel	
0x2B0: Read Pointer Stim 6	Memory Pointer	
0x2C0: MEM Control Stim 7	Reserved	Segment Selector*

0x2C4: Start Pointer Stim 7	Memory Pointer (unused by default)	
0x2C8: End Pointer Stim 7	Memory Pointer	
0x2CC: Write Pointer Stim 7	Memory Pointer, write will clear Channel	
0x2D0: Read Pointer Stim 7	Memory Pointer	
0x2E0: MEM Control Stim 8	Reserved	Segment Selector*
0x2E4: Start Pointer Stim 8	Memory Pointer (unused by default)	
0x2E8: End Pointer Stim 8	Memory Pointer	
0x2EC: Write Pointer Stim 8	Memory Pointer, write will clear Channel	
0x2F0: Read Pointer Stim 8	Memory Pointer	

*Segments:

```
Segment 0 to 255 reflect the Segment ID 0 to 255 of Trigger
```

**Initialisation:

```
Poll Bit after writing a '1' until it is '0' to wait on end of request!
```

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x300: (WO)EEPRom Instruction Code Register	Reserved																							Fifo Reset	Instruction Code							
0x300: (RO)EEPRom Status Register	Reserved																					FIFO_empty	FIFO_full	Statemachine busy	Flash Status Register							
0x304 : EEPRom Memory Address Register	Reserved								Flash Address																							
0x308 : EEPRom Data FIFO Register	256 Data Bytes in 64 DWords to/from Flash																															
0x30C : EEPROM HW configuration register	Address length (1,2,3 Bytes)		Clock Divider Register (Multiples of 2 divide 38,4 MHz)																													

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x400: MEM Test Status Register	Reserved														MEM Test read/write in process	MEM Test failed	Reserved					MEM statemachine busy	Reserved								
0x404: MEM Test Register	Reserved													Stop Test	MEM Statemachine start	Write/notRead	Reserved							Patternmode*								
0x408: MEM Pattern Register	Pattern																															
0x40C: MEM Test Fail Counter Register	# of data compare fails (writes clear value)																															
0x410: MEM Test Address Ptr. Register	Reserved	Reflects the address Pointer to see the progress of the test (max Value: h"0FFFFF80")																														

0x414: First Fail Address Ptr. Register	Reserved	Reflects the address Pointer of the first failed address of mem test				
0x420: MEM Status Register	Reserved		MEM cal done (RO)	selfref. mode (RO)	Reserved	selfref. enter

*Pattermode options:

```
0: Counter
1: Pattern
2: Shift Pattern right
3: Shift Pattern left
4: Toggle Pattern after every write
```

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x500: Voltage Range Reg.	Value; Default: 0x0000 2EE0 (12000 mV)																															
0x504: Voltage Resolution Reg.	Value; Default: 0x8000 023B (571 uV)																															
0x508: Current Range Reg.	Value; Default: 0x0000 05DC (1500 uA)																															
0x50C: Current Resolution Reg.	Value; Default: 0x0000 0032 (50 nA)																															
0x520: DAC Offset correction	Reserved																DAC A offset															
0x524: DAC Offset correction	Reserved																DAC C offset															
0x528: DAC Offset correction	Reserved																DAC E offset															
0x530: DAC Offset correction	Reserved																DAC B offset															
0x534: DAC Offset correction	Reserved																DAC D offset															
0x538: DAC Offset correction	Reserved																DAC F offset															

Register	Byte 4	Byte 3	Byte 2	Byte 1
0xF00: MEM Write Address Register	MEM Address(RW)			
0xF04: MEM Data Register	Write Data (WO), Read Data(RO)			
0xF08 - 0xF1C: Reserved	Reserved			
0xF20: Write Stim 1 Data Register	Channel 0 Data Vector*			
0xF24: Write Stim 2 Data Register	Channel 1 Data Vector*			
0xF28: Write Stim 3 Data Register	Channel 2 Data Vector*			
0xF2C: Write Stim 4 Data Register	Channel 3 Data Vector*			
0xF30: Write Stim 5 Data Register	Channel 4 Data Vector*			
0xF34: Write Stim 6 Data Register	Channel 5 Data Vector*			
0xF38: Write Stim 7 Data Register	Channel 6 Data Vector*			
0xF3C: Write Stim 8 Data Register	Channel 7 Data Vector*			
0xF40: Clear and Write Stim 1 Data Register	Channel 0 Data Vector*			
0xF44: Clear and Write Stim 2 Data Register	Channel 1 Data Vector*			
0xF48: Clear and Write Stim 3 Data Register	Channel 2 Data Vector*			
0xF4C: Clear and Write Stim 4 Data Register	Channel 3 Data Vector*			
0xF50: Clear and Write Stim 5 Data Register	Channel 4 Data Vector*			
0xF54: Clear and Write Stim 6 Data Register	Channel 5 Data Vector*			
0xF58: Clear and Write Stim 7 Data Register	Channel 6 Data Vector*			
0xF5C: Clear and Write Stim 8 Data Register	Channel 7 Data Vector*			

Sideband Data:

```
Bit 0: Amplifier Protection Switch on Headstage/Blanking
Bit 3: Stimulation Switch
Bit 4: Stimulus Selector
```

*Data Vector decoding:

```
 Bit 31: Reserved
 Bit 30 – 28:
     000: DAC/SBS Data Vector
     001: Loopptr. Vector
     010: Long Loop Ptr. Vector
     011: Long Loop Ctr. Vector
     100: Reserved
     ...
     110: Reserved
     111: END Command
```

```
 DAC/SBS Data Vector(000):
 Bit 27: Reserved
 Bit 26: Repeat Timebase (0: 20 us, 1: 1000*20us)
 Bit 25 – 16: Number of Repeats (0: Pattern is used 1x Timebase; 1: Pattern is used for 2x Timebase; ...)
 Bit 15 – 0 : DAC data value (unsigned 16 bit value, 0x8000 is zero level) / SBS data value
```

```
    SBS Bit 0    : Amplifier Protection Switch/Blanking
    SBS Bit 3    : Stimulation Switch
    SBS Bit 4    : Stimulus Select
    SBS Bit 8-15 : Electrode Config ID
```

```
Loop Ptr. Vector(001):
Bit 27 - 26: Loop Level
Bit 25 - 16: Number of Repeats (2: Vectors are repeated once, thus used twice)
Bit 15 - 0 : Address Offset (Number of Vectors to jump backward, 1: One Vector before the LoopPtr is repeated)
```

```
Long Loop Ptr. Vector(010):
Bit 27 - 0 : Address Offset (Number of Vectors to jump backward)
```

```
Long Loop Ctr. Vector(011):
Bit 27 - 0 : Number of Repeats
```

```
End Command(111):
Bit 27 - 0 : Reserved
```

HS <-> STG interconection (bits 32-0)

Pin	Name	Bit	HS FPGA Pin	Function	to FPGA: STG FPGA Pin
3	STG01	0	V1	Reset	X2
4	STG02	1	V2	Suspend	X2
5	STG03	2	T3	Reserved	X2
6	STG04	3	U1	Reserved	X2
7	STG05	4	T1	Reserved	X2
8	STG06	5	V3	Reserved	X2
9	STG07	6	P3	Reserved	X2
10	STG08	7	R1	Blanking	X2
11	STG09	8	V3	50 kHz Impulse	X2
12	STG10	9	W1	SPI: read data ready	X2
13	STG11	10	Y1	SPI: SCLK	X2
14	STG12	11	Y2	SPI: CS	X2
15	STG13	12	W3	SPI: MOSI	X2
16	STG14	13	AA1	SPI: MISO	X2
17	STG15	14	AA2	25.6 MHz clock	X1,X2
18	STG16	15	Y3	400 kHz Power sync.	X1,X2
19	STG17	16	H6	400 kHz Power sync.	X1,X2
20	STG18	17	H4	38.4 MHz clock	X1,X2
21	STG19	18	G3	Reset	X1
22	STG20	19	H8	Suspend	X1
23	STG21	20	G6	Reserved	X1

24	STG22	21	G4	Reserved	X1
25	STG23	22	F5	Reserved	X1
26	STG24	23	G7	Reserved	X1
27	STG25	24	H5	Reserved	X1
28	STG26	25	J7	Blanking	X1
29	STG27	26	J3	50 kHz	X1
30	STG28	27	J4	SPI: read data ready	X1
31	STG29	28	J6	SPI: SCLK	X1
32	STG30	29	K4	SPI: CS	X1
33	STG31	30	K5	SPI: MOSI	X1
34	STG32	31	K6	SPI: MISO	X1

Quellennachweise

[1] http://www.ti.com/product/tms320c6454

Quelle(n) und Bearbeiter des/der Artikel(s)

MEA2100 User Guide *Quelle*: http://wiki.mcs.de.com/index.php?oldid=24618 *Bearbeiter*: Jesinger, Kurnoth, 412 anonyme Bearbeitungen

Quelle(n), Lizenz(en) und Autor(en) des Bildes

Image:MEA2100_Overview.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MEA2100_Overview.png *Lizenz*: unbekannt *Bearbeiter*: Jesinger, Kurnoth

Image:MEA2100_OverviewDetail.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MEA2100_OverviewDetail.png *Lizenz*: unbekannt *Bearbeiter*: Jesinger, Kurnoth

Image:MEA2100_DataStream_noDSP.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MEA2100_DataStream_noDSP.png *Lizenz*: unbekannt *Bearbeiter*: Jesinger

Image:MEA2100_DataStream_withDSP.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MEA2100_DataStream_withDSP.png *Lizenz*: unbekannt *Bearbeiter*: Jesinger

Image:MEA2100_StimulationBlanking.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MEA2100_StimulationBlanking.png *Lizenz*: unbekannt *Bearbeiter*: Jesinger, Kurnoth

Image:Memmory Layout.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:Memmory_Layout.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:MEA2100_Blanking.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MEA2100_Blanking.png *Lizenz*: unbekannt *Bearbeiter*: Jesinger, Kurnoth

Image:STG_Mux_Stimswtich_control.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:STG_Mux_Stimswtich_control.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:STG_Config_Overview.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:STG_Config_Overview.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:MEA2100_Interface Board HS1_1.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MEA2100_Interface_Board_HS1_1.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:MEA2100_Interface Board direct streaming mode.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MEA2100_Interface_Board_direct_streaming_mode.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:MCS_USB_Update_MEA2100_Digital_screen.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MCS_USB_Update_MEA2100_Digital_screen.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:MCS_USB_Update_MEA2100_Filter_screen.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MCS_USB_Update_MEA2100_Filter_screen.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:MCS_USB_Update_MEA2100_initial_screen.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MCS_USB_Update_MEA2100_initial_screen.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:MCS_USB_Update_MEA2100_Interfaceboard_screen.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MCS_USB_Update_MEA2100_Interfaceboard_screen.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:MCS_USB_Update_MEA2100_Headstage_screen.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MCS_USB_Update_MEA2100_Headstage_screen.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:MCS_USB_Update_MEA2100_STG_screen.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MCS_USB_Update_MEA2100_STG_screen.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth

Image:MCS_USB_Update_MEA2100_STGMEM_screen.png *Quelle*: http://wiki.mcs.de.com/index.php?title=Datei:MCS_USB_Update_MEA2100_STGMEM_screen.png *Lizenz*: unbekannt *Bearbeiter*: Kurnoth