AGIPD 1.0
Tests & Plans

Ulrich Trunk

AGIPD 1.0 Functional Testing

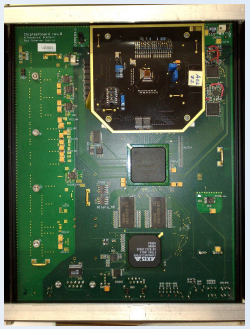
	Satus	Results
Interface	works	works
Memory Addressing	works	works
Readout	works	last data pulse possibly truncated (can be overcome by firmware)
Readout	works	different speeds on ports (under investigation)
Noise	ongoing	~260e RMS
Dynamic Range	ongoing	calibration required

AGIPD 1.0 Command Parser

Translates „Assembly Language style“ commands to AGIPD binary code

```
VTHCMP 106     \106 default
IBPCDS 211
VPC1LO 134
VPC1HI 2
VPC2LO 30
VPC2HI 03
SWPXBI 255     \> default was 106
SWROBI 255     \> default was 224
SWCALV 3       \> default was 3
ROSHFT 6
SETUPR 8       \ masks 40 40 8
LENGTH 24
TIROWM 0       \ masks 255 255 0
TICOLM 0       \ masks 85 170 0
ACQMEM 172
ACQMEM 173
ACQMEM 174
ACQMEM 175
SETMEM 174     \ still needs to be 24 cycles long
LENGTH 128     \ default 48
SETMEM 174
RPAMPN 30 29   \ precharge first row
RPAMPA 31 30
RPAMPA 32 31
RPAMPA 33 32
RPNMPA 34 33
```

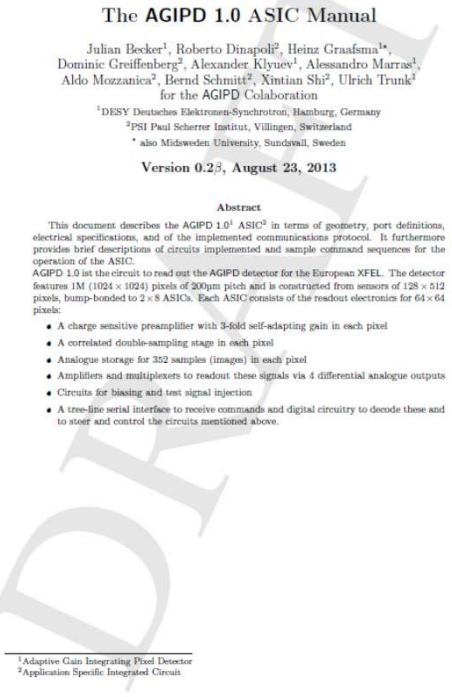
Parser



```
00010000000000000000000000000000
00010000000000000000000000000000
C startat 0
C flipmask 0xF0000000
00010000000000000100000000000000
00011000000000000100000000000000
00010000000000000100000000000000
00011000000000000100000000000000
00010000000000000100000000000000
00011000000000000100000000000000
00010000000000000100000000000000
00011000000000000100000000000000
00010000000000000100000000000000
00011000000000000100000000000000
00010000000000000100000000000000
00011000000000000100000000000000
00010000000000000100000000000000
00011000000000000100000000000000
00010000000000000100000000000000
00011000000000000100000000000000
00010000000000000100000000000000
```

AGIPD 1.0 Documentation

The AGIPD 1.0 Manual (~180 pages!) is available at:
http://www.desy.de/~trunk/AGIPD10_Manual.pdf



The **AGIPD 1.0 ASIC Manual**

Julian Becker[1], Roberto Dinapoli[2], Heinz Graafsma[1*], Dominic Greiffenberg[2], Alexander Klyuev[1], Alessandro Marras[1], Aldo Mozzanica[2], Bernd Schmitt[2], Xintian Shi[2], Ulrich Trunk[1] for the AGIPD Colaboration

[1]DESY Deutsches Elektronen-Synchrotron, Hamburg, Germany
[2]PSI Paul Scherrer Institut, Villingen, Switzerland
[*] also Midsweden University, Sundsvall, Sweden

Version 0.2β, August 23, 2013

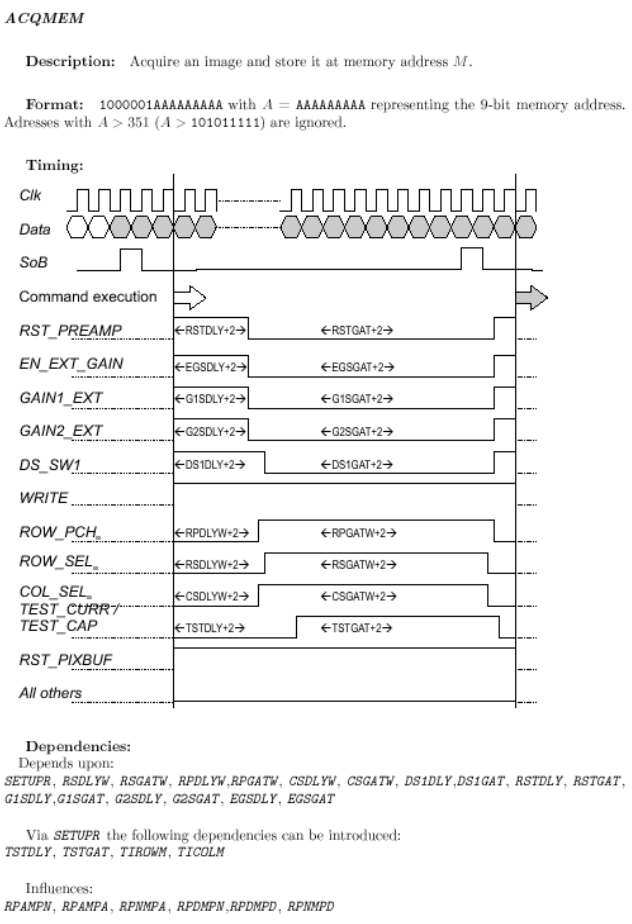
Abstract

This document describes the AGIPD 1.0[1] ASIC[2] in terms of geometry, port definitions, electrical specifications, and of the implemented communications protocol. It furthermore provides brief descriptions of circuits implemented and sample command sequences for the operation of the ASIC.
AGIPD 1.0 ist the circuit to read out the AGIPD detector for the European XFEL. The detector features 1M (1024×1024) pixels of 200µm pitch and is constructed from sensors of 128×512 pixels, bump-bonded to 2×8 ASICs. Each ASIC consists of the readout electronics for 64×64 pixels:

- A charge sensitive preamplifier with 3-fold self-adapting gain in each pixel
- A correlated double-sampling stage in each pixel
- Analogue storage for 352 samples (images) in each pixel
- Amplifiers and multiplexers to readout these signals via 4 differential analogue outputs
- Circuits for biasing and test signal injection
- A tree-line serial interface to receive commands and digital circuitry to decode these and to steer and control the circuits mentioned above.

[1]Adaptive Gain Integrating Pixel Detector
[2]Application Specific Integrated Circuit

1



ACQMEM

Description: Acquire an image and store it at memory address M.

Format: `1000001AAAAAAAAA` with $A =$ `AAAAAAAAA` representing the 9-bit memory address. Adresses with $A > 351$ ($A >$ `101011111`) are ignored.

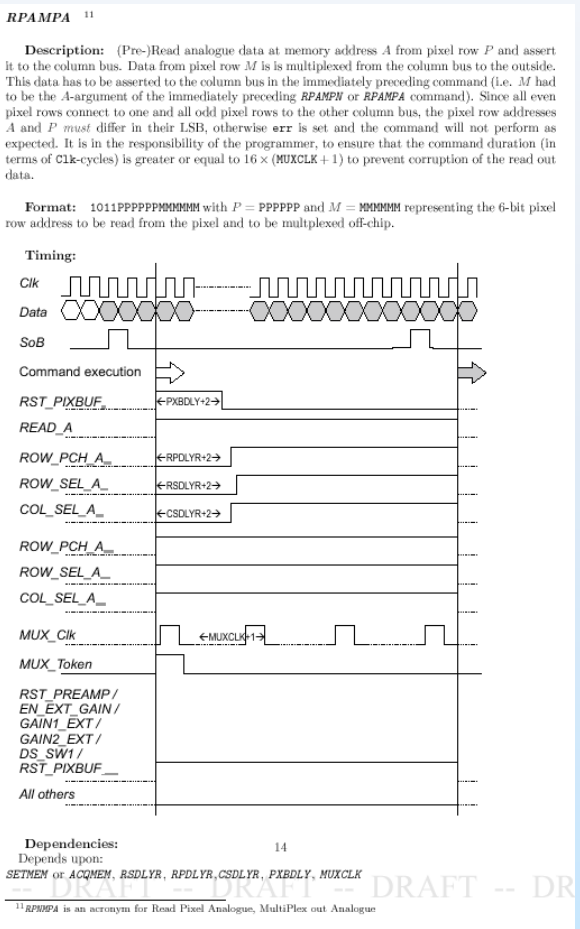
Timing:

Clk
Data
SoB
Command execution
RST_PREAMP ←RSTDLY+2→ ←RSTGAT+2→
EN_EXT_GAIN ←EGSDLY+2→ ←EGSGAT+2→
GAIN1_EXT ←G1SDLY+2→ ←G1SGAT+2→
GAIN2_EXT ←G2SDLY+2→ ←G2SGAT+2→
DS_SW1 ←DS1DLY+2→ ←DS1GAT+2→
WRITE
ROW_PCH_ ←RPDLYW+2→ ←RPGATW+2→
ROW_SEL_ ←RSDLYW+2→ ←RSGATW+2→
COL_SEL_ ←CSDLYW+2→ ←CSGATW+2→
TEST_CURR / TEST_CAP ←TSTDLY+2→ ←TSTGAT+2→
RST_PIXBUF
All others

Dependencies:
Depends upon:
SETUPR, *RSDLYW*, *RSGATW*, *RPDLYW*,*RPGATW*, *CSDLYW*, *CSGATW*, *DS1DLY*,*DS1GAT*, *RSTDLY*, *RSTGAT*, *G1SDLY*,*G1SGAT*, *G2SDLY*, *G2SGAT*, *EGSDLY*, *EGSGAT*

Via *SETUPR* the following dependencies can be introduced:
TSTDLY, *TSTGAT*, *TIROWM*, *TICOLM*

Influences:
RPAMPN, *RPAMPA*, *RPNMPA*, *RPDMPN*,*RPDMPD*, *RPNMPD*



RPAMPA [11]

Description: (Pre-)Read analogue data at memory address A from pixel row P and assert it to the column bus. Data from pixel row M is is multiplexed from the column bus to the outside. This data has to be asserted to the column bus in the immediately preceding command (i.e. M had to be the A-argument of the immediately preceding *RPAMPN* or *RPAMPA* command). Since all even pixel rows connect to one and all odd pixel rows to the other column bus, the pixel row addresses A and P *must* differ in their LSB, otherwise `err` is set and the command will not perform as expected. It is in the responsibility of the programmer, to ensure that the command duration (in terms of `Clk`-cycles) is greater or equal to $16 \times$ (`MUXCLK` $+ 1$) to prevent corruption of the read out data.

Format: `1011PPPPPPMMMMMM` with $P =$ `PPPPPP` and $M =$ `MMMMMM` representing the 6-bit pixel row address to be read from the pixel and to be multplexed off-chip.

Timing:

Clk
Data
SoB
Command execution
RST_PIXBUF_ ←PXBDLY+2→
READ_A
ROW_PCH_A_ ←RPDLYR+2→
ROW_SEL_A_ ←RSDLYR+2→
COL_SEL_A_ ←CSDLYR+2→
ROW_PCH_A__
ROW_SEL_A_
COL_SEL_A__
MUX_Clk ←MUXCLK+1→
MUX_Token
RST_PREAMP / EN_EXT_GAIN / GAIN1_EXT / GAIN2_EXT / DS_SW1 / RST_PIXBUF__
All others

Dependencies:
Depends upon:
SETMEM or *ACQMEM*, *RSDLYR*, *RPDLYR*,*CSDLYR*, *PXBDLY*, *MUXCLK*

14

[11]*RPNMPA* is an acronym for Read Pixel Analogue, MultiPlex out Analogue

DRAFT -- DRAFT -- DRAFT -- DR

AGIPD 1.0 Measurement Plans

- define bias points
- identify a “Golden Device”
  - Produce o(5) boards with chips from different sites on wafer
- measure dynamic range
  - with pulsed capacitor and laser
- access nonlinearity
  - @ first point after gain switching
  - @ end of dynamic range
    - crosscheck with AGIPD 0.5
    - investigate suitable fits (3rd order polynomial?)
    - investigate possible reasons
      - diode bias?
- Identify cause for different speeds on different outputs/ADCs
  - speed map
  - investigate ADCs/buffers on chiptester box
- Define and implement an algorithm for wafer-testing