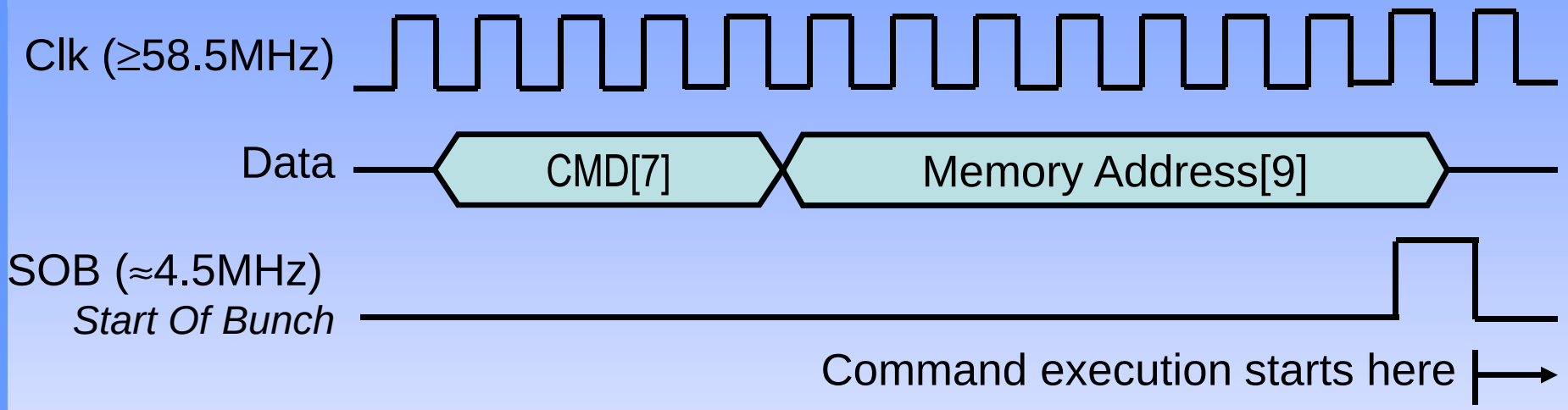
AGIPD 1.0 Periphery Circuit

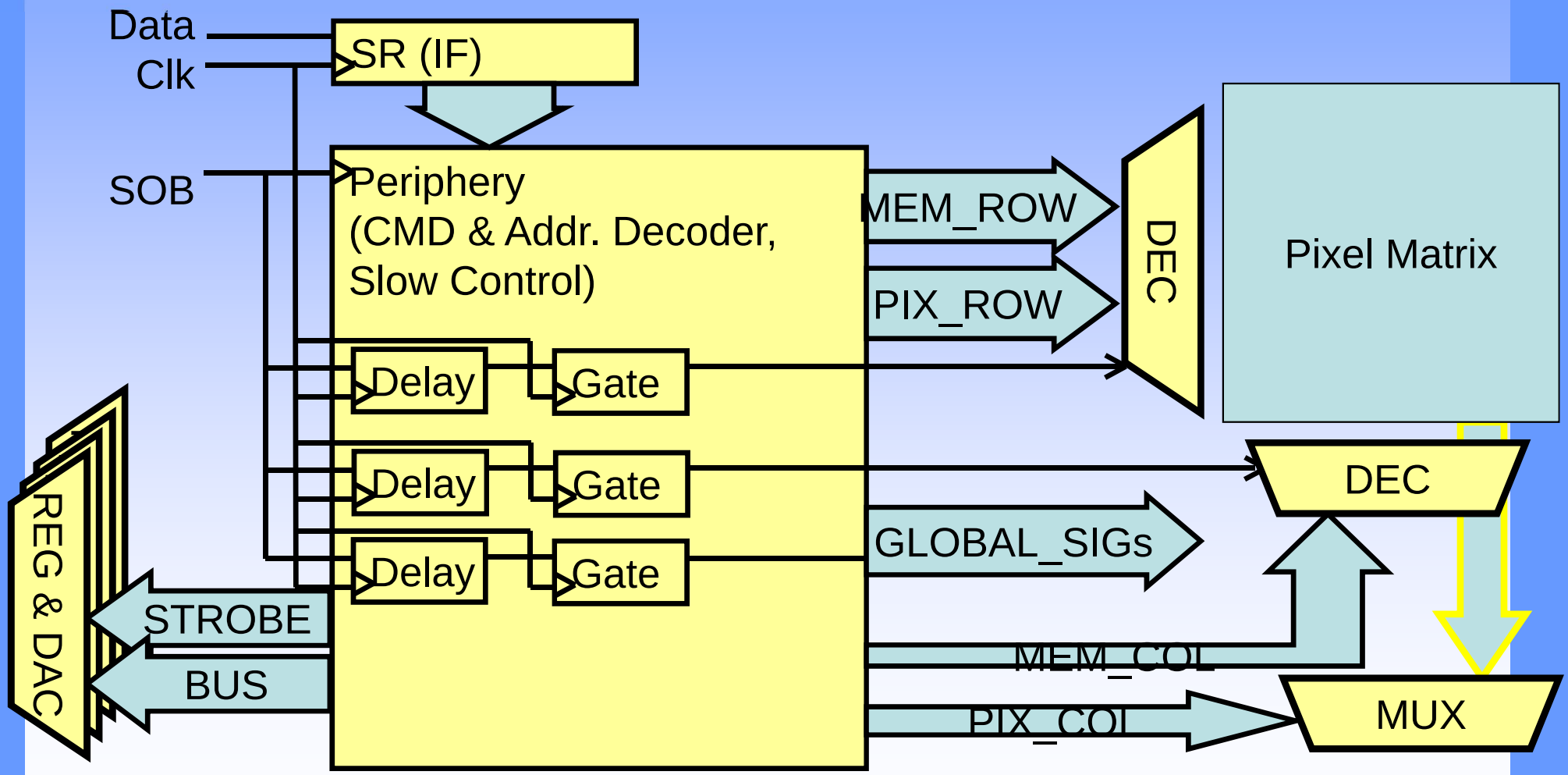
Ulrich Trunk

AGIPD Decoder based Periphery

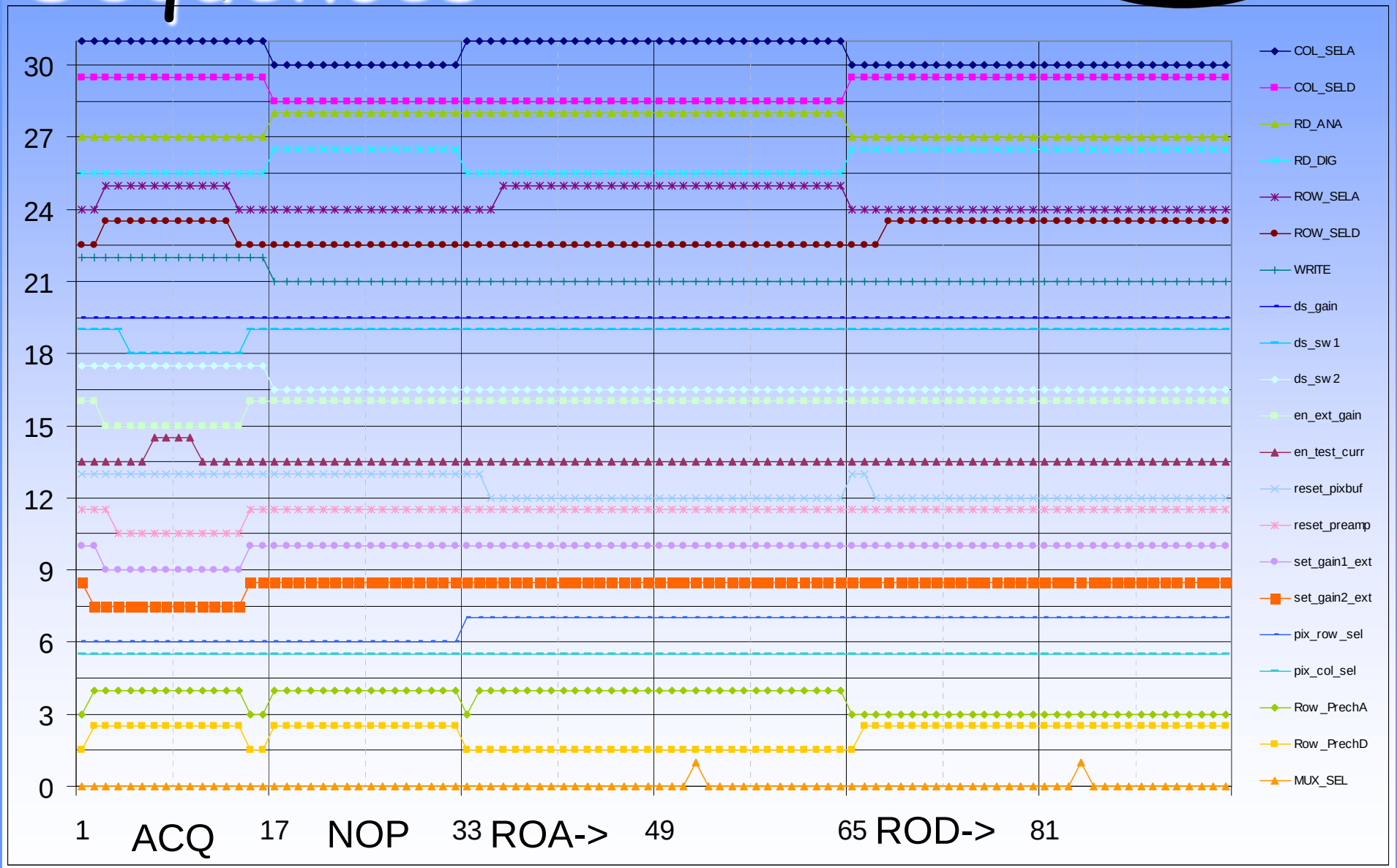


- 16bit Commands
- SoB-rate = frame rate:
- $f_{clk} \geq 16 \times f_{SoB} \approx 72$ MHz
- Readout in 99.4ms:
- $f_{clk} \geq (352_{frames} \times 2 \times (64+3)_{pixel}) \times 16/99.4ms \approx 7.6$ MHz

AGIPD Decoder based Periphery



AGIPD Periphery: Sequences



AGIPD Periphery Circuit

Commands

```
// Cmd[8]        VAL[8]                          |Cmd[8]                                                    VAL[8]
// F E D C B A 9 8 7 6 5 4 3 2 1 0      <-Bit#
// 1 0 0 0 0 0 0 M M M M M M M M M      |Cmd               SET MEM / NOP                                     MemAddr (no write operation is performed!)
// 1 0 0 0 0 0 1 M M M M M M M M M      |Cmd               Write                                             MemAddr (Sampling to MemAddr)
// 1 0 1 0 0 P P P P P C C C C C C      |READ_PIX_ROW_A PREF MUX PrefPixRow[6] MuxPixRow[6] (Prefetch only -> read mem and drive signals on col bus, MuxPixRow is ignored)
// 1 0 1 1 0 P P P P P C C C C C C      |READ_PIX_ROW_A PREF MUX PrefPixRow[6] MuxPixRow[6] (Prefetch PrefetchPixRow to bus and mux MuxPixRow to output)
// 1 0 0 1 0 P P P P P C C C C C C      |READ_PIX_ROW_A PREF MUX PrefPixRow[6] MuxPixRow[6] (Mux only -> MuxPixRow to output, PrefPixRow is ignored)
// 1 1 1 0 0 P P P P P C C C C C C      |READ_PIX_ROW_D PREF MUX PrefPixRow[6] MuxPixRow[6] (Prefetch only -> read mem and drive signals on col bus, MuxPixRow is ignored)
// 1 1 1 1 0 P P P P P C C C C C C      |READ_PIX_ROW_D PREF MUX PrefPixRow[6] MuxPixRow[6] (Prefetch PrefetchPixRow to bus and mux MuxPixRow to output)
// 1 1 0 1 0 P P P P P C C C C C C      |READ_PIX_ROW_D PREF MUX PrefPixRow[6] MuxPixRow[6] (Mux only -> MuxPixRow to output, PrefPixRow is ignored)
// 0 1 0 0 0 0 0 0 R B B B B B B B      |Cmd               SETUP                                                                  Reset,StatusReg Bits
// 0 1 0 0 0 0 0 0 1 0 V V V V V V      |Cmd               SET DELAY 0                                       0, Delay (Adjust timing of control signals like ampReset, ds1 etc. wrt.
to SoB (Start of Bunch))
// 0 1 0 0 0 0 0 0 1 1 V V V V V V      |Cmd               SET GATE 0                                        1, Gate Length      ROW_SEL (for write)
// 0 1 0 0 0 0 1 0 0 0 V V V V V V      |Cmd               SET DELAY 1                                       0, Delay            ROW_SEL (for read)
// 0 1 0 0 0 0 1 0 1 0 V V V V V V      |Cmd               SET GATE 1                                        1, Gate Length
// 0 1 0 0 0 0 1 1 0 0 V V V V V V      |Cmd               SET DELAY 2                                       0, Delay            ROW_PCH (for write)
// 0 1 0 0 0 0 1 1 1 0 V V V V V V      |Cmd               SET GATE 2                                        1, Gate Length
// 0 1 0 0 0 1 0 0 0 0 V V V V V V      |Cmd               SET DELAY 3                                       0, Delay            ROW_PCH (for read)
// 0 1 0 0 0 1 0 0 1 0 V V V V V V      |Cmd               SET GATE 3                                        1, Gate Length
// 0 1 0 0 0 1 0 1 0 0 V V V V V V      |Cmd               SET DELAY 4                                       0, Delay            COL_SEL (for write)
// 0 1 0 0 0 1 0 1 1 0 V V V V V V      |Cmd               SET GATE 4                                        1, Gate Length
// 0 1 0 0 0 1 1 0 0 0 V V V V V V      |Cmd               SET DELAY 5                                       0, Delay            COL_SEL (for read)
// 0 1 0 0 0 1 1 0 1 0 V V V V V V      |Cmd               SET GATE 5                                        1, Gate Length
// 0 1 0 0 0 1 1 1 0 0 V V V V V V      |Cmd               SET DELAY 6                                       0, Delay            DS1 timing
// 0 1 0 0 0 1 1 1 1 0 V V V V V V      |Cmd               SET GATE 6                                        1, Gate Length
// 0 1 0 0 1 0 0 0 0 0 V V V V V V      |Cmd               SET DELAY 7                                       0, Delay            Preamp reset timing
// 0 1 0 0 1 0 0 0 1 0 V V V V V V      |Cmd               SET GATE 7                                        1, Gate Length
// 0 1 0 0 1 0 0 1 0 0 V V V V V V      |Cmd               SET DELAY 8                                       0, Delay            Gain1 switch timing (reset)
// 0 1 0 0 1 0 0 1 1 0 V V V V V V      |Cmd               SET GATE 8                                        1, Gate Length
// 0 1 0 0 1 0 1 0 0 0 V V V V V V      |Cmd               SET DELAY 9                                       0, Delay            Gain2 switch timing (reset)
// 0 1 0 0 1 0 1 0 1 0 V V V V V V      |Cmd               SET GATE 9                                        1, Gate Length
// 0 1 0 0 1 0 1 1 0 0 V V V V V V      |Cmd               SET DELAY A                                       0, Delay            Ext_gain timing (reset)
// 0 1 0 0 1 0 1 1 1 0 V V V V V V      |Cmd               SET GATE A                                        1, Gate Length
// 0 1 0 0 1 1 0 0 0 0 V V V V V V      |Cmd               SET DELAY B                                       0, Delay            Pixbuf reset timing
// 0 1 0 0 1 1 0 0 1 0 V V V V V V      |Cmd               SET GATE B                                        1, Gate Length
// 0 1 0 0 1 1 0 1 0 0 V V V V V V      |Cmd               SET DELAY C                                       0, Delay            Test pulse timing
// 0 1 0 0 1 1 0 1 1 0 V V V V V V      |Cmd               SET GATE C                                        1, Gate Length
// 0 1 0 0 1 1 1 0 0 0 V V V V V V      |Cmd               SET DELAY D                                       0, Delay
// 0 1 0 0 1 1 1 0 1 0 V V V V V V      |Cmd               SET GATE D                                        1, Gate Length
// 0 1 0 0 1 1 1 1 0 0 V V V V V V      |Cmd               SET DELAY E                                       0, Delay
// 0 1 0 0 1 1 1 1 1 0 V V V V V V      |Cmd               SET GATE E                                        1, Gate Length
// 0 1 0 1 0 0 0 0 0 V V V V V V V      |Cmd               SET REG/DAC 0F                                    Value (Registers like DAC settings etc.) MUX_Clk_Div
// 0 1 0 1 0 0 0 0 1 V V V V V V V      |Cmd               SET REG/DAC 10                                    Value TI_ROW_Mask - Test injector pattern
// 0 1 0 1 0 0 1 0 0 V V V V V V V      |Cmd               SET REG/DAC 11                                    Value TI_COL_Mask - Test injector pattern
// 0 1 0 1 0 0 1 1 0 V V V V V V V      |Cmd               SET REG/DAC 12                                    Value
// 0 1 0 1 0 1 0 0 0 V V V V V V V      |Cmd               SET REG/DAC 13                                    Value
// 0 1 0 1 0 1 0 1 0 V V V V V V V      |Cmd               SET REG/DAC 14                                    Value

.................

// 0 1 1 0 1 1 1 0 V V V V V V V      |Cmd               SET REG/DAC 2D                                    Value
// 0 1 1 0 1 1 1 1 V V V V V V V      |Cmd               SET REG/DAC 2E                                    Value
// 0 0 0 0 0 0 0 0 V V V V V V V      |Spare REG00....
// 0 0 1 1 1 1 1 V V V V V V V V      |Sapre REG1F (32)

// Status Register Bits: Reset,err,n.c.,en_test_inj,ds_gain,en_ext_gain,set_gain1,set_gain2
```

AGIPD Periphery Circuit
Signal Combinatorics....

	GLOBAL	ROW	DECODER	STATIC	TIMER	REGBIT	DEFINITION
COL_SEL_A	X		X	X	X		(MemDec1 & ((Write & Gate[4]) | (RD_ANA & Gate[5])))|MemDec2
COL_SEL_D	X		X	X	X		(MemDec1 & ((Write & Gate[4]) | (RD_DIG & Gate[5])))|MemDec2
RD_ANA		X		X			!PIX_ROW_SEL | ANA
RD_DIG		X		X			!PIX_ROW_SEL | DIG
ROW_SEL_A		X	X		X		MemDec & ((Gate[0] & Write) | (RO_ANA & PIX_ROW1_SEL & Gate[1])|PIX_ROW2_SEL)
ROW_SEL_D		X	X		X		MemDec & ((Gate[0] & Write) | (RO_DIG & PIX_ROW1_SEL & Gate[1])|PIX_ROW2_SEL)
WRITE	X			X			Write
ds_gain	X			X		X	StatusReg[3]
ds_sw1	X				X		!(Write & Gate[6])
ds_sw2	X			X			Write
en_ext_gain	X				X	X	StatusRegBit[2] | !Write | !Gate[A]
test_inj	X				X	X	Write & StatusRegBit[4] & Gate[C]
reset_pixbuf		X			X		!Read | !(Gate[B] & PIX_ROW_SEL1)|!PIX_ROW_SEL2)
reset_preamp	X				X		!(Write & Gate[7])
set_gain1_ext	X				X		StatusReg2Bit[0] | !Write | !Gate[8]
set_gain2_ext	X				X		StatusReg2Bit[1] | !Write | !Gate[9]
PIX_ROW_SEL		X	X	X			(PIX_ROW_SEL1 | PIX_ROW_SEL2) & Read
PIX_COL_SEL	X			X			1 (obsolete)
ROW_PCH_A		X	X		X		(MemDec1 & ((Write & Gate[2]) | (RD_ANA & Gate[3])))|MemDec2
ROW_PCH_D		X	X		X		(MemDec1 & ((Write & Gate[2]) | (RD_DIG & Gate[3])))|MemDec2
MUX_SEL			X		X		(MUX_CNTR,PIX_ROW2[0]) & (ANA | DIG)
RegBus							
RegStrobe							

AGIPD Periphery Circuit

Status

Module	Fkt.	ex.	done	sim	comment
AGIPD10_Periphery.v	top	x	-	-	moving target
AGIPD10_CIFP.v	Interpreter	x	-	-	moving target
P1M32_C_PeriDec.v	col. dec.	x	x	-	
P64M11_R_PeriDec.v	row dec.	x	x	-	
Mux32_C_PeriDec.v	mux	x	x	-	TBR
Mux_Adr_Gen.v	mux	x	x	-	TBR