

ÖAW

AUSTRIAN
ACADEMY OF
SCIENCES



RD50 HV-CMOS Meeting

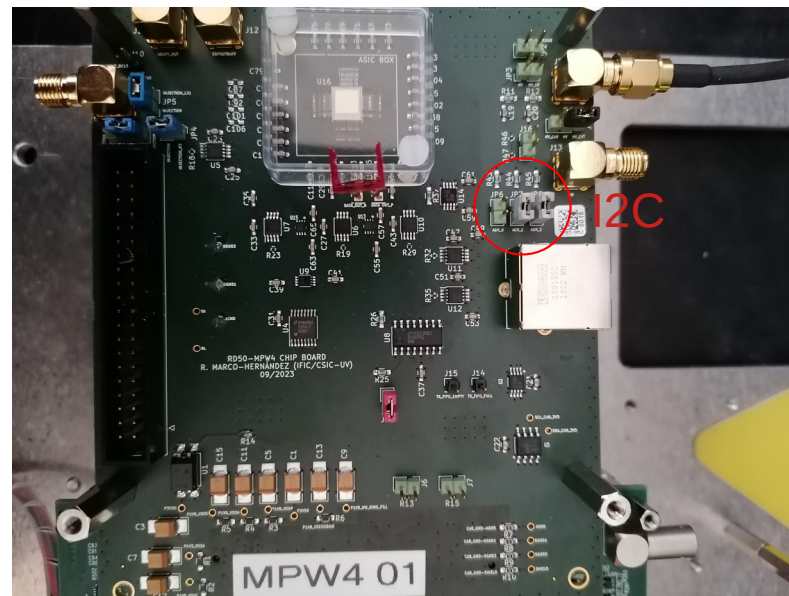
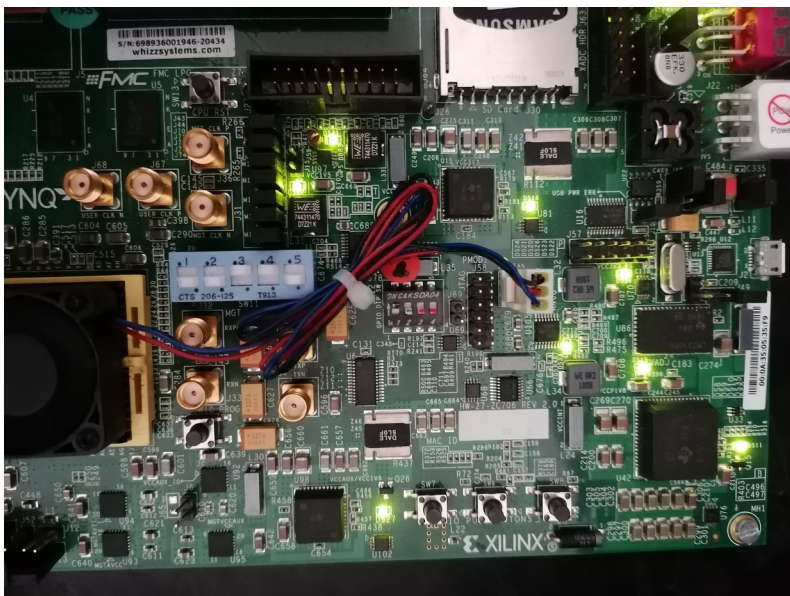
RD50-MIPW4

How to DAQ

Bernhard Pils

Jumper / DIP-Switch settings

- Default I2C address set to 0x41
 - Jumpers in picture set to this value
 - Can be changed in Peary config file (if needed)

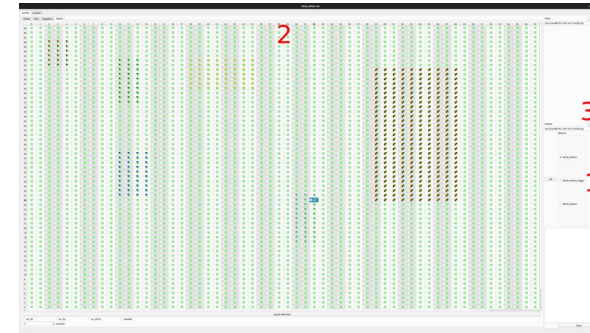


GUI changes

- Interest by Caribou community lead to „generalization“ of our GUI
- The GUI now needs a config file itself (specify with “-c”)
 - Either you start GUI without providing any → Wizard will help you set up one
 - For RD50-MPW x ($x \geq 3$) you’ll find one in *rd50-mpw3-gui/misc/template_gui.json*
- Commands also stored in config
 - No need for *Get Commands* anymore
- We now use *pearyd* executable
 - Runs more stable / less buggy than earlier hacky (*pearycli-*) SSH solution

GUI config tab

- 1) Choose which device to configure
- 2) Set desired config options
 - DAC values in *Power* tab
 - I2C address,... in *Misc* tab
 - Chip registers in *Registers* tab
 - Matrix config by checking pixel checkboxes and alter values of mask, TDAC,... with inputs beneath



- 3) Deploy to Caribou system

Config Control			
Power	Misc	Registers	Matrix
	power	U [V]	I_max [A]
1	bl	0.9	3
2	del_hi	0.7	3
3	del_lo	0.9	3
4	p1v3_vssa	1.3	3
5	p1v8_nw_ring	1.8	3
6	p1v8_vdd1	1.8	3
7	p1v8_vdda	1.8	3
8	p1v8_vddc	1.8	3
9	p1v8_vsensbus	1.8	3
10	p2v5d	2.5	3
11	p3v3_base	3.3	3
12	th	0.95	3

Config Control		
Power	Misc	
Registers	Matrix	
key	value	
1	calib_file	calib_base.txt
2	config_si5345	clock_config.txt
3	en_freeze	1
4	execute_file	execute.txt
5	i2c_addr	0x41
6	i2c_dev	/dev/i2c-9
7	inj_t0	0
8	matrix_config	matrix_config_base.txt

Config Control		
Power	Misc	
Registers	Matrix	
register	value	
1	conf_reg_ts_ctrl	0x00
2	conf_reg_ts_ini	0x00
3	cu_ctrl	0x00
4	en_ext_ctrl	0x00
5	en_ser_out_dcol	0x00
6	en_sfout_dcol	0x00
7	ideo	0xfb
8	idle1	0xf7
9	idle2	0xf7
10	idle3	0xf7
11	tx_ctrl	0x00
12	vfb	0x26
13	vbr	0x26
14	vn	0x15
15	vnb	0x12
16	vnsensbias	0x32
17	vnsf	0x2d
18	vpbias	0x25
19	vpcomp	0x13
20	vprim	0x24

GUI control tab

- 1) Choose device
- 2) Specify and connect to host
- 3) Power
- 4) Configure
- 5) Choose command to execute
 - With proper arguments
- 6) Execute and observe output
 - Alternatively use *pearycli* and not GUI to operate

