DEEP UNDERGROUND
NEUTRINO EXPERIMENT

University of Pennsylvania

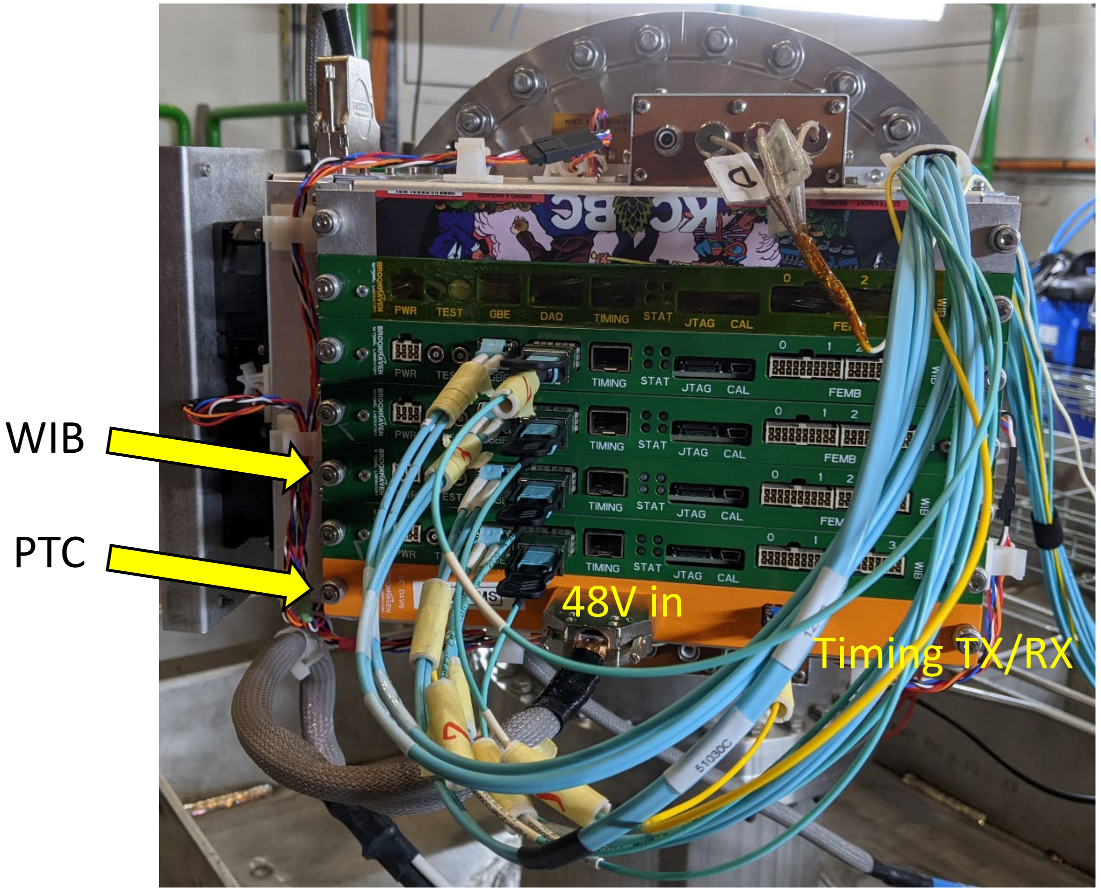
PTCv4 Status at ICEBERG

Adrian Nikolica

25 January 2024

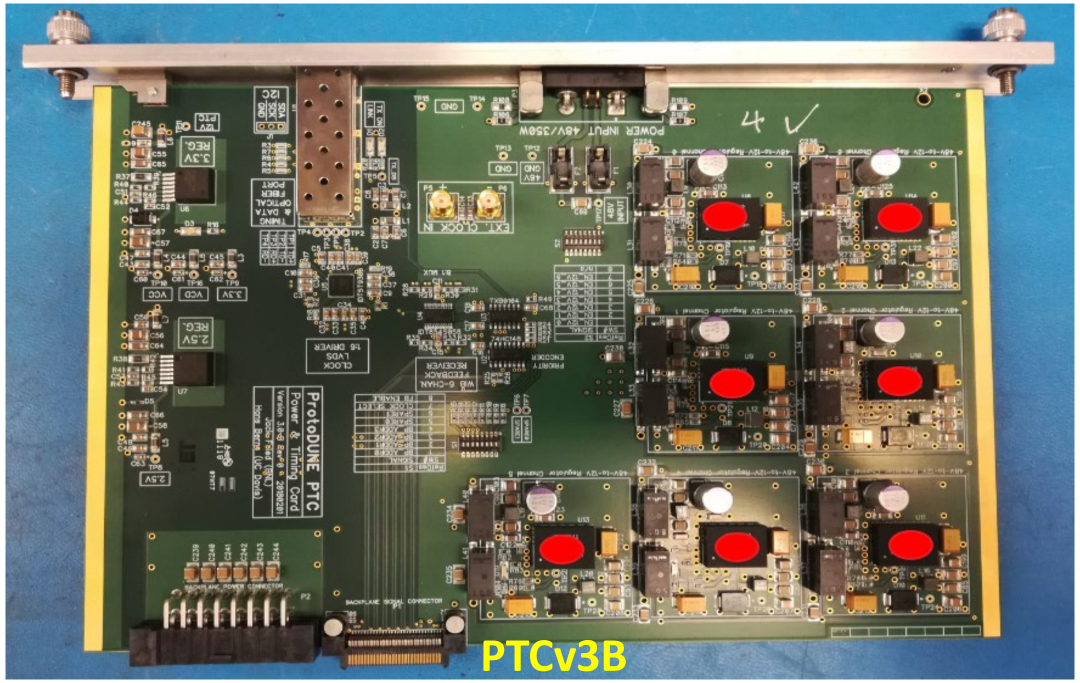
What is PTC?

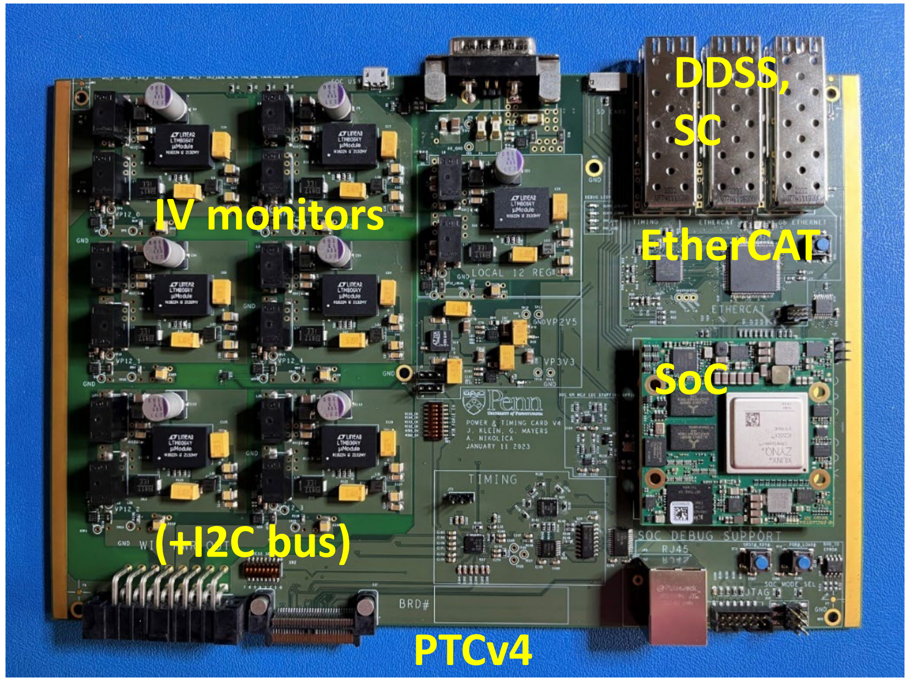
- Power and Timing Card
  - Provides Warm Interface Boards (WIBs) with 12V power on backplane (Power and Timing Backplane, or PTB)
  - Distributes DUNE timing master clock and data (62.5MHz) to WIB over PTB
  - Priority encodes WIB transmission back to timing master (one WIB at a time)



Introduction

- Reasons for re-designing the PTC:
  - Monitoring of local voltages and temperatures
  - Slow Control (SC) interface
  - DUNE Detector Safety System (DDSS) interface, or DUNE cold electronics interlock (CE Interlock)
  - Individual WIB control* and/or communications



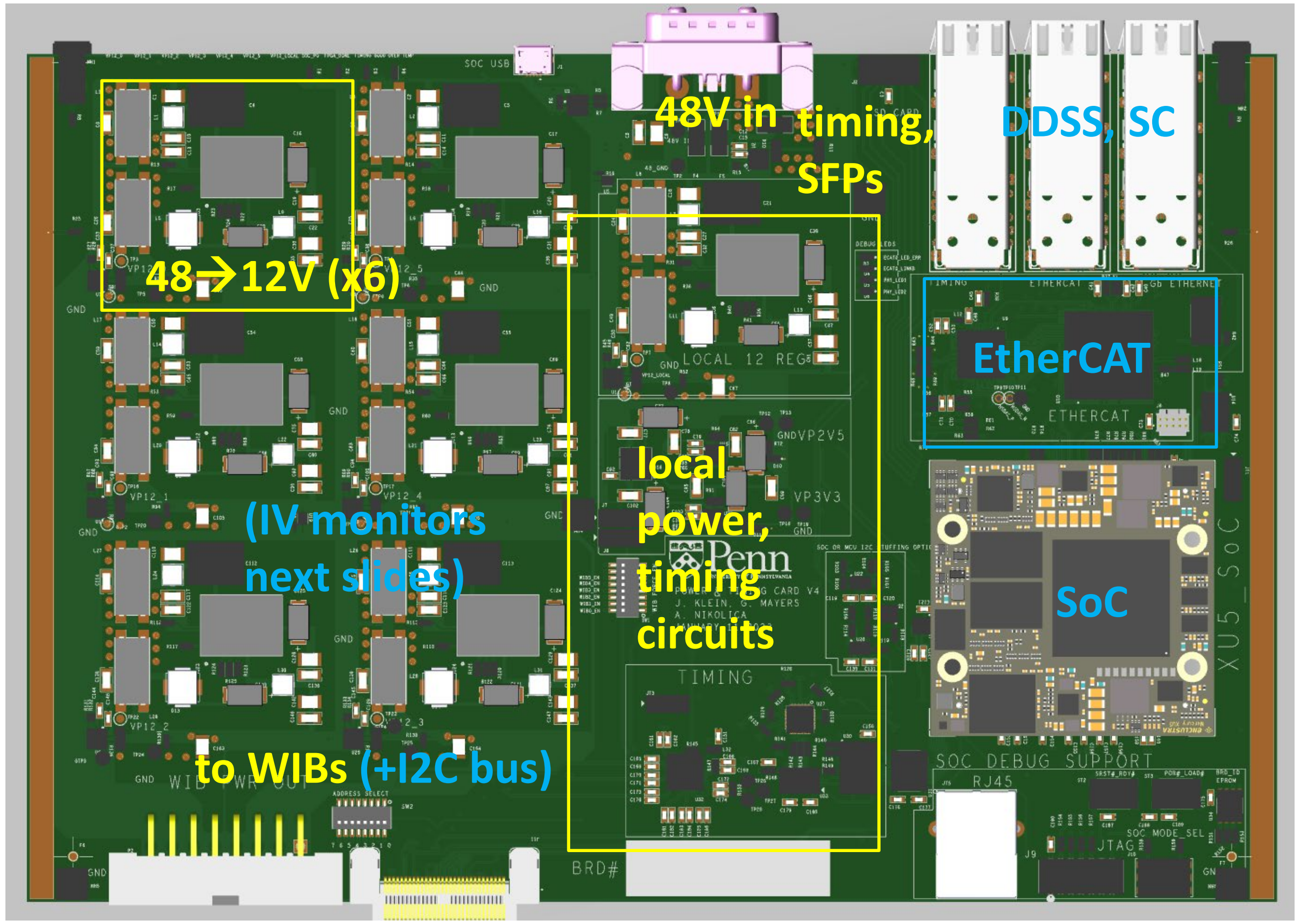


Documents:
https://edms.cern.ch/document/2893862/1
https://edms.cern.ch/document/2339398/2

**Existing WIBv3 not able to be powered down in a crate with PTCv4 without affecting some data lines – details later in this presentation*

PTCv4: external interfaces

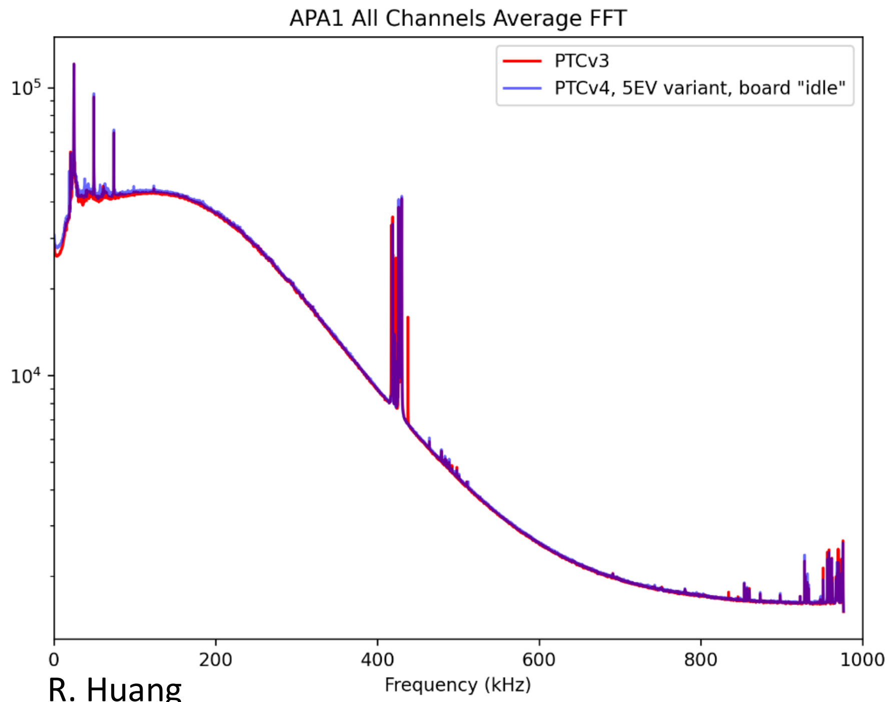
- 48V in
- Bristol timing system via 1000Base-BX SFP
- Slow Control via Ethernet over 1000Base-LX SFP
- DDSS via EtherCAT over 100Base-FX SFP
- WIB Interfaces
  - 12V out to WIB
  - Timing clock and data out to WIB
  - Timing transmit from WIB
  - I2C to/from WIB
- SoC functions:
  - Power sequencing and control
  - GbE to SC
  - UART communications to EtherCAT microcontroller
  - I2C power monitoring (local, and WIB)
  - Minimal timing endpoint interface



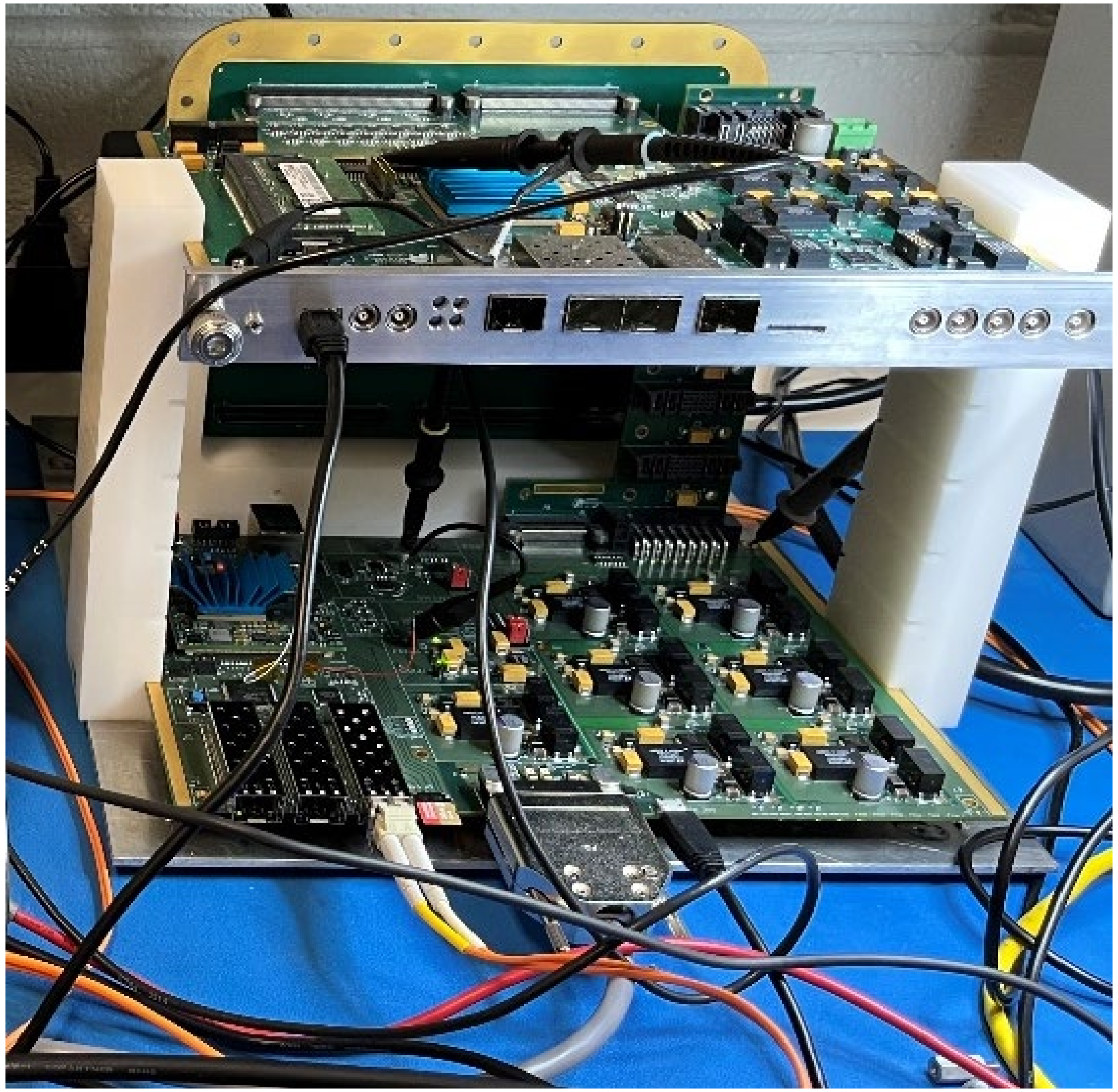
PCB image exported from Cadence (populated)
NOTE: **BLUE** = new feature

Powering a WIB

- Carefully checked new interfaces
  - Level translation on backplane addressing and timing priority encode
  - Power sequencing
- Powering works, expected current consumption measured
- “Noise” test conducted at CERN

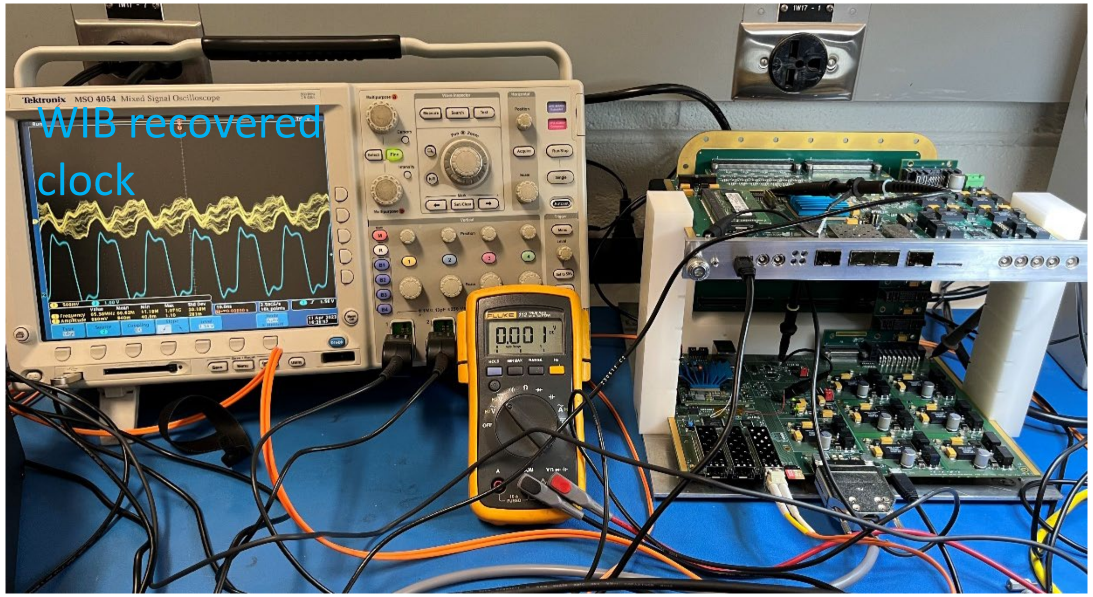


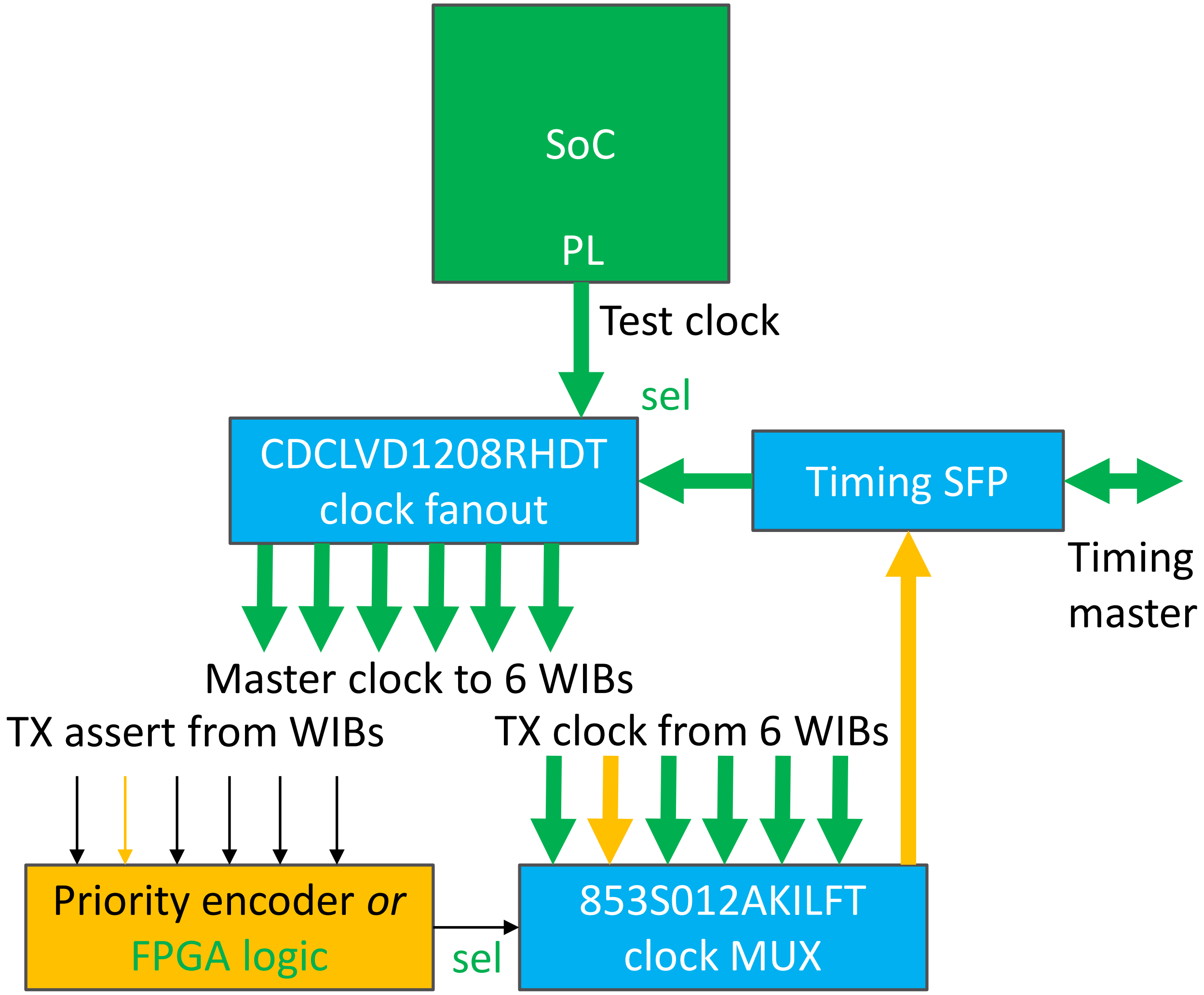
R. Huang



Timing tests

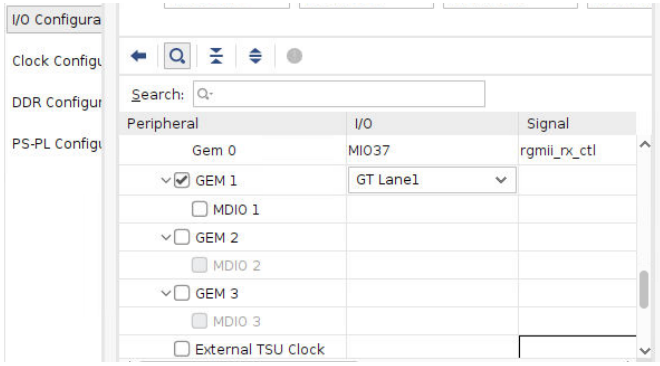
- PTC timing distribution:
  - Have the ability to generate a test clock
  - Or use the fiber connection from timing master for DCSK stream
- Transmission back to timing master:
  - Timing information is only passed through in hardware, and hardware priority encoder used to MUX
  - Can also use FPGA-controlled MUX



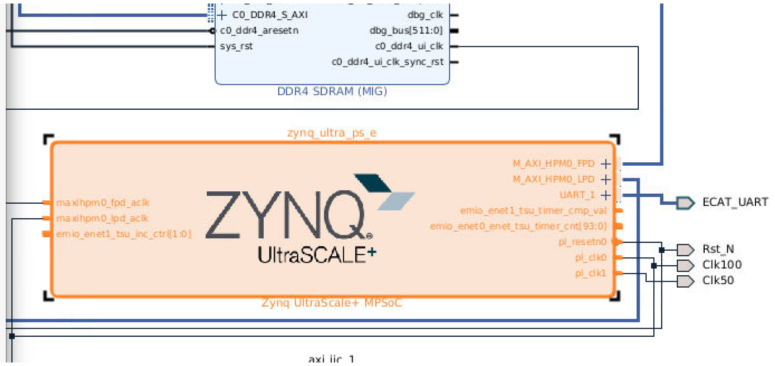


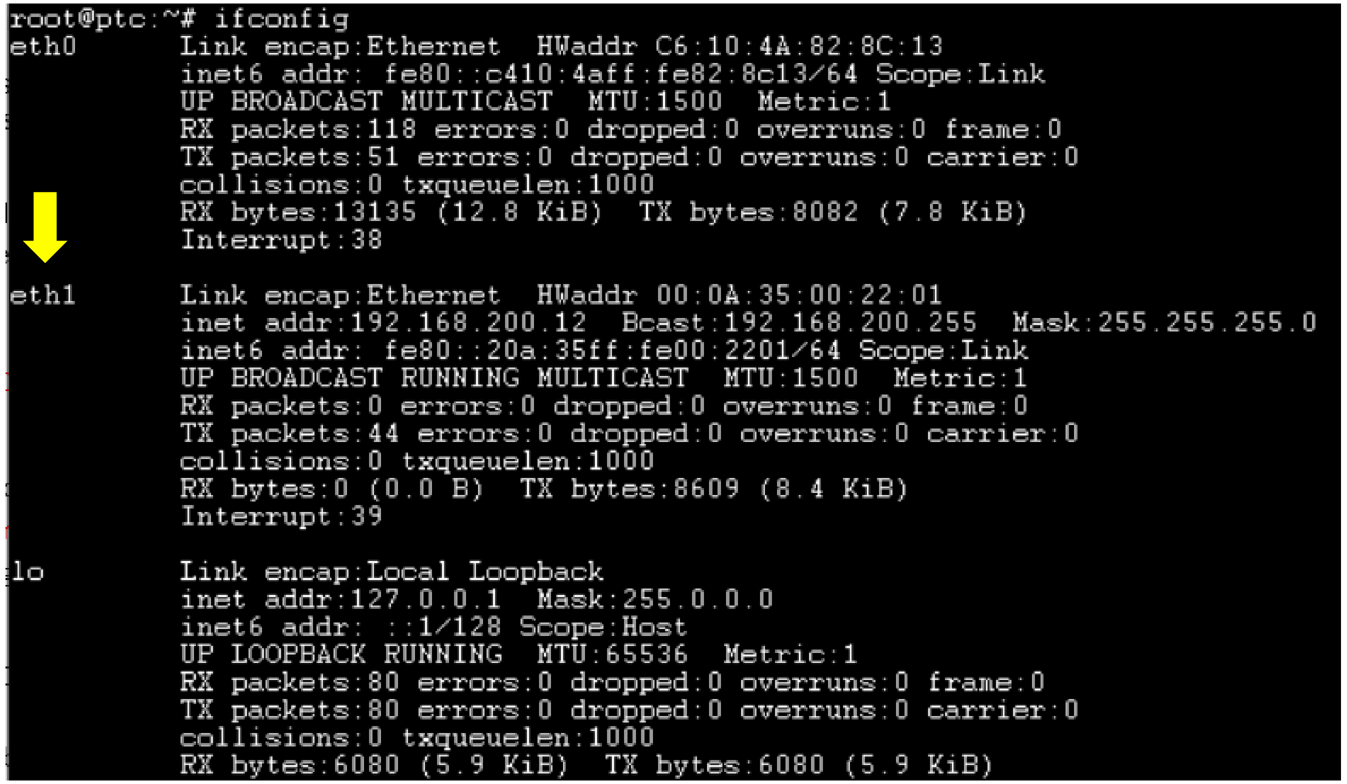
GbE to SC

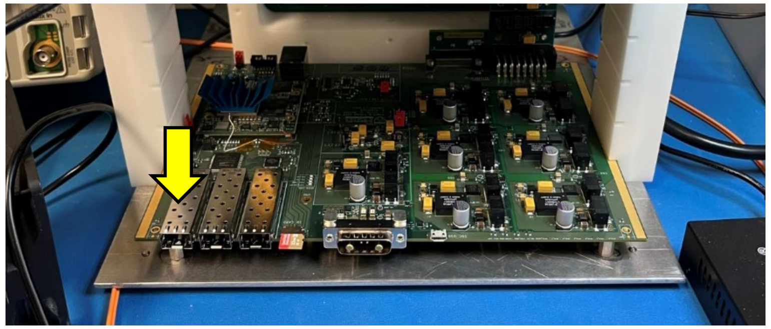
- SC GbE is on front panel SFP
- Configured in Zynq settings
  - PS transceivers
  - Same HW/config as WIB
  - Verified reference clock is okay
  - Need to write one Zynq register config to bring up interface (same as WIB)
- Using 10GTek A7S2-33-1GX1GT-SFP/GT3 fiber-to-copper converter
  - Same exact HW as WIB test stand



Peripheral	I/O	Signal
Gem 0	MIO37	rgmii_rx_ctl
☑ GEM 1	GT Lane1	
☐ MDIO 1		
☐ GEM 2		
☐ MDIO 2		
☐ GEM 3		
☐ MDIO 3		
☐ External TSU Clock		

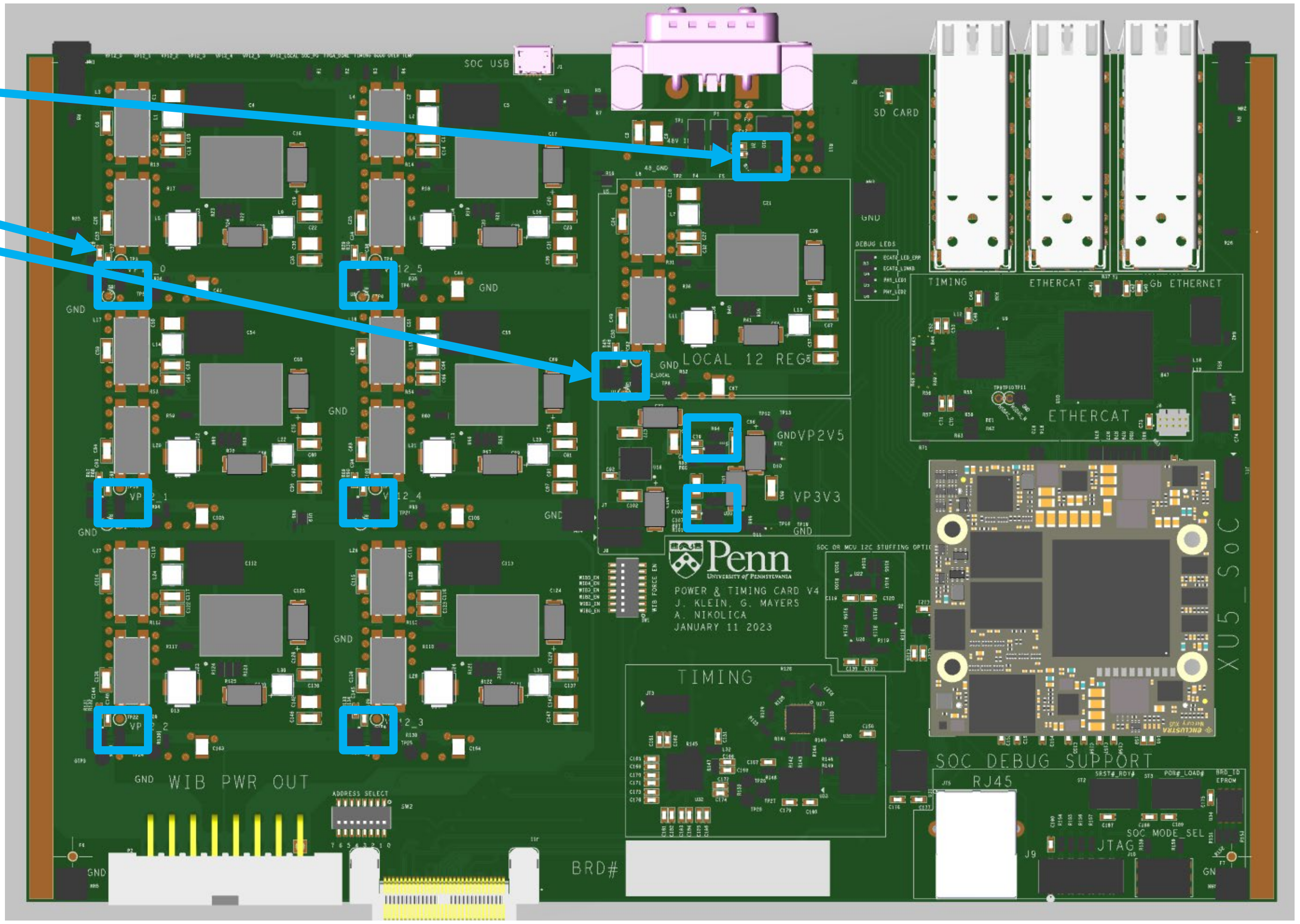



```
root@ptc:~# ifconfig
eth0      Link encap:Ethernet  HWaddr C6:10:4A:82:8C:13
          inet6 addr: fe80::c410:4aff:fe82:8c13/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:118 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13135 (12.8 KiB)  TX bytes:8082 (7.8 KiB)
          Interrupt:38

eth1      Link encap:Ethernet  HWaddr 00:0A:35:00:22:01
          inet addr:192.168.200.12  Bcast:192.168.200.255  Mask:255.255.255.0
          inet6 addr: fe80::20a:35ff:fe00:2201/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:8609 (8.4 KiB)
          Interrupt:39

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:80 errors:0 dropped:0 overruns:0 frame:0
          TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6080 (5.9 KiB)  TX bytes:6080 (5.9 KiB)
```



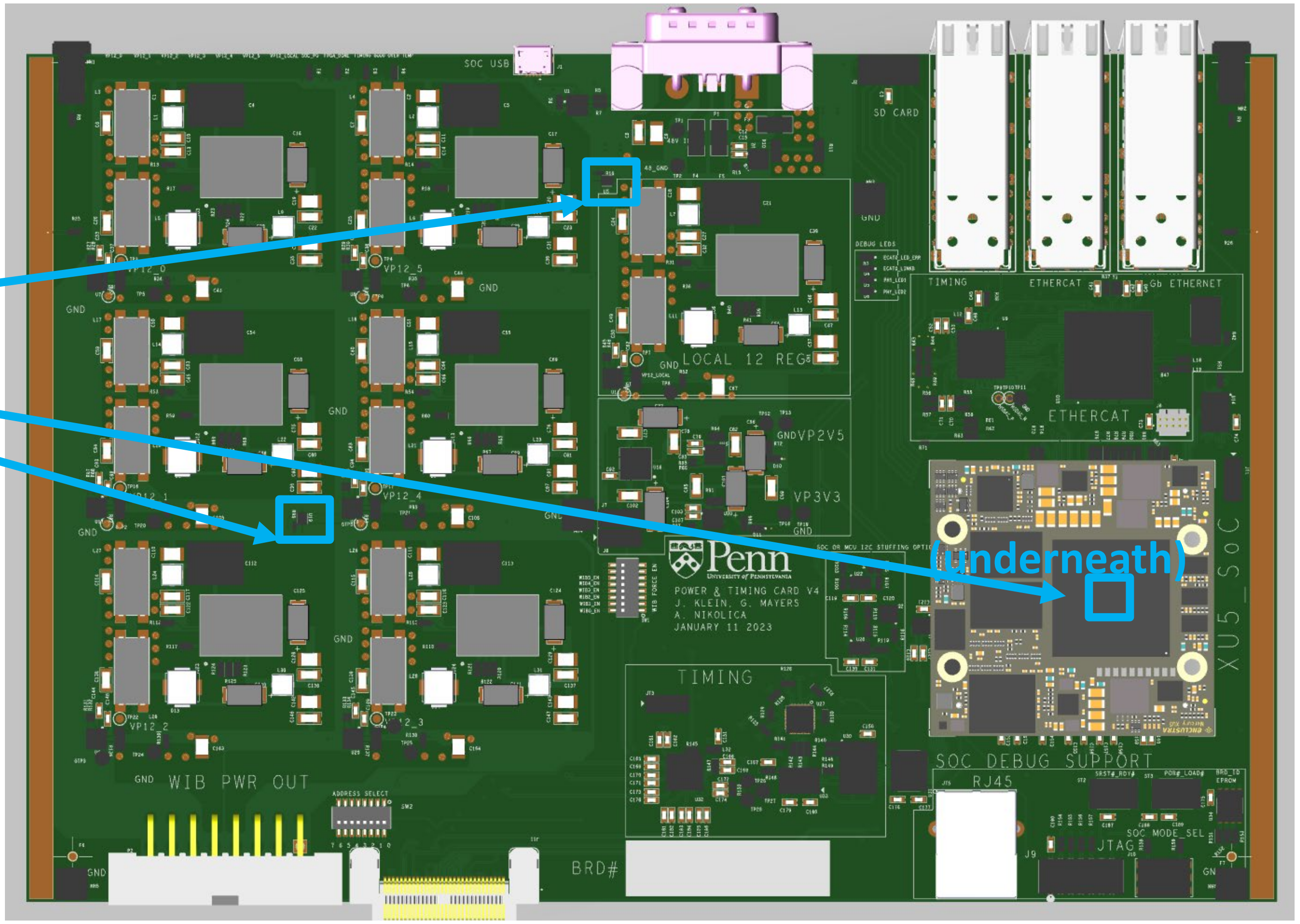
Monitored quantities

- Voltages and currents (using LTC2945):
  - 48V input
  - All 6x WIB 12V rails
  - Local 12V (3.3V and 2.5V optional)



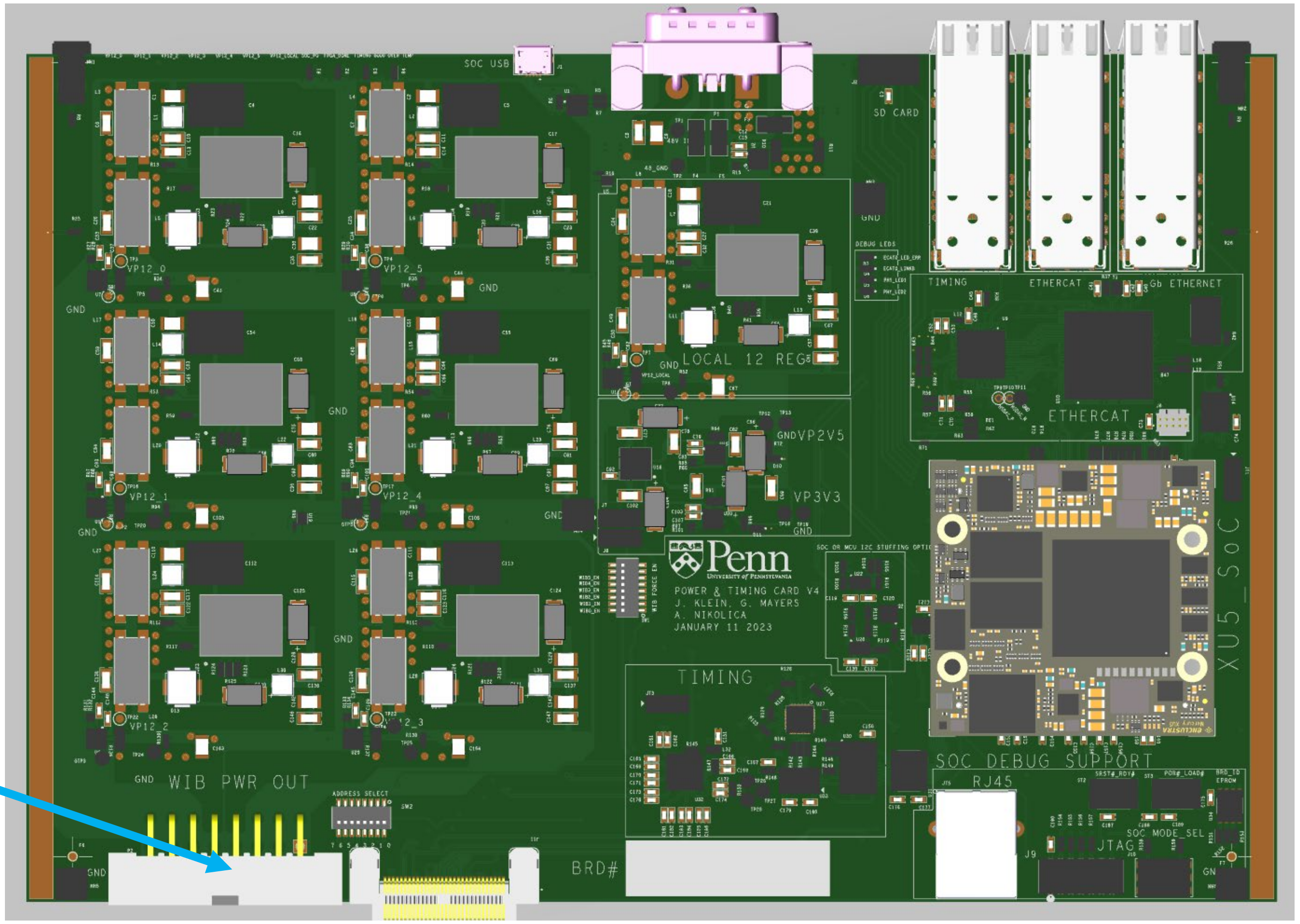
Monitored quantities

- Voltages and currents (using LTC2945):
  - 48V input
  - All 6x WIB 12V rails
  - Local 12V (3.3V and 2.5V optional)
- Temperatures (using TMP117)
  - 3x locations on board
  - SoC can monitor its own FPGA internal temperature



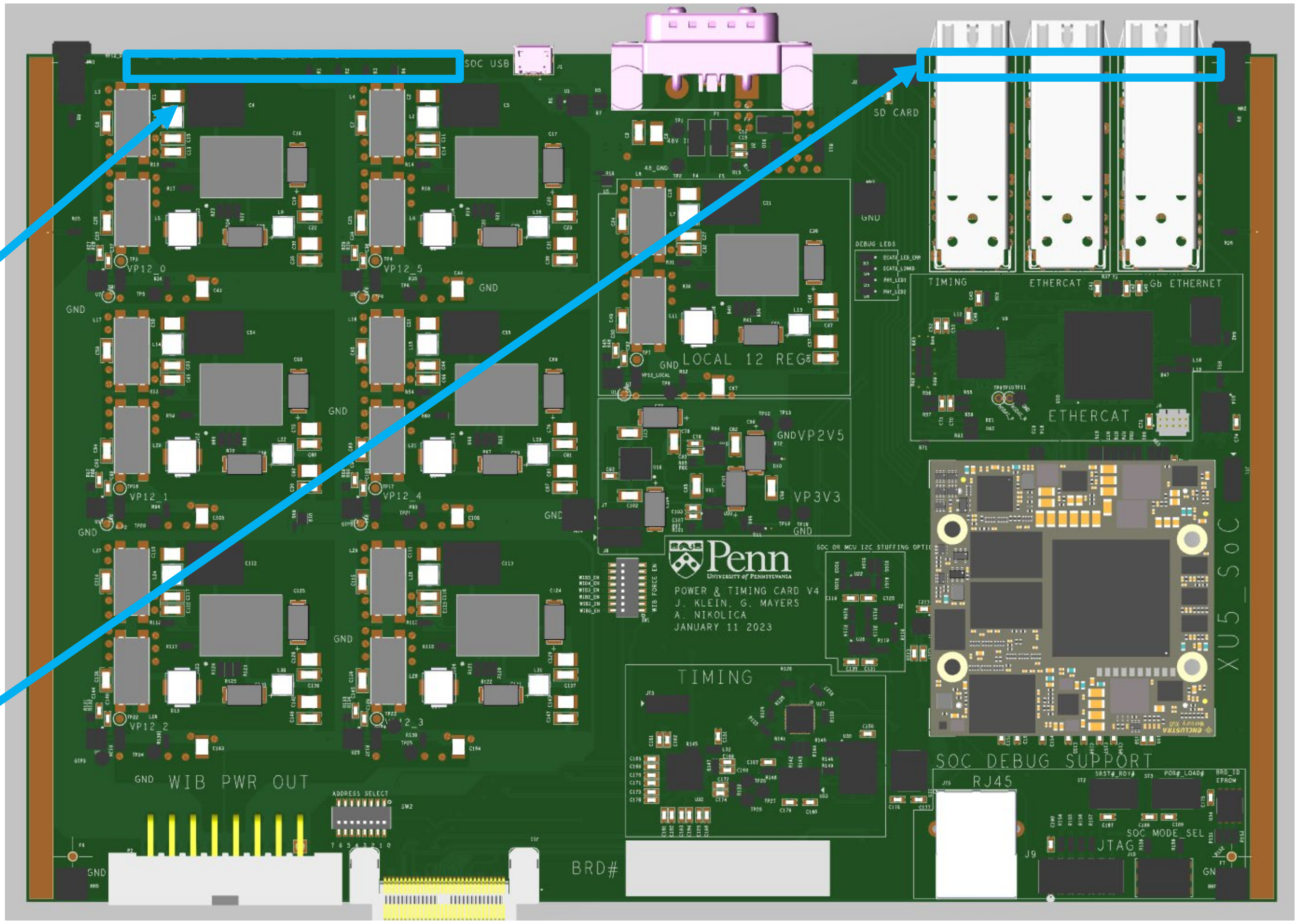
Monitored quantities

- Voltages and currents (using LTC2945):
  - 48V input
  - All 6x WIB 12V rails
  - Local 12V (3.3V and 2.5V optional)
- Temperatures (using TMP117)
  - 3x locations on board
  - SoC can monitor its own FPGA internal temperature
- WIB I2C



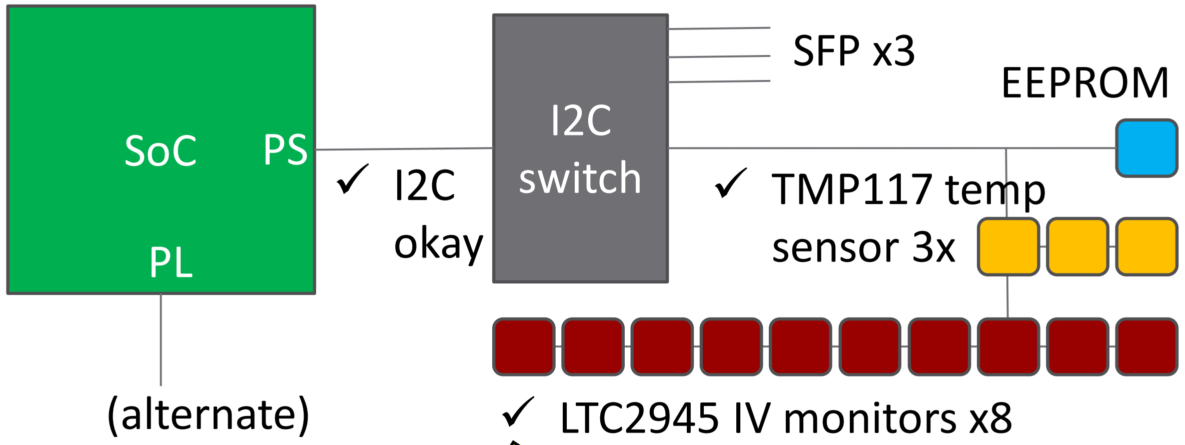
Monitored quantities

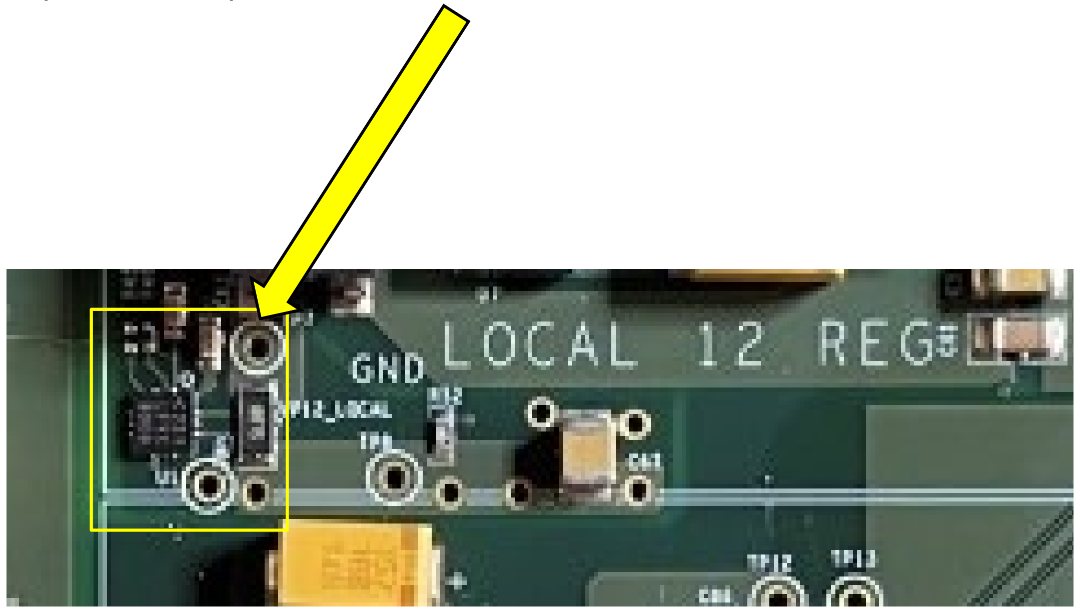
- Voltages and currents (using LTC2945):
  - 48V input
  - All 6x WIB 12V rails
  - Local 12V (3.3V and 2.5V optional)
- Temperatures (using TMP117)
  - 3x locations on board
  - SoC can monitor its own FPGA internal temperature
- WIB I2C
- LEDs
  - 6x WIB power indicators
  - Over temperature (FPGA programmable)
  - SoC: power good, and FPGA done
  - Timing signal okay (decoded by FPGA)
  - SFP LEDs:
    - 3x loss of signal (LOS)
    - Timing: WIB transmit back indicator
    - DDSS: EtherCAT link active
  - (Other debug LEDs for bench only)



I2C tests to PTC sensors

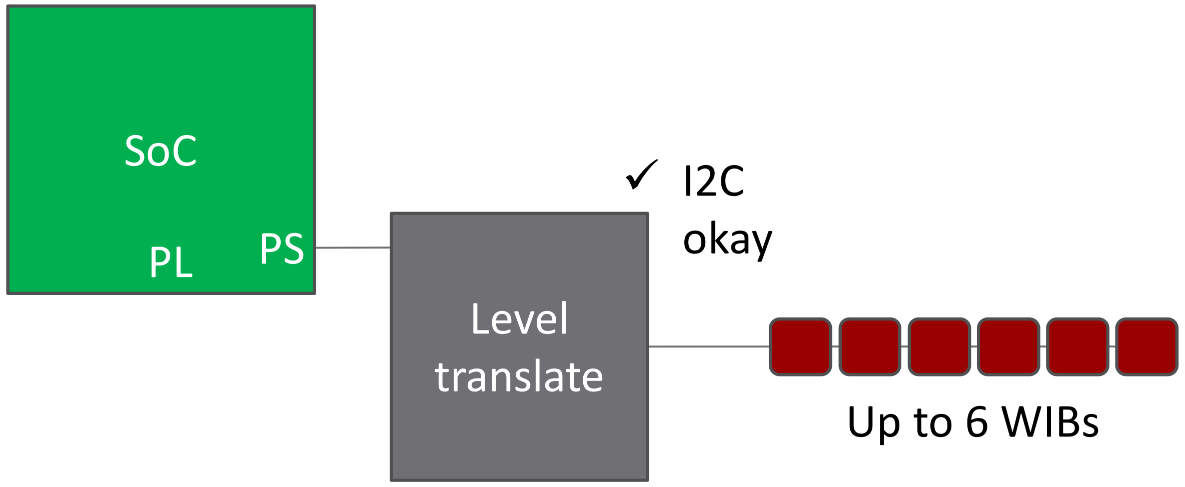
- I2C from Zynq PS through I2C switch to temperature sensors works

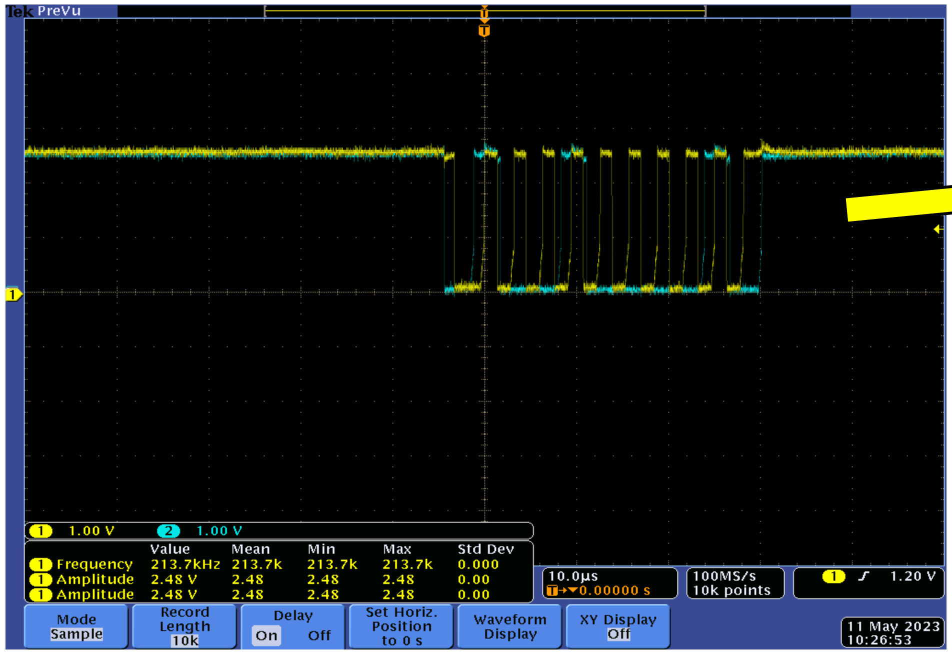




I2C tests to WIB

- I2C from Zynq PS through level translator to WIB seems to work
  - Need to test all possible sensors

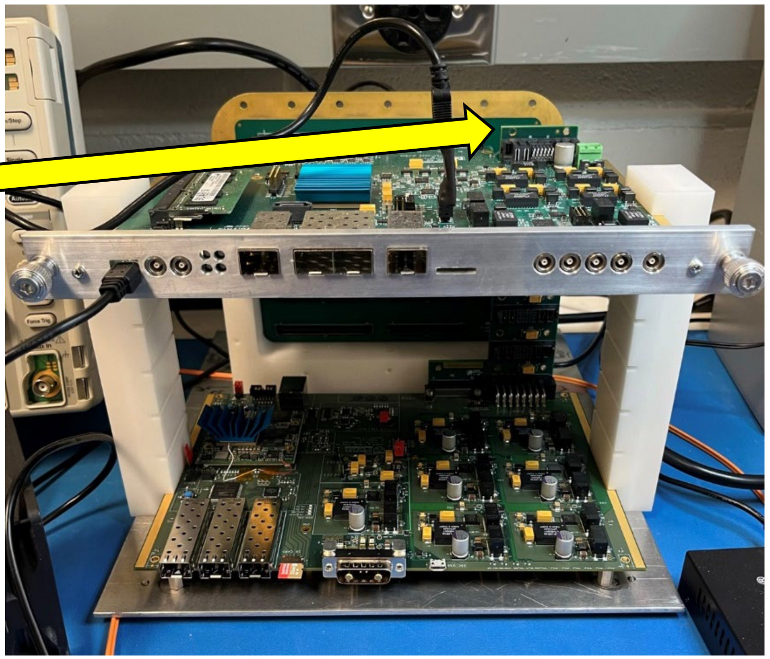




1 1.00 V 2 1.00 V

	Value	Mean	Min	Max	Std Dev
1 Frequency	213.7kHz	213.7k	213.7k	213.7k	0.000
1 Amplitude	2.48 V	2.48	2.48	2.48	0.00
1 Amplitude	2.48 V	2.48	2.48	2.48	0.00

10.0µs
T→▼0.00000 s
100MS/s
10k points
1 ʃ 1.20 V
11 May 2023
10:26:53

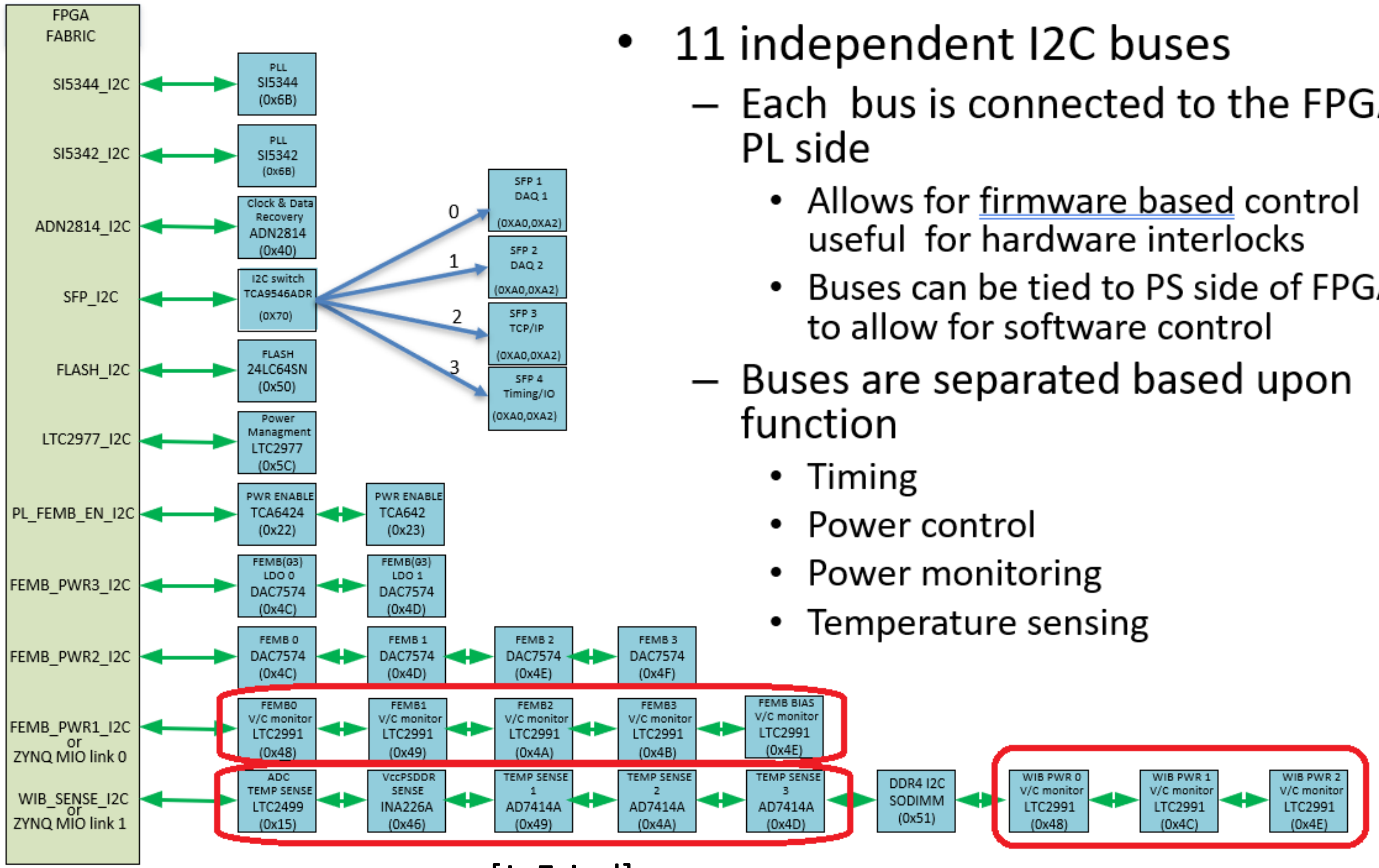


WIB monitored quantities

BROOKHAVEN NATIONAL LABORATORY

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

WIB I2C MAP


FPGA FABRIC
SI5344_I2C
PLL SI5344 (0x6B)
SI5342_I2C
PLL SI5342 (0x6B)
ADN2814_I2C
Clock & Data Recovery ADN2814 (0x40)
SFP_I2C
I2C switch TCA9546ADR (0X70)
0
1
2
3
SFP 1 DAQ 1 (0XA0,0XA2)
SFP 2 DAQ 2 (0XA0,0XA2)
SFP 3 TCP/IP (0XA0,0XA2)
SFP 4 Timing/IO (0XA0,0XA2)
FLASH_I2C
FLASH 24LC64SN (0x50)
LTC2977_I2C
Power Managment LTC2977 (0x5C)
PL_FEMB_EN_I2C
PWR ENABLE TCA6424 (0x22)
PWR ENABLE TCA642 (0x23)
FEMB_PWR3_I2C
FEMB(03) LDO 0 DAC7574 (0x4C)
FEMB(03) LDO 1 DAC7574 (0x4D)
FEMB_PWR2_I2C
FEMB 0 DAC7574 (0x4C)
FEMB 1 DAC7574 (0x4D)
FEMB 2 DAC7574 (0x4E)
FEMB 3 DAC7574 (0x4F)
FEMB_PWR1_I2C or ZYNQ MIO link 0
FEMB0 V/C monitor LTC2991 (0x48)
FEMB1 V/C monitor LTC2991 (0x49)
FEMB2 V/C monitor LTC2991 (0x4A)
FEMB3 V/C monitor LTC2991 (0x4B)
FEMB BIAS V/C monitor LTC2991 (0x4E)
WIB_SENSE_I2C or ZYNQ MIO link 1
ADC TEMP SENSE LTC2499 (0x15)
VccPSDDR SENSE INA226A (0x46)
TEMP SENSE 1 AD7414A (0x49)
TEMP SENSE 2 AD7414A (0x4A)
TEMP SENSE 3 AD7414A (0x4D)
DDR4 I2C SODIMM (0x51)
WIB PWR 0 V/C monitor LTC2991 (0x48)
WIB PWR 1 V/C monitor LTC2991 (0x4C)
WIB PWR 2 V/C monitor LTC2991 (0x4E)

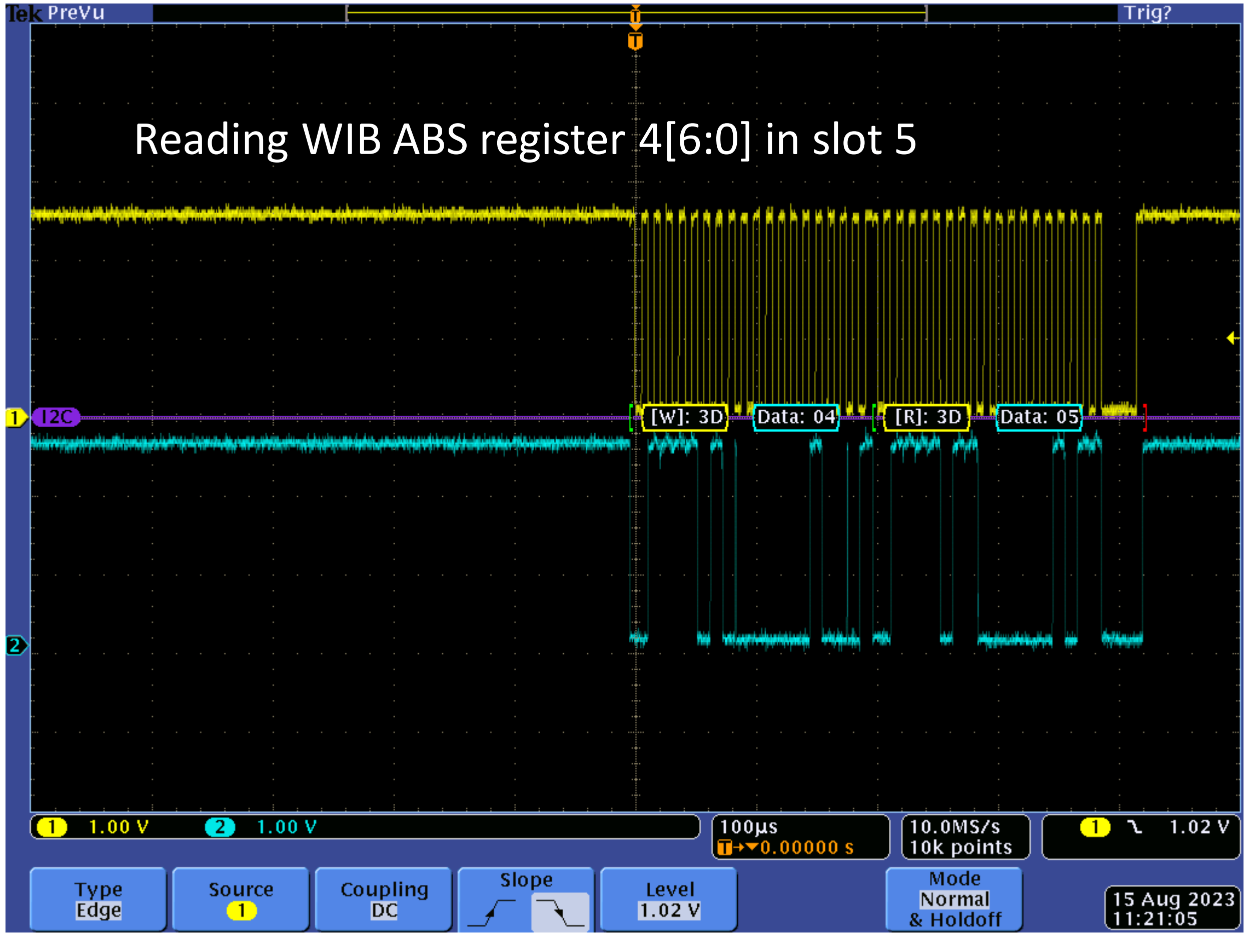
- 11 independent I2C buses
  - Each bus is connected to the FPGA PL side
    - Allows for firmware based control useful for hardware interlocks
    - Buses can be tied to PS side of FPGA to allow for software control
  - Buses are separated based upon function
    - Timing
    - Power control
    - Power monitoring
    - Temperature sensing

Commands on WIB to enable interface:

`poke 0xff5e00c4 0x1033200 //` to enable 10MHz clock
`peek 0xa00c008c //` reads `0xd`; we only care about bits [2:0], or `0x5`
WIB address is `0x38` + `0x5` = `0x3d`

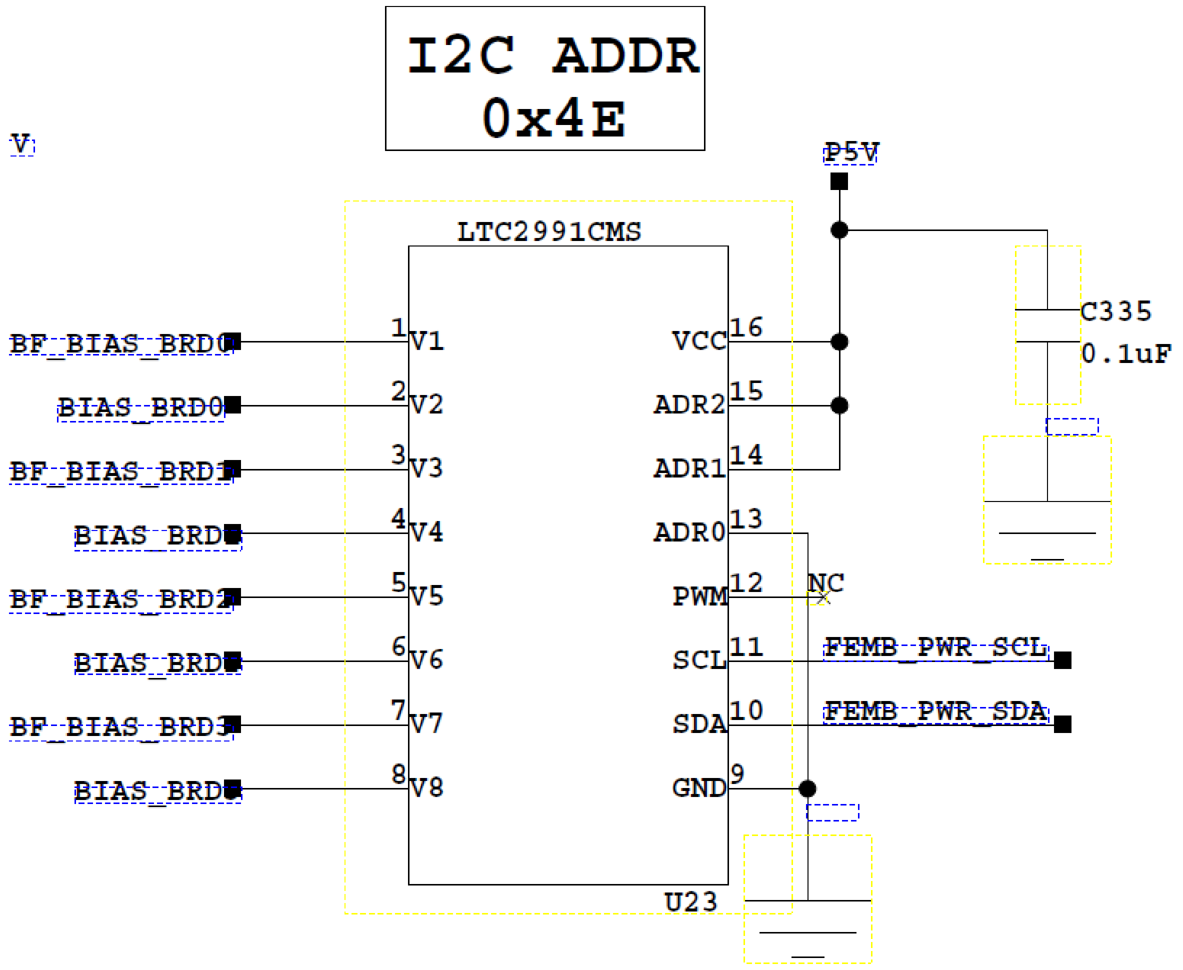
Commands on PTC to detect WIB:

```
root@ptc:~# i2cdetect -y -r 2
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- 3d -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```


Reading WIB ABS register 4[6:0] in slot 5

Commands on PTC to read a sensor on FEMB_PWR bus:

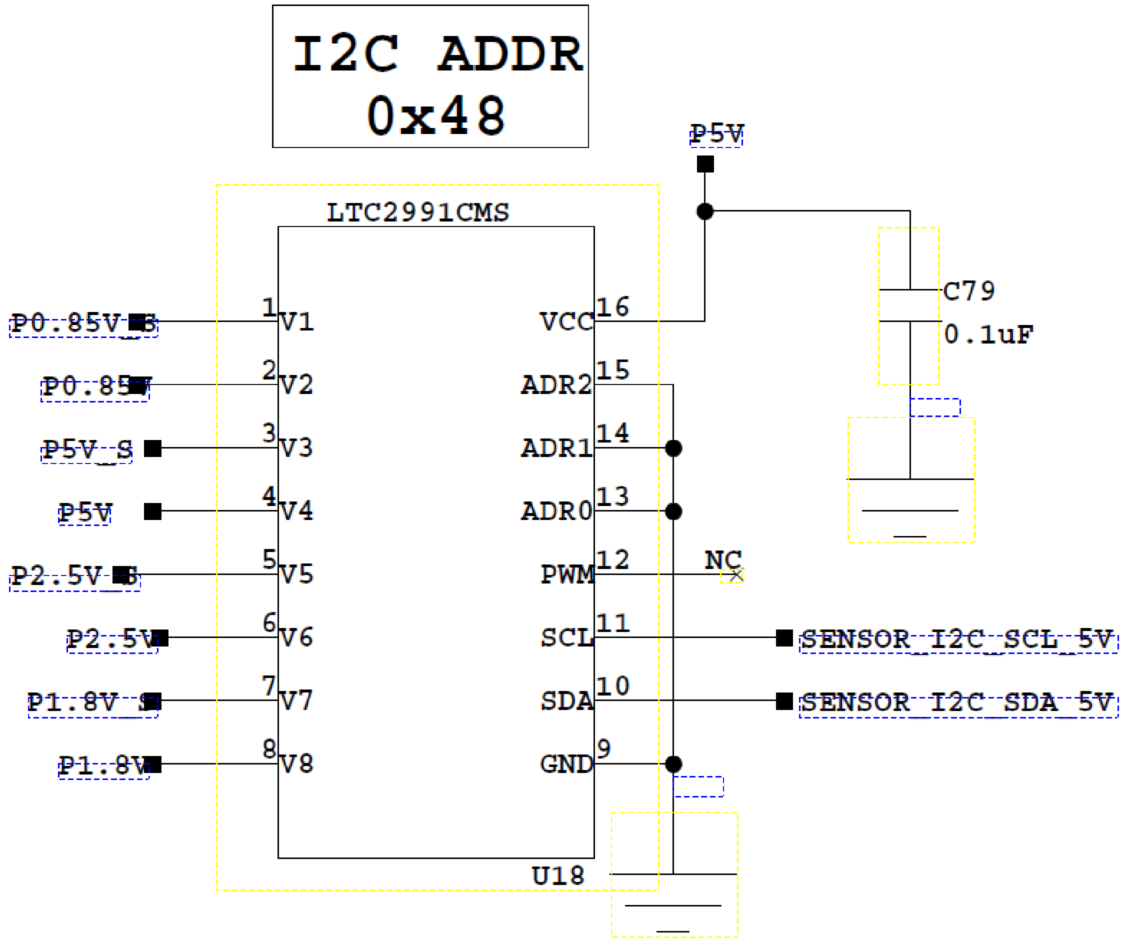
```
root@ptc:~# i2cset -y 2 0x3d 0x00 0x01 // to enable

root@ptc:~# i2cdetect -y -r 2
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- 22 23 -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- 3d -- --
40: -- -- -- -- -- -- -- -- 48 49 4a 4b -- -- 4e --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --

root@ptc:~# i2cset -y 2 0x4e 0x01 0x1f // to enable ch, read
root@ptc:~# i2cget -y 2 0x4e 0x1d // read twice to clear DV
0xec
root@ptc:~# i2cget -y 2 0x4e 0x1c
0x9f
root@ptc:~# i2cget -y 2 0x4e 0x1d // read twice to clear DV
0xec
root@ptc:~# i2cget -y 2 0x4e 0x1c
0x1f
```



According to LTC2991 datasheet:
`0x1fec` is Vcc measurement
d8172 * 305.18uV + 2.5V = 4.99V, okay

Commands on PTC to read a sensor on SENSOR_I2C bus:

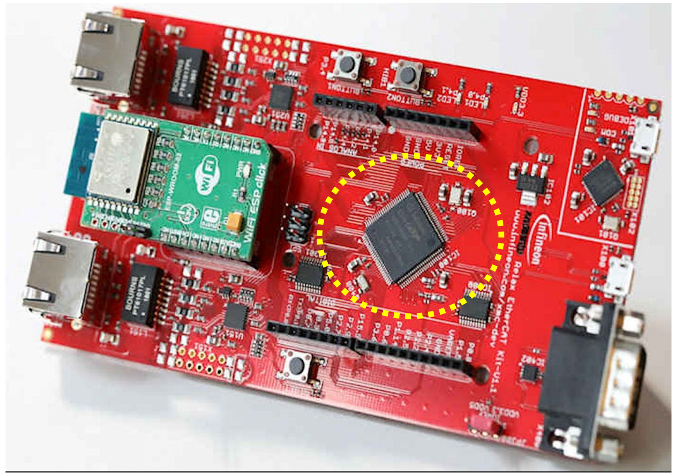
```
root@ptc:~# i2cset -y 2 0x3d 0x00 0x03 // to switch bus

root@ptc:~# i2cdetect -y -r 2
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- 15 -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: 30 -- -- -- -- -- 36 -- -- -- -- -- -- 3d -- --
40: -- -- -- -- -- -- 46 -- 48 49 4a -- 4c 4d 4e --
50: -- 51 -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --

root@ptc:~# i2cset -y 2 0x48 0x01 0x1f // to enable ch, read
root@ptc:~# i2cget -y 2 0x48 0x0a // read twice to clear DV
0x8b
root@ptc:~# i2cget -y 2 0x48 0x0b
0x05
root@ptc:~# i2cget -y 2 0x48 0x0a // read twice to clear DV
0x0b
root@ptc:~# i2cget -y 2 0x48 0x0b
0x05
```

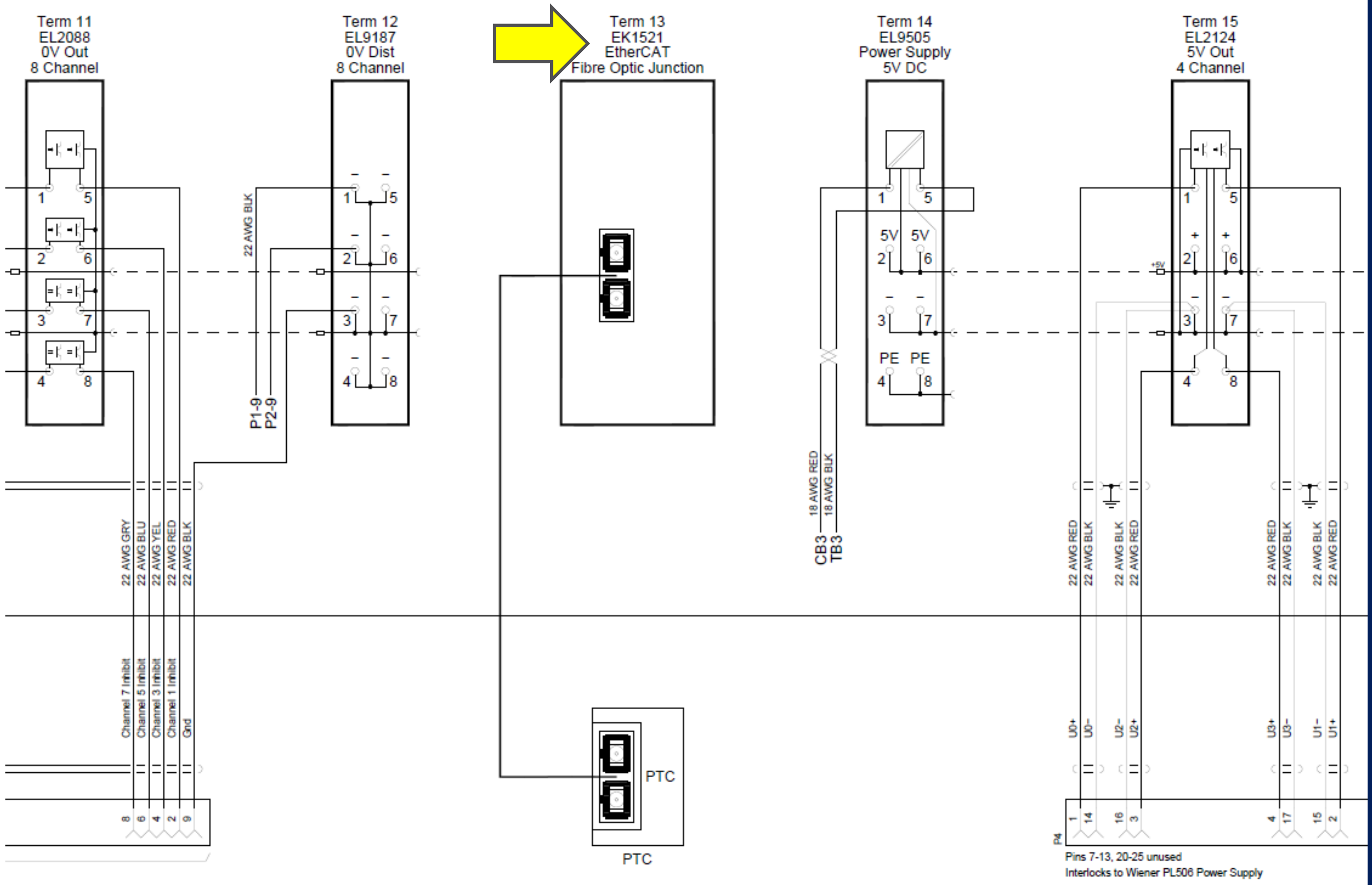


According to LTC2991 datasheet:
`0x0b05` is V1 measurement
d2821 * 305.18uV = 0.86V, okay

Design: EtherCAT

- EtherCAT interface is required for DDSS connection (already designed)
- Implemented with an Infineon XMC4300 EtherCAT-capable microcontroller
    - Beckhoff firmware solution was prohibitively expensive

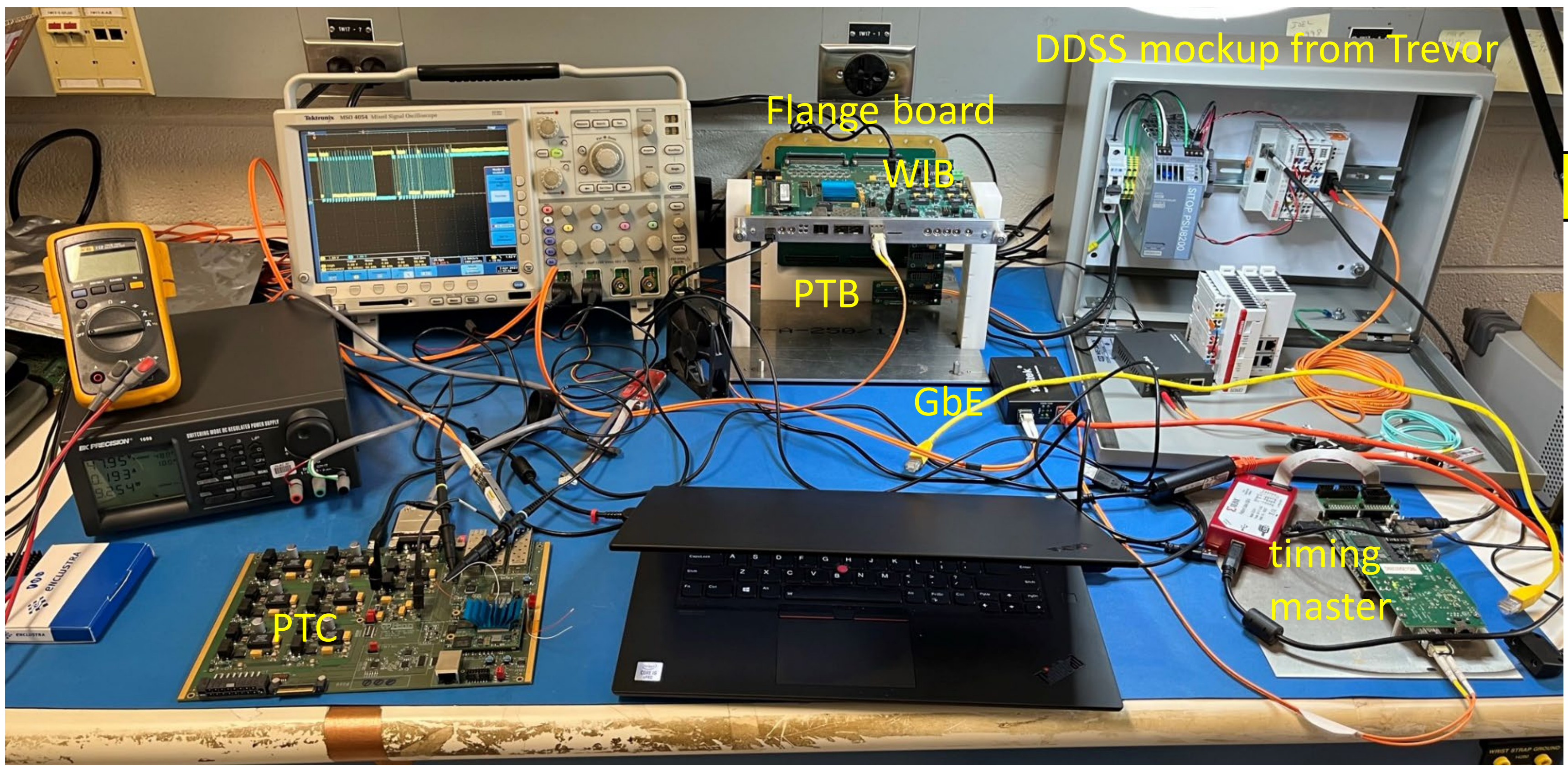


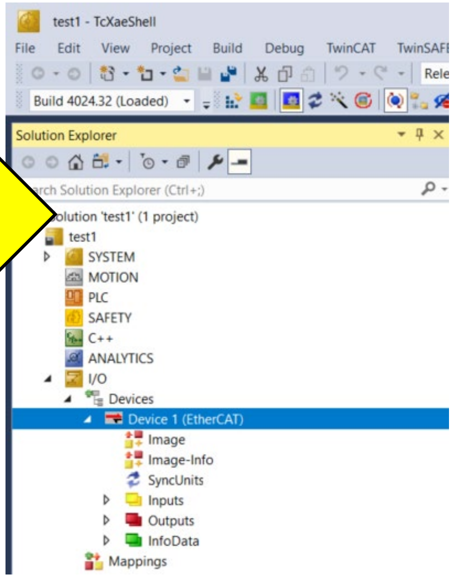


Term 11 EL2088 0V Out 8 Channel
Term 12 EL9187 0V Dist 8 Channel
Term 13 EK1521 EtherCAT Fibre Optic Junction
Term 14 EL9505 Power Supply 5V DC
Term 15 EL2124 5V Out 4 Channel
22 AWG BLK
P1-9
P2-9
18 AWG RED
18 AWG BLK
CB3
TB3
22 AWG GRY
22 AWG BLU
22 AWG YEL
22 AWG RED
22 AWG BLK
Channel 7 Inhibit
Channel 5 Inhibit
Channel 3 Inhibit
Channel 1 Inhibit
Gnd
PTC
Pins 7-13, 20-25 unused
Interlocks to Wiener PL506 Power Supply

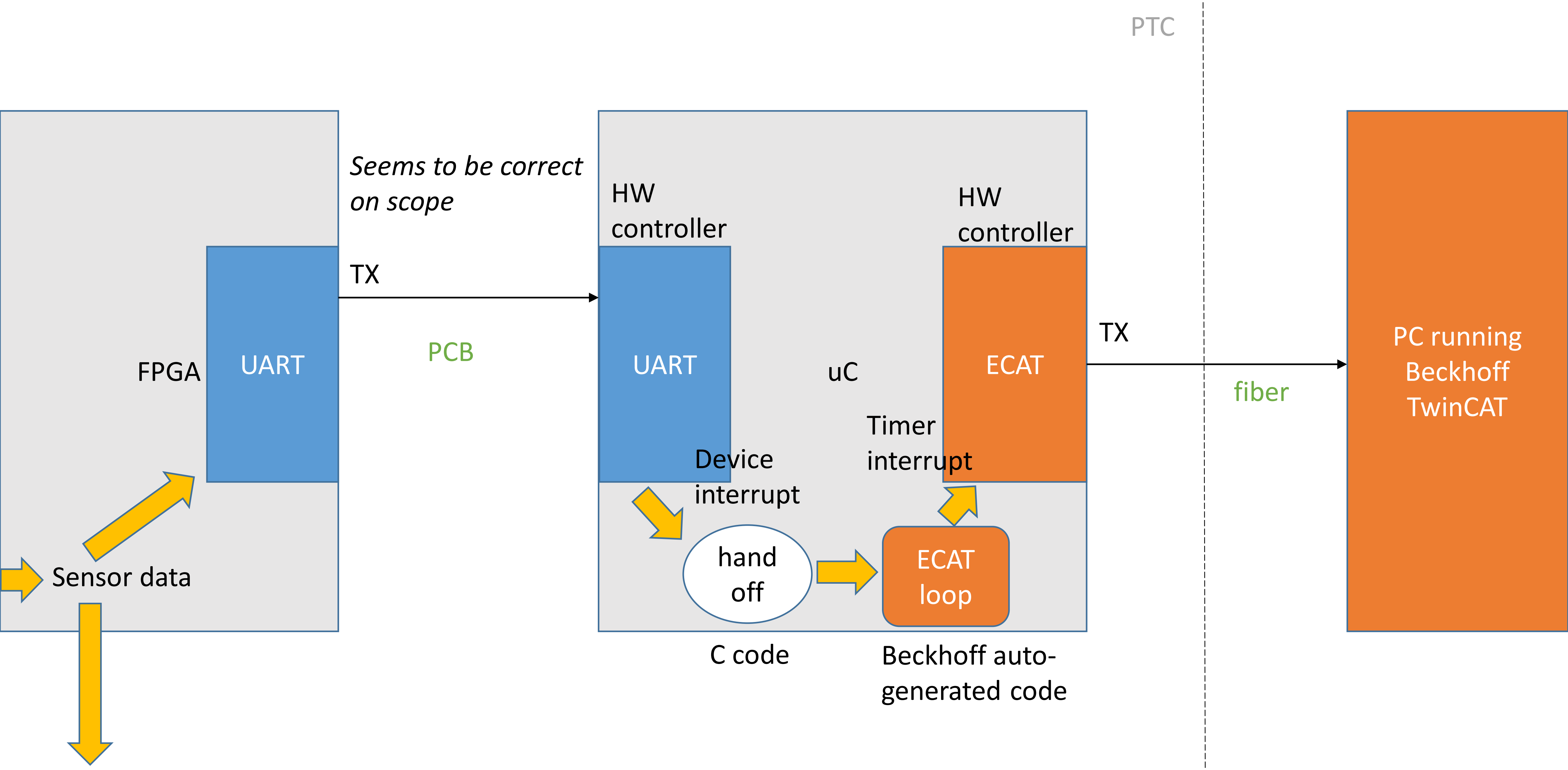
(Section of DDSS design -- link in backup slides)

Penn test stand

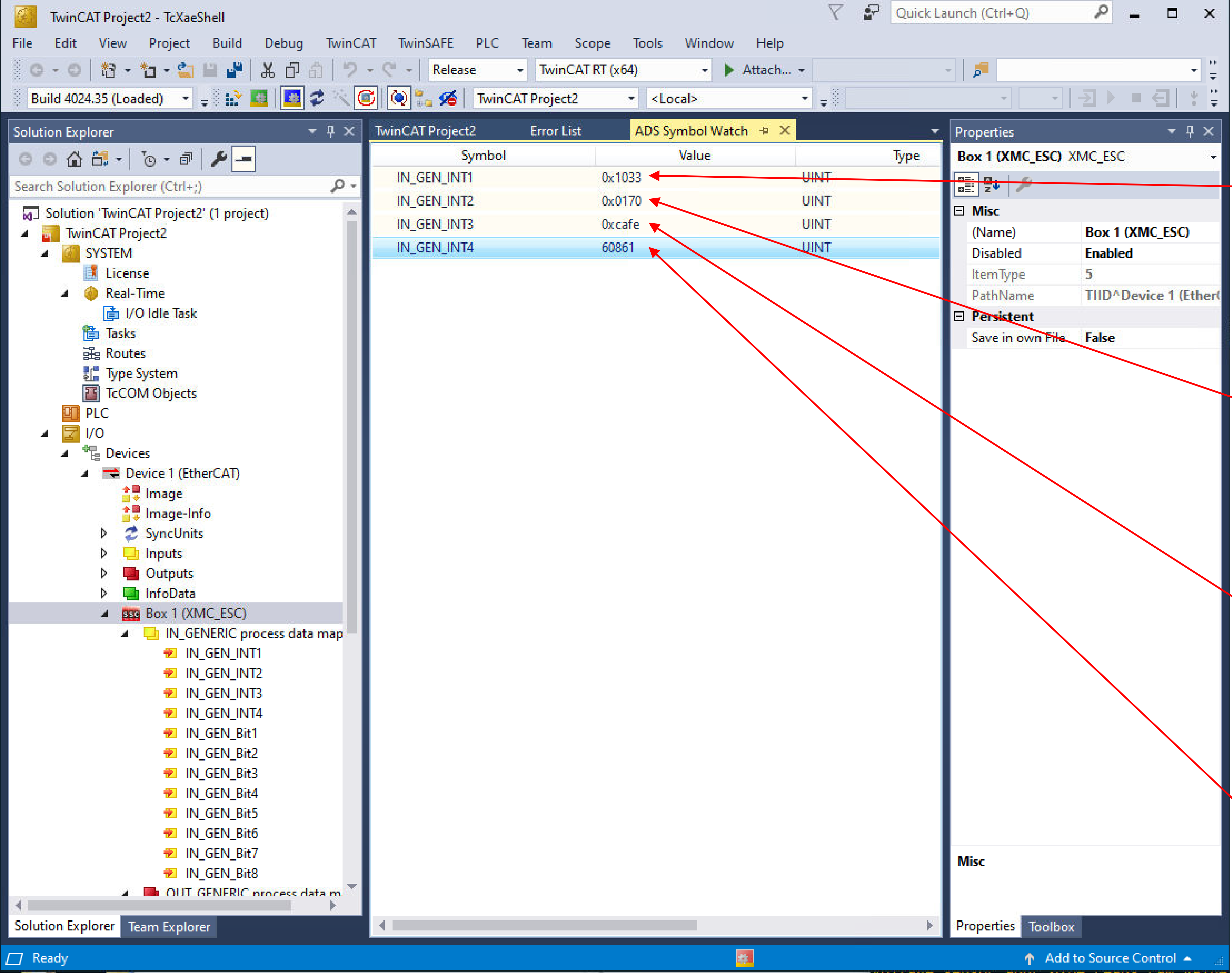




Beckhoff TwinCAT EtherCAT software master



- ✓ *Always as expected in log file*



- Solution 'TwinCAT Project2' (1 project)
  - TwinCAT Project2
    - SYSTEM
      - License
      - Real-Time
        - I/O Idle Task
      - Tasks
      - Routes
      - Type System
      - TcCOM Objects
    - PLC
    - I/O
      - Devices
        - Device 1 (EtherCAT)
          - Image
          - Image-Info
          - SyncUnits
          - Inputs
          - Outputs
          - InfoData
          - Box 1 (XMC_ESC)
            - IN_GENERIC process data map
              - IN_GEN_INT1
              - IN_GEN_INT2
              - IN_GEN_INT3
              - IN_GEN_INT4
              - IN_GEN_Bit1
              - IN_GEN_Bit2
              - IN_GEN_Bit3
              - IN_GEN_Bit4
              - IN_GEN_Bit5
              - IN_GEN_Bit6
              - IN_GEN_Bit7
              - IN_GEN_Bit8
            - OUT GENERIC process data m

Symbol	Value	Type
IN_GEN_INT1	0x1033	UINT
IN_GEN_INT2	0x0170	UINT
IN_GEN_INT3	0xcafe	UINT
IN_GEN_INT4	60861	UINT

Box 1 (XMC_ESC) XMC_ESC

Misc	
(Name)	Box 1 (XMC_ESC)
Disabled	Enabled
ItemType	5
PathName	TIID^Device 1 (Ether
Persistent	
Save in own File	False

Misc

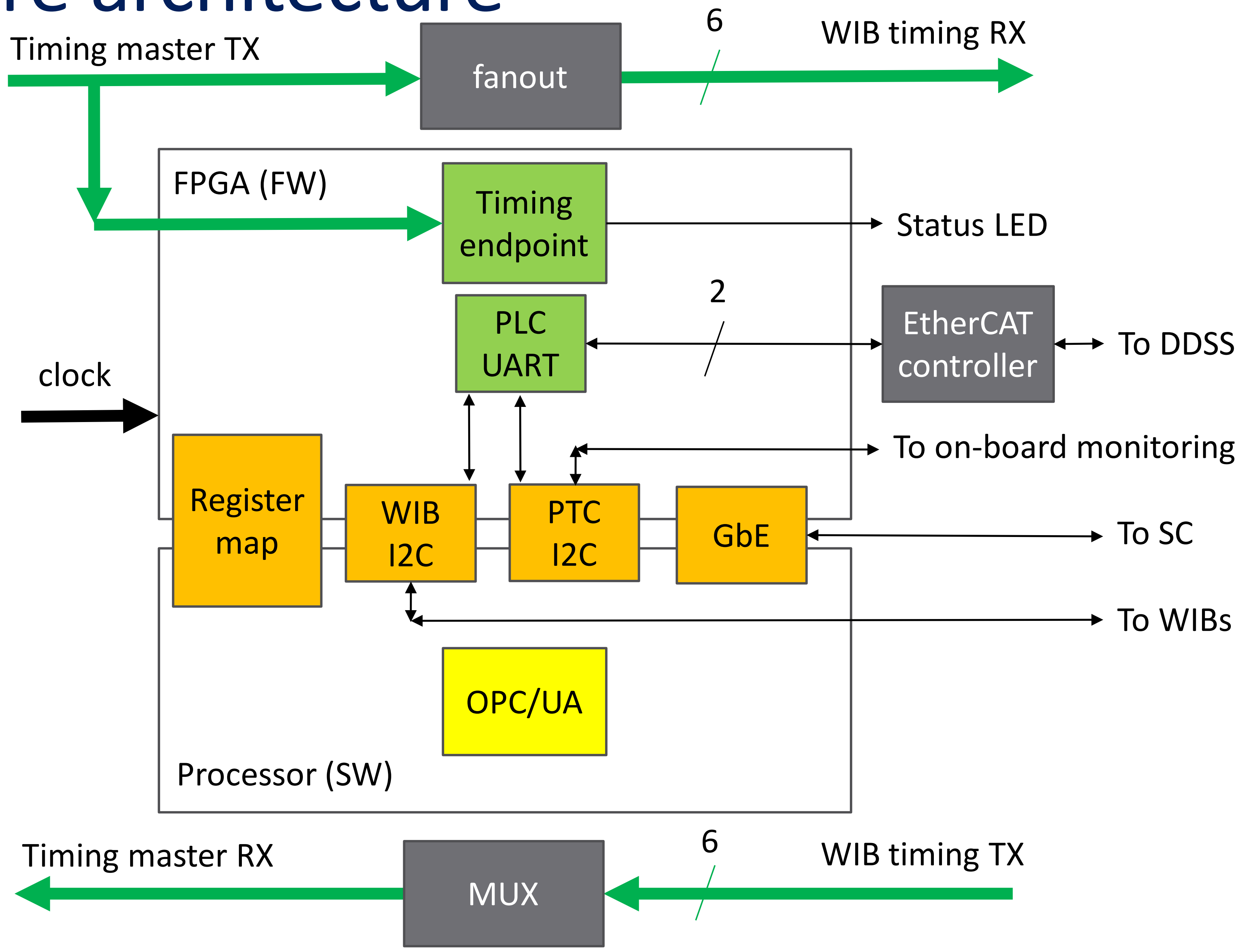
0x1033 raw ADC value from TMP117 temp sensor
0x1033 = 4147
4147 * 7.285 mC =
30.2C

0x0170 raw ADC value from LTC2945 power monitor
((0x170)>>4) = 23
(23 * 25uV)/2.5mOhm =
230mA

0xcafe is an alignment word

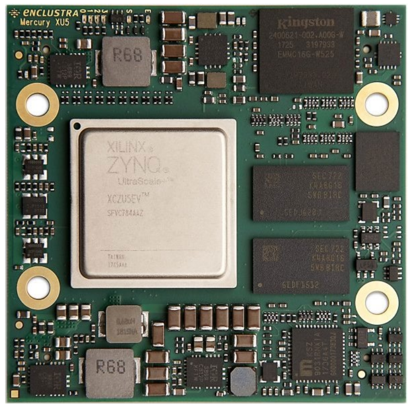
This is a sequence number that goes 0-65535, one increment per group of sensor reads, and then rolls over

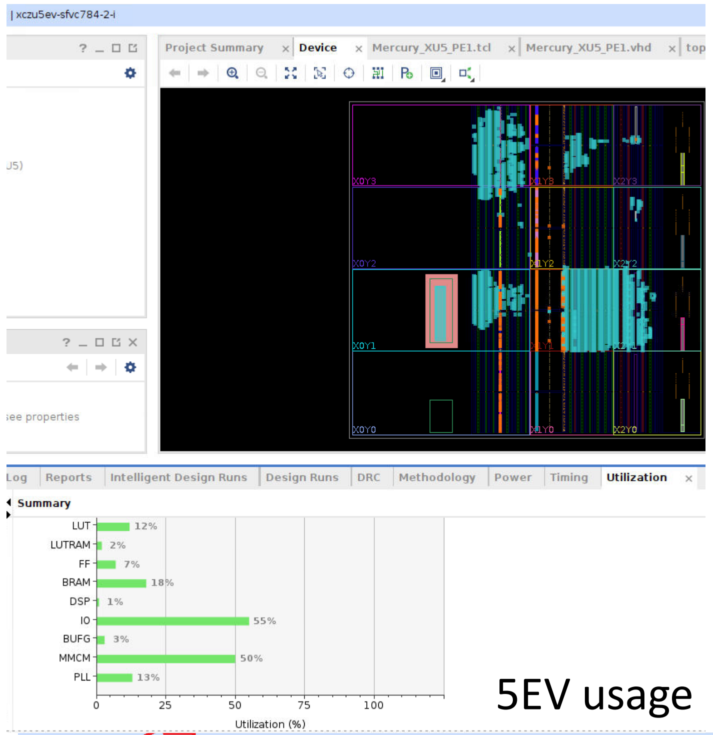
Firmware architecture



FPGA choice, components

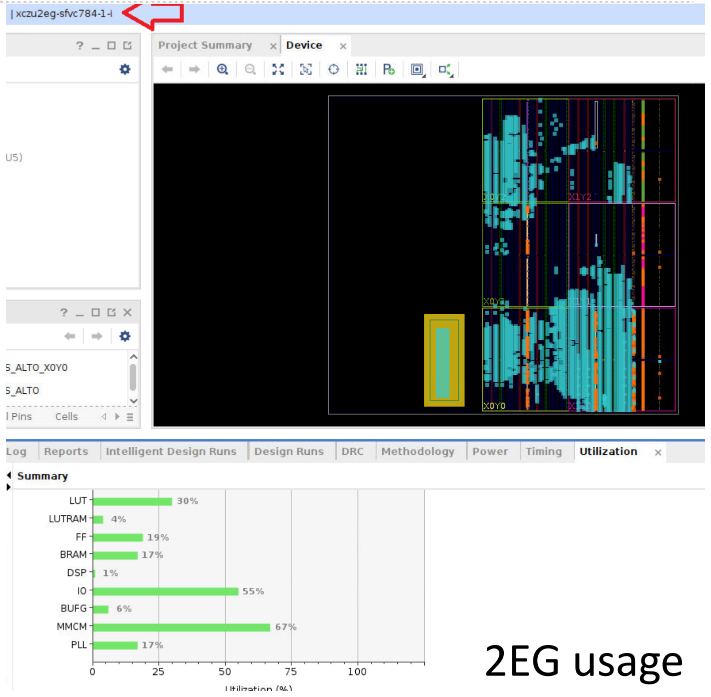
- FPGA history
  - Originally suggested to use same FPGA as WIB
  - Agreed on using commercial mezzanine:
    - Prototype part: Enclustra ME-XU5-5EV-2I-D12E
    - Alternate part: ME-XU5-2EG-1I-D11E
    - Same Xilinx Zynq UltraScale+ FPGA family as WIB
    - Simple to design with, upgradeable during detector life
- Other long lead time components:
  - LTM8064 DC-DC converters for WIBs – purchased
  - LTC2945 power monitors – purchased
  - FPGA – purchase pending cooling study results





Resource	Utilization
LUT	12%
LUTRAM	2%
FF	7%
BRAM	18%
DSP	1%
IO	55%
BUFG	3%
MMCM	50%
PLL	13%

5EV usage

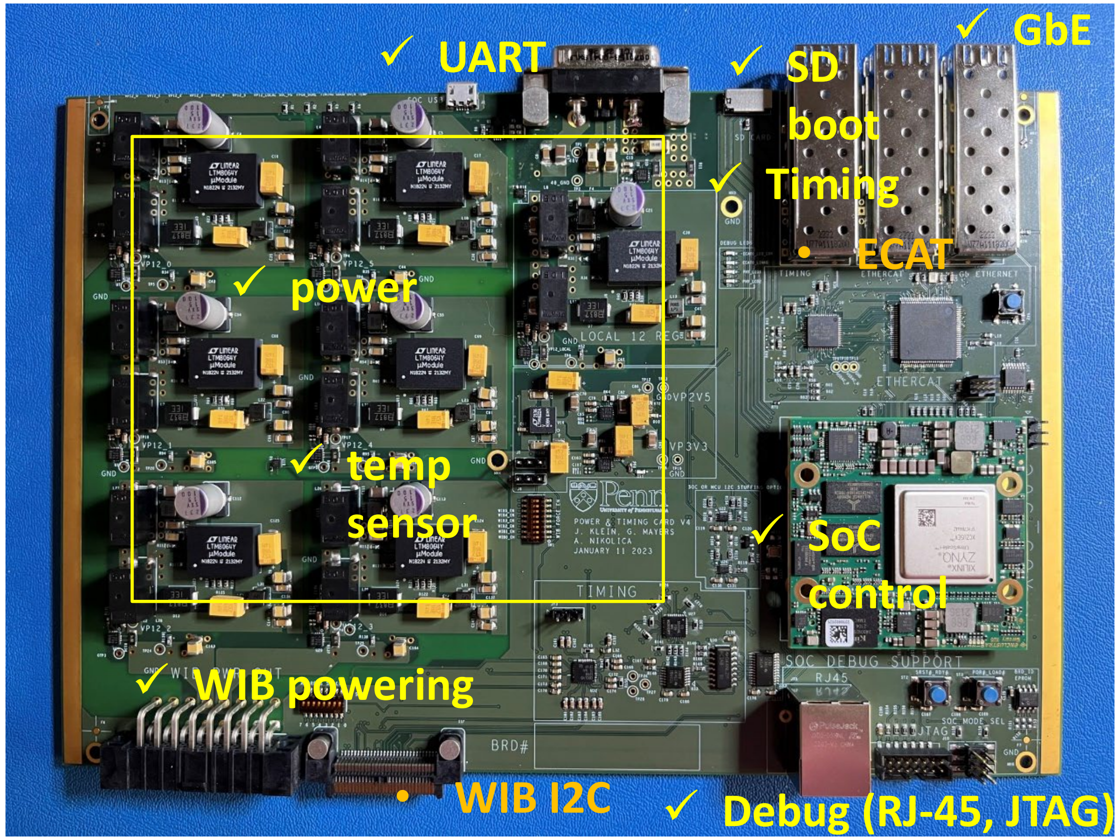


Resource	Utilization
LUT	30%
LUTRAM	4%
FF	19%
BRAM	17%
DSP	1%
IO	55%
BUFG	6%
MMCM	67%
PLL	17%

2EG usage

Testing summary

- ✓ All six 12V regulators power up, can be enabled via FPGA register bit
- ✓ Local 12V, 3.3V, 2.5V power all ICs with no excessive current
- ✓ Enclustra Mercury XU5 mezznanine (Zynq 5EV Ultrascale+):
  - ✓ Boots via SD card
  - ✓ Front panel UART, debug RJ-45, and JTAG work
  - ✓ SFP status signals can be read in via FPGA register bits
  - ✓ Can talk to temperature sensors
  - ✓ Can talk to LTC2945 (and LTC2945-1 backup variant)
- ✓ Can power WIBs (one tested, full WEIC at BNL)
- ✓ Timing distribution test (TX and RX) work on bench
- ✓ GbE can talk over front panel
- ✓ WIB I2C preliminarily tested
  - Need to scale up to all sensors
- ✓ EtherCAT – in progress, but can transmit 2 sensor readings to the Beckhoff system at ICEBERG
  - Need to scale up to all 90 sensors
- Errata:
  - ✓ One minor footprint error on reset pushbuttons – already worked around on prototypes
  - ✓ Wrong part purchased for IV monitor – not a design issue; mitigation works
  - ✓ Component change for a level translator – works
  - ✓ Minor signal polarity corrections

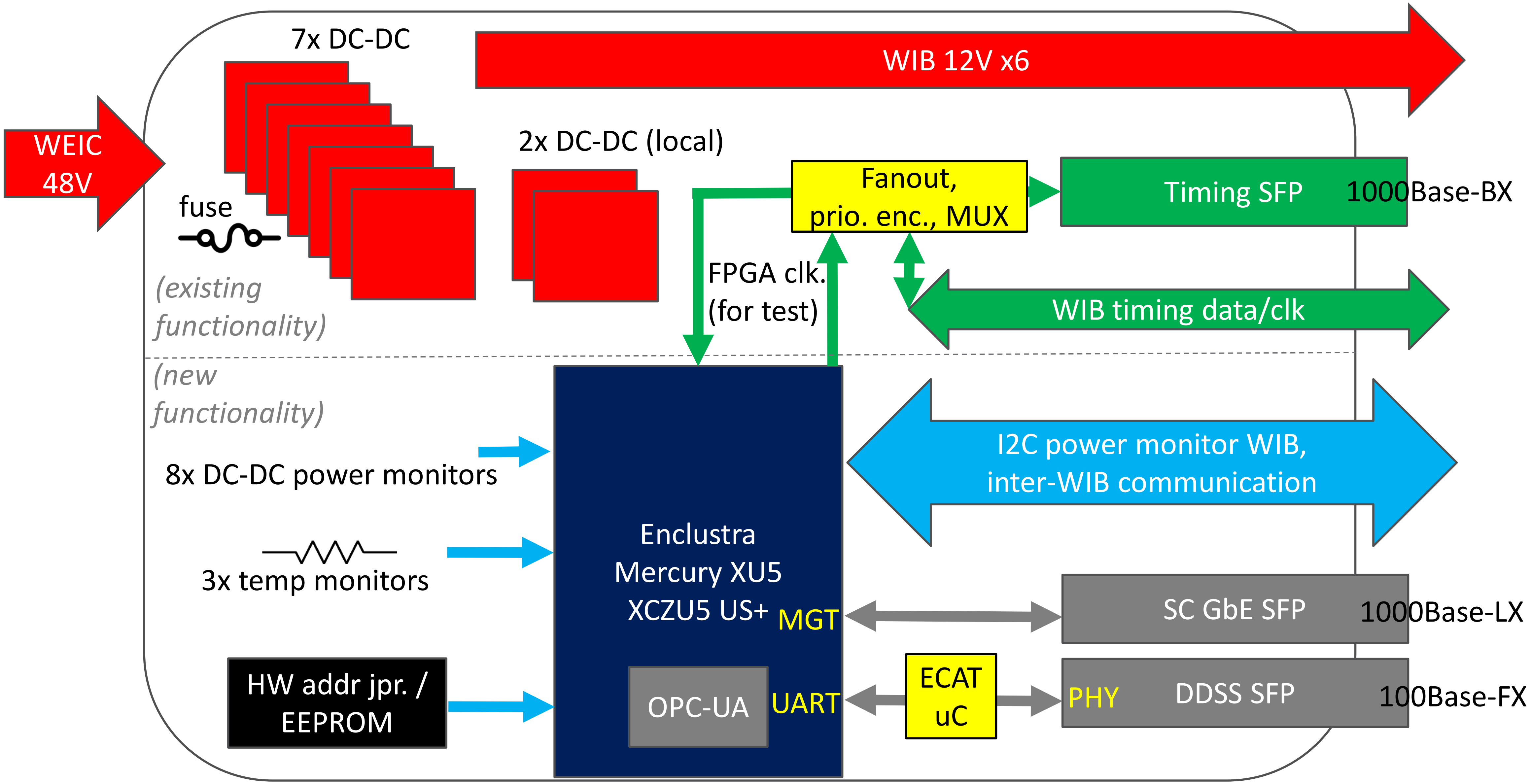


Backup

Architectural questions

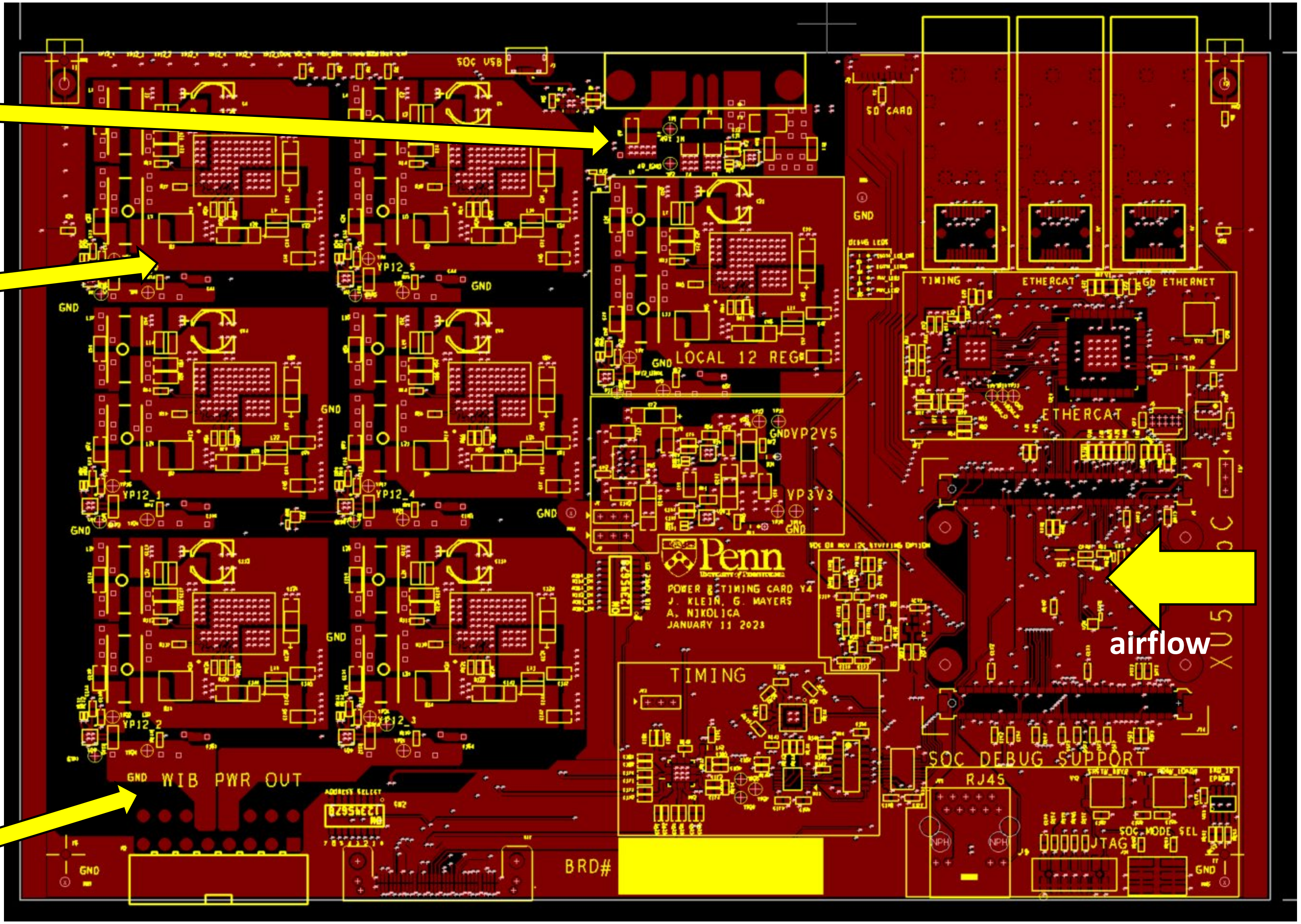
- Powered-off WIBs
  - It was confirmed in conversation with Jack Fried in July 2022 that there is no way to permanently power off a WIB in a crate with PTCv4 without affecting functionality of shared IO lines to other WIBs
  - This is because WIBs have no IO buffering for the following signals: timing TX enables, crate addresses, spare IOs
  - This means a powered off WIB's FPGA can pull down a line (confirmed with measurement)
  - **There are two solutions, for prototypes ONLY:**
    - **Do not power off WIB for the first prototypes; only power cycle if needed, or remove from crate**
  - For the long term, the **WIB will be re-spun with additional buffers added**
    - See: https://indico.fnal.gov/event/61871/contributions/277991/attachments/172265/232697/DUNE_FEMB_WIB_design_and%20revision_10242023.pdf
- IO bandwidth
  - I2C bus @ 400kHz will limit local BW
  - ~90 quantities that can be monitored
  - EtherCAT 100Mbps is also a hard limit total BW
  - Really only need 1-2Hz for a lot of these power and temp measurements – much more reasonable
- Is PTC an independent OPC/UA endpoint, or the OPC/UA endpoint for all WIBs in a crate, or not an OPC/UA endpoint at all?
  - PTCv4 can do any / all options
- On PTC, we retain the same DIP switch and pullup method of setting the 8-bit address (this is in the requirements)
- Current WIB FW does not have the 4 new backplane addresses wired [7:4] – this is a small firmware change on WIB

Top level PTC v4 block diagram



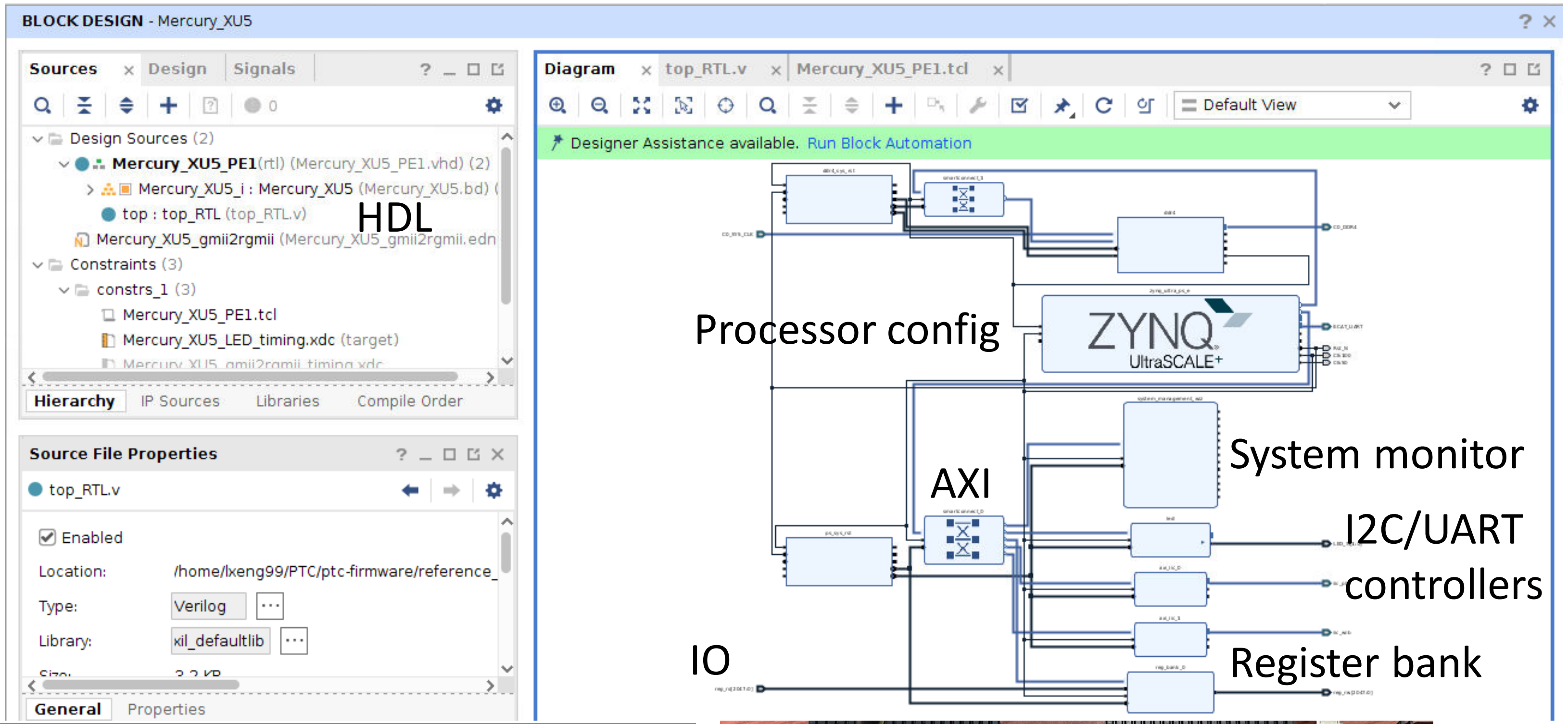
Layout

- Main fusing added
- Regulators layed out as modules.
- Moved to other side so SoC does not intrude on power plane
- All trace widths checked for max power capability
- Isolated return paths



Firmware status

- Snapshot of Vivado project
  - Much work done in software
  - Additional custom firmware can be added in HDL files
- Firmware started on vendor development board
  - Already migrated to PTC and testing succesfully



BLOCK DESIGN - Mercury_XU5

Sources | Design | Signals

```
Design Sources (2)
  Mercury_XU5_PE1(rtl) (Mercury_XU5_PE1.vhd) (2)
    Mercury_XU5_i : Mercury_XU5 (Mercury_XU5.bd) (
    top : top_RTL (top_RTL.v)
  Mercury_XU5_gmii2rgmii (Mercury_XU5_gmii2rgmii.edn
Constraints (3)
  constrs_1 (3)
    Mercury_XU5_PE1.tcl
    Mercury_XU5_LED_timing.xdc (target)
    Mercury_XU5_gmii2rgmii_timing.xdc
```

Hierarchy | IP Sources | Libraries | Compile Order

Source File Properties

top_RTL.v

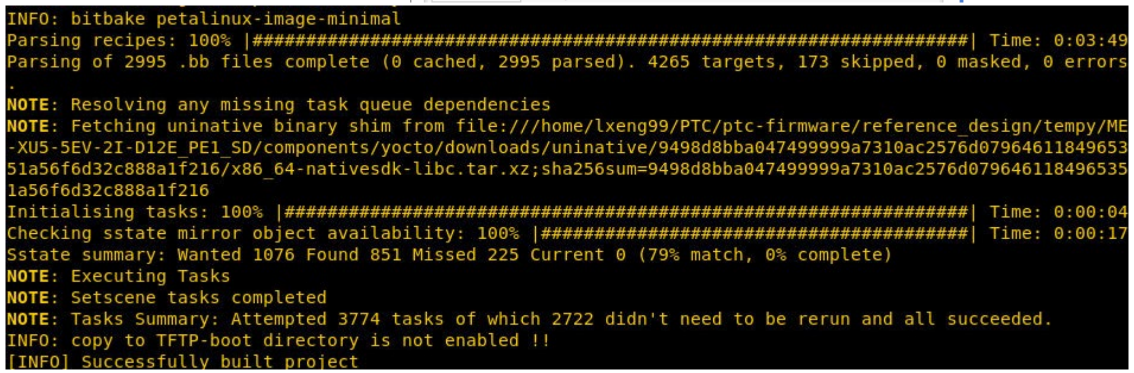
Enabled	✓
Location:	/home/lxeng99/PTC/ptc-firmware/reference_
Type:	Verilog
Library:	xil_defaultlib
Size:	3.2 KB

General | Properties

Diagram | top_RTL.v | Mercury_XU5_PE1.tcl

Default View

Designer Assistance available. Run Block Automation

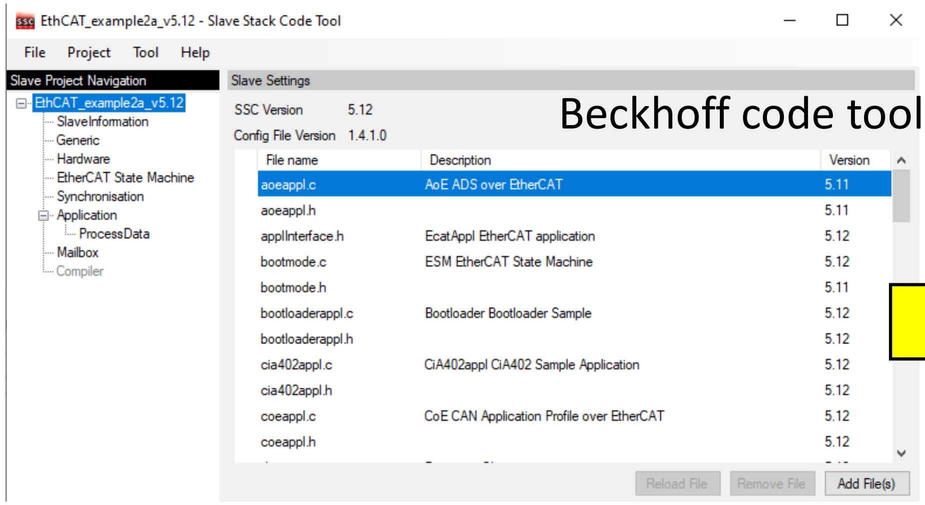


```
INFO: bitbake petalinux-image-minimal
Parsing recipes: 100% |########################################################################| Time: 0:03:49
Parsing of 2995 .bb files complete (0 cached, 2995 parsed). 4265 targets, 173 skipped, 0 masked, 0 errors
.
NOTE: Resolving any missing task queue dependencies
NOTE: Fetching uninative binary shim from file:///home/lxeng99/PTC/ptc-firmware/reference_design/tempy/ME
-XU5-5EV-2I-D12E_PE1_SD/components/yocto/downloads/uninative/9498d8bba047499999a7310ac2576d07964611849653
51a56f6d32c888a1f216/x86_64-nativesdk-libc.tar.xz;sha256sum=9498d8bba047499999a7310ac2576d079646118496535
1a56f6d32c888a1f216
Initialising tasks: 100% |#####################################################################| Time: 0:00:04
Checking sstate mirror object availability: 100% |#############################################| Time: 0:00:17
Sstate summary: Wanted 1076 Found 851 Missed 225 Current 0 (79% match, 0% complete)
NOTE: Executing Tasks
NOTE: Setscene tasks completed
NOTE: Tasks Summary: Attempted 3774 tasks of which 2722 didn't need to be rerun and all succeeded.
INFO: copy to TFTP-boot directory is not enabled !!
[INFO] Successfully built project
```

PetaLinux build environment

EtherCAT software

- Proprietary Beckhoff toolchain generates .c files for EtherCAT from Excel/.xml config files
- We got an example working on the development board
  - Still in process of migrating to PTC



EthCAT_example2a_v5.12 - Slide Stack Code Tool

Slave Project Navigation: EthCAT_example2a_v5.12 – SlaveInformation, Generic, Hardware, EtherCAT State Machine, Synchronisation, Application (ProcessData), Mailbox, Compiler

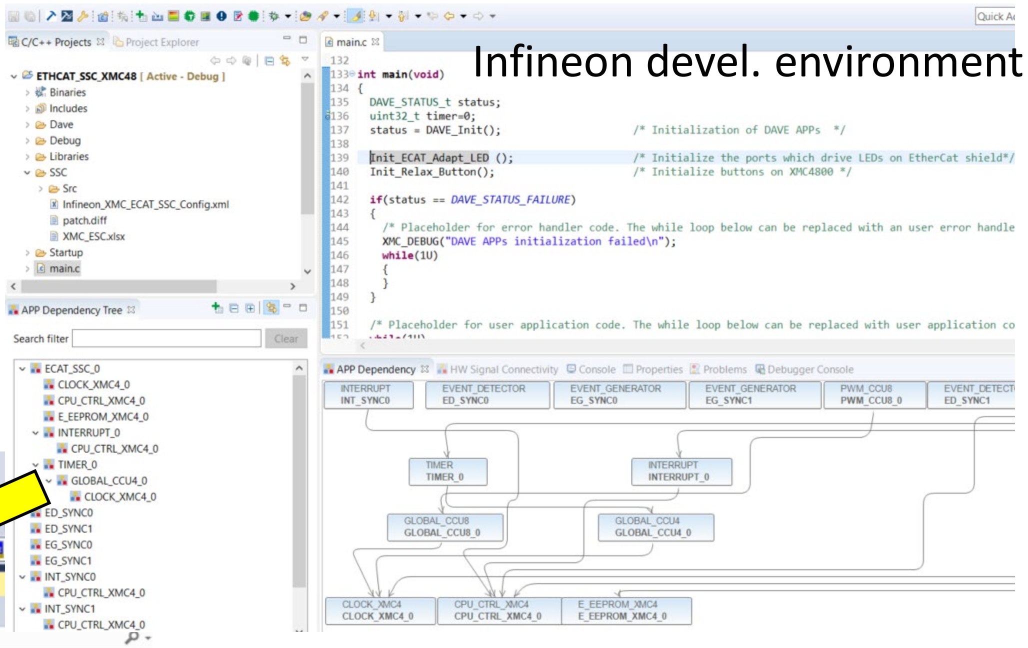
Slave Settings

SSC Version 5.12

Config File Version 1.4.1.0

File name	Description	Version
aoeappl.c	AoE ADS over EtherCAT	5.11
aoeappl.h		5.11
applInterface.h	EcatAppl EtherCAT application	5.12
bootmode.c	ESM EtherCAT State Machine	5.12
bootmode.h		5.11
bootloaderappl.c	Bootloader Bootloader Sample	5.12
bootloaderappl.h		5.12
cia402appl.c	CiA402appl CiA402 Sample Application	5.12
cia402appl.h		5.12
coeappl.c	CoE CAN Application Profile over EtherCAT	5.12
coeappl.h		5.12

Beckhoff code tool



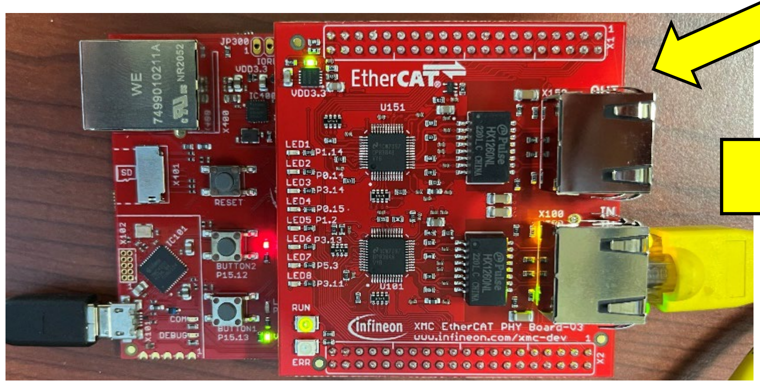
Infineon devel. environment

C/C++ Projects: ETHCAT_SSC_XMC48 [Active - Debug] – Binaries, Includes, Dave, Debug, Libraries, SSC (Src, Infineon_XMC_ECAT_SSC_Config.xml, patch.diff, XMC_ESC.xlsx), Startup, main.c

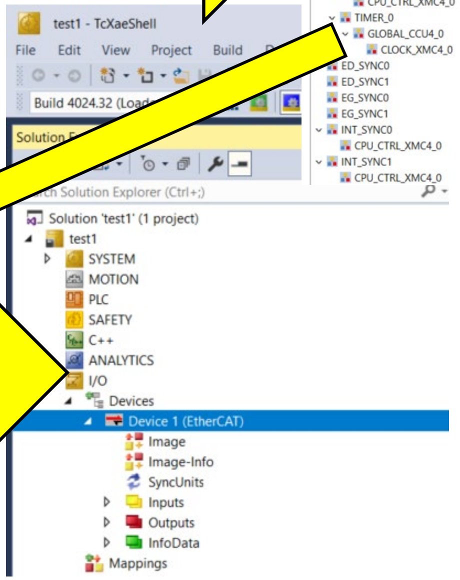
main.c

```
int main(void)
{
  DAVE_STATUS_t status;
  uint32_t timer=0;
  status = DAVE_Init();                  /* Initialization of DAVE APPs  */

  Init_ECAT_Adapt_LED ();                /* Initialize the ports which drive LEDs on EtherCat shield*/
  Init_Relax_Button();                   /* Initialize buttons on XMC4800 */

  if(status == DAVE_STATUS_FAILURE)
  {
    /* Placeholder for error handler code. The while loop below can be replaced with an user error handle
    XMC_DEBUG("DAVE APPs initialization failed\n");
    while(1U)
    {
    }
  }

  /* Placeholder for user application code. The while loop below can be replaced with user application co
```

APP Dependency Tree: ECAT_SSC_0 (CLOCK_XMC4_0, CPU_CTRL_XMC4_0, E_EEPROM_XMC4_0), INTERRUPT_0 (CPU_CTRL_XMC4_0), TIMER_0 (GLOBAL_CCU4_0 (CLOCK_XMC4_0)), ED_SYNC0, ED_SYNC1, EG_SYNC0, EG_SYNC1, INT_SYNC0 (CPU_CTRL_XMC4_0), INT_SYNC1 (CPU_CTRL_XMC4_0)



Infineon devel. board

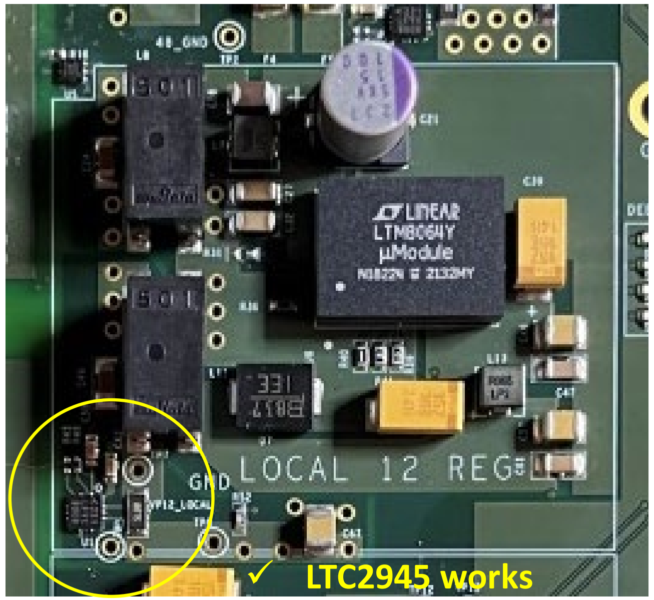


test1 - TcXaeShell

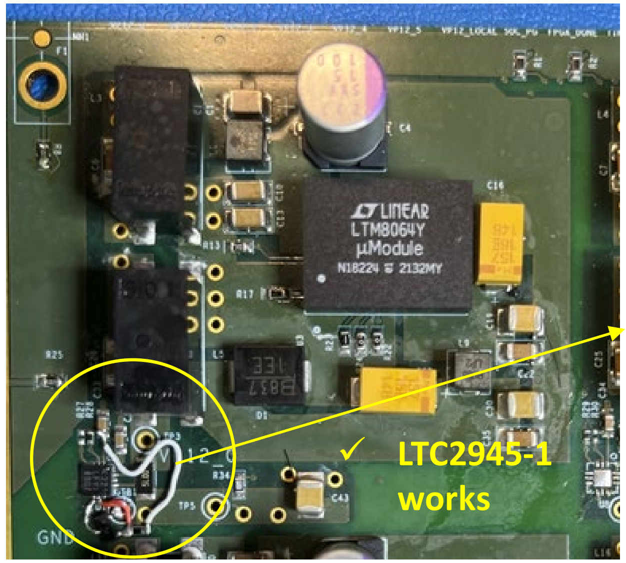
Solution 'test1' (1 project): test1 – SYSTEM, MOTION, PLC, SAFETY, C++, ANALYTICS, I/O – Devices – Device 1 (EtherCAT) – Image, Image-Info, SyncUnits, Inputs, Outputs, InfoData; Mappings

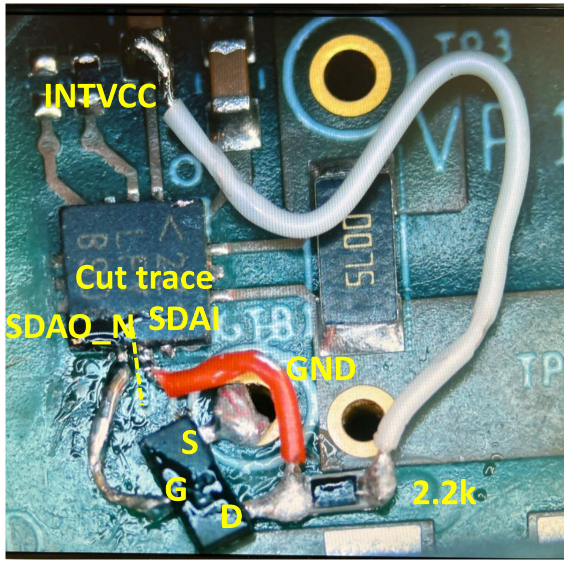
Beckhoff TwinCAT EtherCAT software master

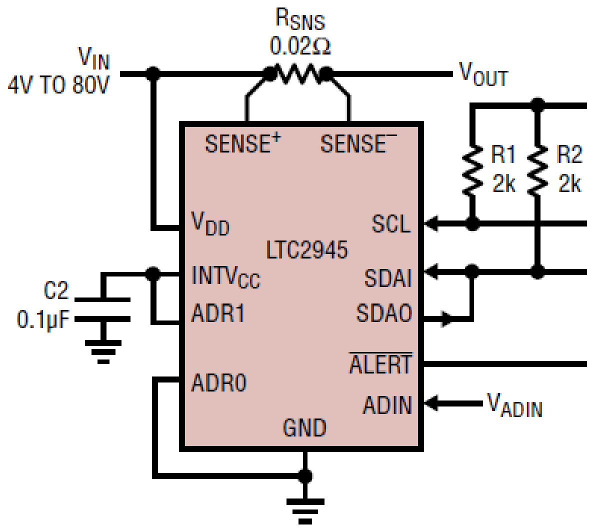
I2C: ability to use LTC2945 or LTC2945-1



OR

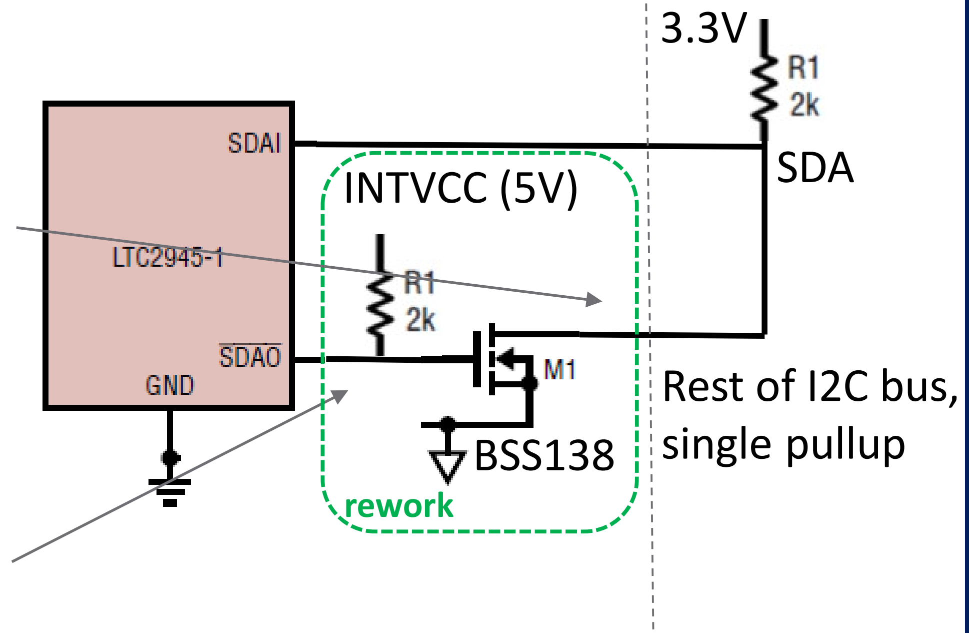




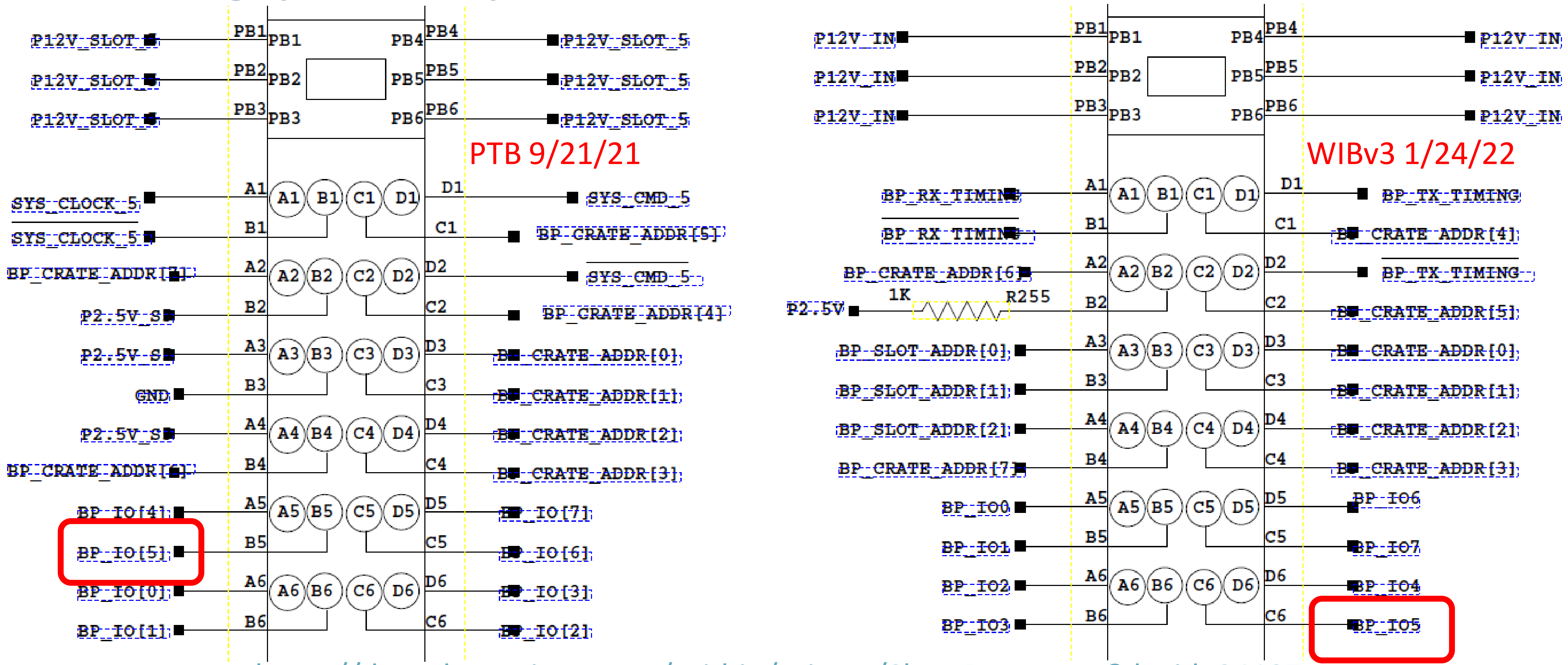


Normally open drain, pulled high by bus controller

Normally pulled low by IC



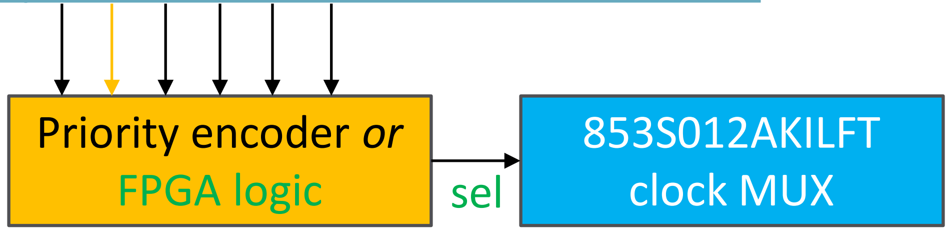
Timing priority encode WIB->PTB



https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=24127

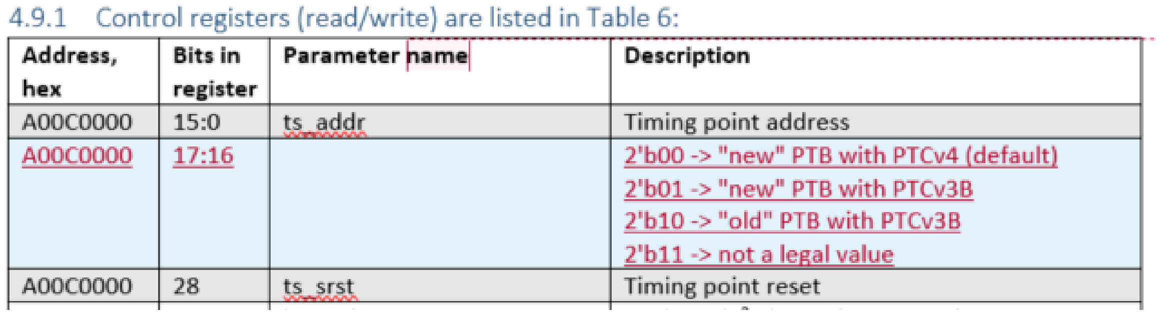
PTCv4 has followed latest PTB mappings

Taken into account in WIB FW



Timing priority encode WIB->PTB

- There are three possible combinations of hardware:

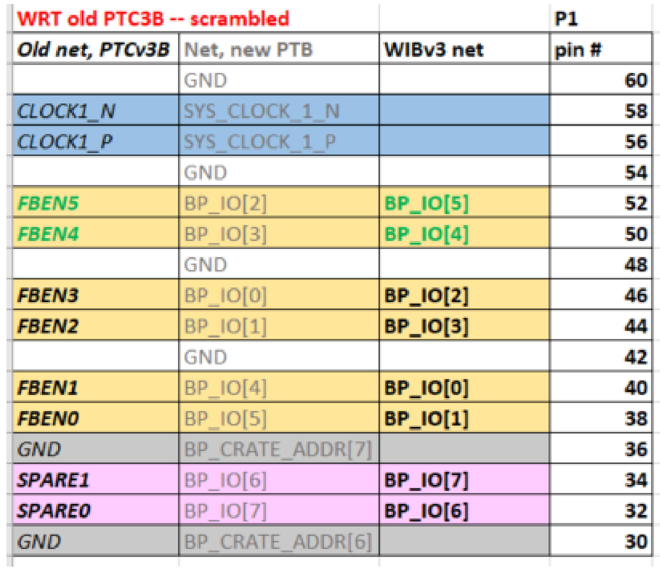


4.9.1 Control registers (read/write) are listed in Table 6:

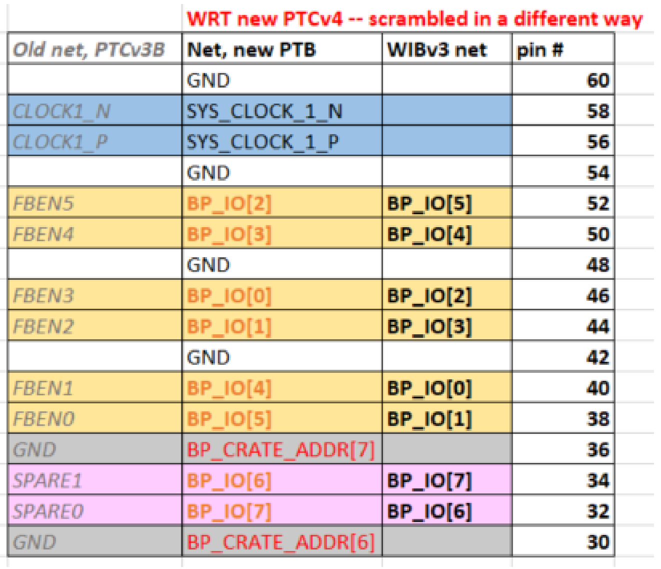
Address, hex	Bits in register	Parameter name	Description
A00C0000	15:0	ts_addr	Timing point address
A00C0000	17:16		2'b00 -> "new" PTB with PTCv4 (default) 2'b01 -> "new" PTB with PTCv3B 2'b10 -> "old" PTB with PTCv3B 2'b11 -> not a legal value
A00C0000	28	ts_srst	Timing point reset

- Change WIB lines mappings in FW, with register bit to select option 1) 2) 3), above
  - WIB firmware addition already tested and deployed

- In production, there will only be one valid mapping



WRT old PTC3B -- scrambled			P1
Old net, PTCv3B	Net, new PTB	WIBv3 net	pin #
	GND		60
CLOCK1_N	SYS_CLOCK_1_N		58
CLOCK1_P	SYS_CLOCK_1_P		56
	GND		54
FBEN5	BP_IO[2]	BP_IO[5]	52
FBEN4	BP_IO[3]	BP_IO[4]	50
	GND		48
FBEN3	BP_IO[0]	BP_IO[2]	46
FBEN2	BP_IO[1]	BP_IO[3]	44
	GND		42
FBEN1	BP_IO[4]	BP_IO[0]	40
FBEN0	BP_IO[5]	BP_IO[1]	38
GND	BP_CRATE_ADDR[7]		36
SPARE1	BP_IO[6]	BP_IO[7]	34
SPARE0	BP_IO[7]	BP_IO[6]	32
GND	BP_CRATE_ADDR[6]		30



	WRT new PTCv4 -- scrambled in a different way		
Old net, PTCv3B	Net, new PTB	WIBv3 net	pin #
	GND		60
CLOCK1_N	SYS_CLOCK_1_N		58
CLOCK1_P	SYS_CLOCK_1_P		56
	GND		54
FBEN5	BP_IO[2]	BP_IO[5]	52
FBEN4	BP_IO[3]	BP_IO[4]	50
	GND		48
FBEN3	BP_IO[0]	BP_IO[2]	46
FBEN2	BP_IO[1]	BP_IO[3]	44
	GND		42
FBEN1	BP_IO[4]	BP_IO[0]	40
FBEN0	BP_IO[5]	BP_IO[1]	38
GND	BP_CRATE_ADDR[7]		36
SPARE1	BP_IO[6]	BP_IO[7]	34
SPARE0	BP_IO[7]	BP_IO[6]	32
GND	BP_CRATE_ADDR[6]		30