MZTIO

发布 *0.0*

Hongyi Wu(吴鸿毅)

2020 年 07 月 01 日

Contents:

CHAPTER 1

README

If you need firmware, please contact Hongyi Wu(wuhongyi@qq.com)

If you want to know how PKU uses MZTIO, please click on the link below: PKUMZTIO

XIA SUPPORT: XIA Blog

The Pixie-16 MZ-TrigIO is designed to route signals from the backplane (rear connectors) to the front panel (front connectors) and make logical combinations between them in FPGA fabric. It has the following features and capabilities:

- Ethernet programmable trigger/coincidence control module for the Pixie-16
- 48+ Pixie-16 backplane trigger connections to local Zynq processor
- 48 front panel LVDS connections to local Zynq processor
- MicroZed Zynq processor with embedded Linux, acting as a standalone PC with built-in SD card drive, USB host, 10/100 Ethernet, webserver, etc
- 1588 PTP and SyncE clock synchronization
- Open source user access to software and firmware
- Use as standalone desktop unit or in 6U PXI chassis
- Custom I/O standards via daughtercards

1.1 Safety

Please take a moment to review these safety precautions. They are provided both for your protection and to prevent damage to the Pixie module and connected equipment. This safety information applies to all operators and service personnel.

- Power Source
  - The Pixie-16 MZ-TrigIO module is powered through an AC/DC wall adapter or a PXI backplane. The default adapter has a variety of AC plug attachments for different localities.

  - Please remember to shut down the Linux OS before removing the power plug from the Pixie-16 MZ-TrigIO or powering down the PXI chassis.

- User Adjustments/Disassembly
  - To avoid personal injury, and/or damage, always disconnect power before accessing the module's interior. There are a few jumpers related to clocking on the board that experienced users may want to use.
- Voltage Ratings
  - Signals on the inputs and outputs must not exceed ± 3.3V. Please review the pinout in the appendix before making any connections.
- Daughtercards
  - Daughtercards can be used as alternatives to front panel and rear inputs, which requires caution to avoid conflicts from FPGA outputs and standard connector inputs.
- Servicing and Cleaning
  - To avoid personal injury, and/or damage to the Pixie module or connected equipment, do not attempt to repair or clean the inside of these units.
- Linux Passwords
  - The Pixie-16 MZ-TrigIO Linux OS comes with default user IDs and passwords for 1) SSH login, 2) SMB file sharing, and 3) Web Operations as described below. Users should immediately change these passwords, especially when the Pixie-16 MZ-TrigIO is connected to external networks. Don't let hackers take over your Pixie-16 MZ-TrigIO!
- Linux Backup
  - The Pixie-16 MZ-TrigIO Linux OS is stored on a removable SD card. SD cards' file systems can become corrupted, which would crash the Linux system and make the Pixie-16 MZ-TrigIO unable to operate. Therefore periodic backup of the SD card is recommended, for example using Win32DiskImager. (Byte for byte copy is required).
  - Note that all Linux passwords are stored on the SD card.

1.2 Logic programming

In order to meet the needs of medium and low energy experimental nuclear physics, we have developed the following basic functions.

- signal delay
- signal extend
- coincidence
- multiplicity
- scaler/counter
- down scale
- remote parameter adjustment
- ……

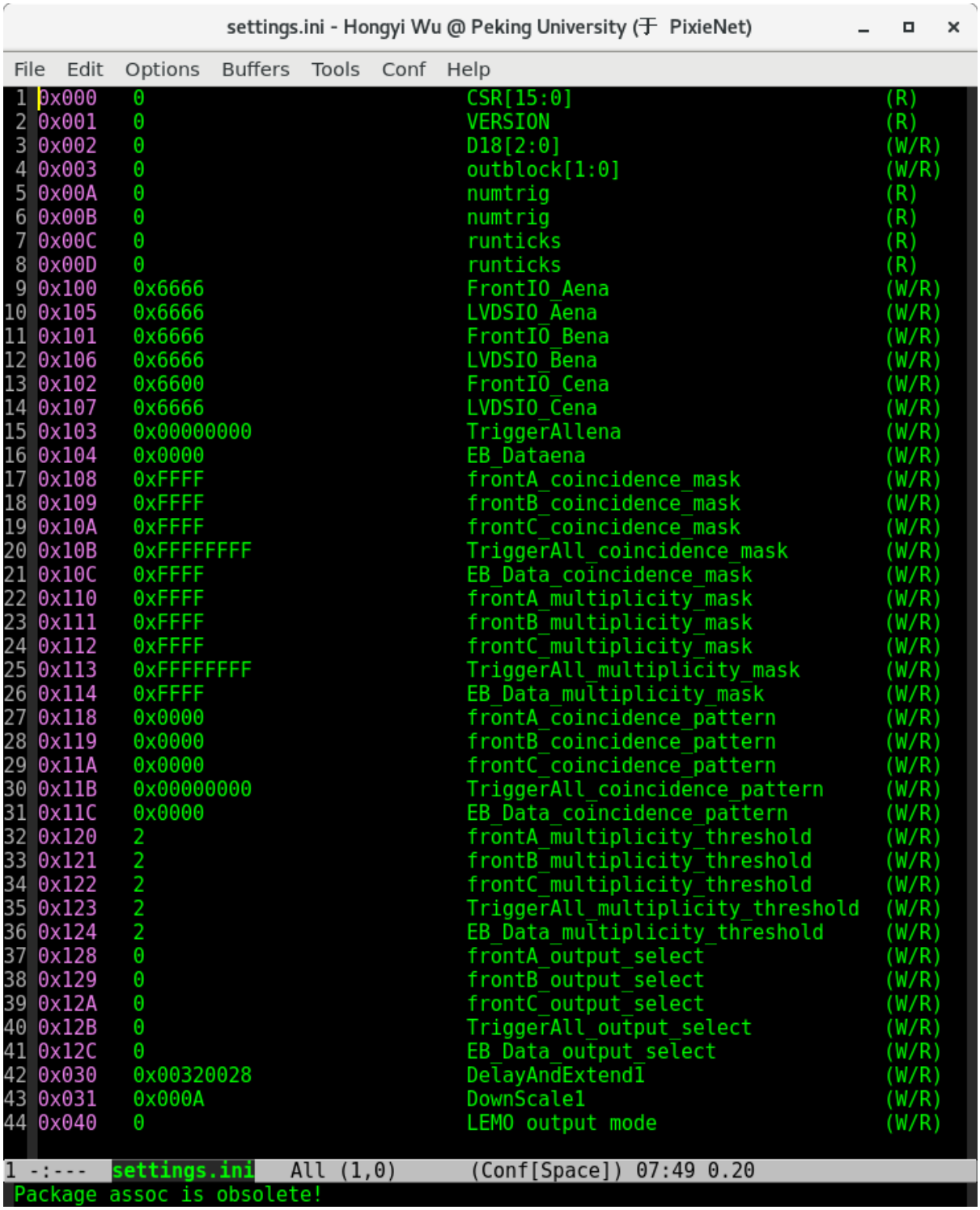
CHAPTER 2

WEB Control GUI

2.1 register

The user can easily adjust the experimental logic by modifying the control registers in the settings.ini file.

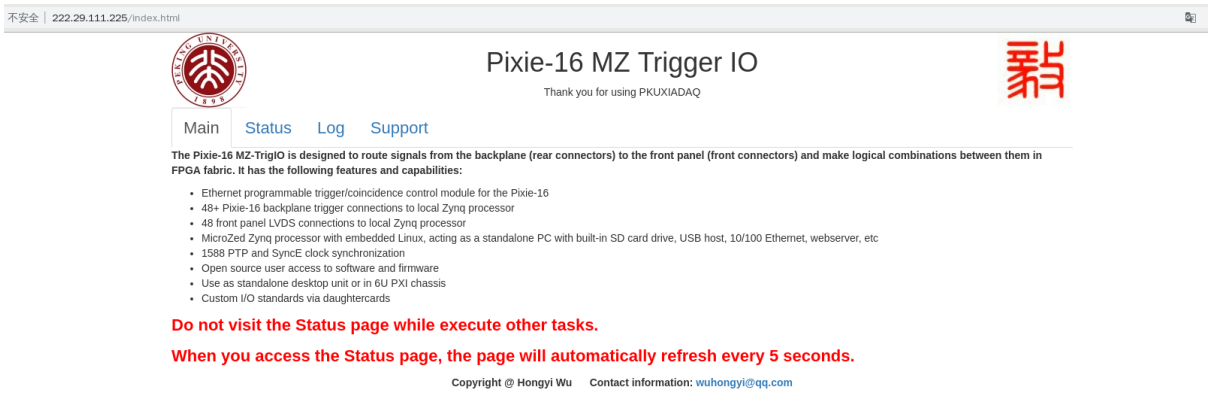
Of course, for different types of experiments, we have specialized software, please refer to the manual of the experiment for the specific register control method.



2.2 web pages

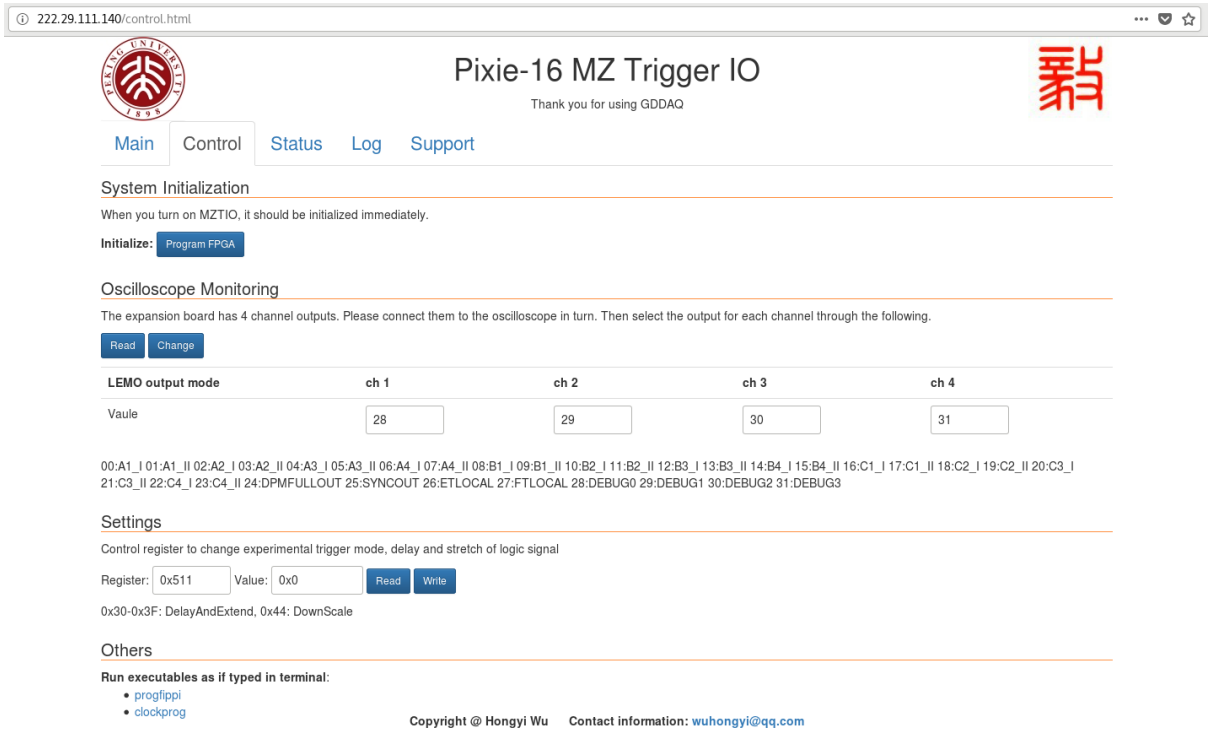
2.2.1 main page

The main page of the web, it will provide basic information and precautions for the module.



2.2.2 control page

The control register is used to change the experimental trigger mode, delay and stretch of logic signals, and so on.



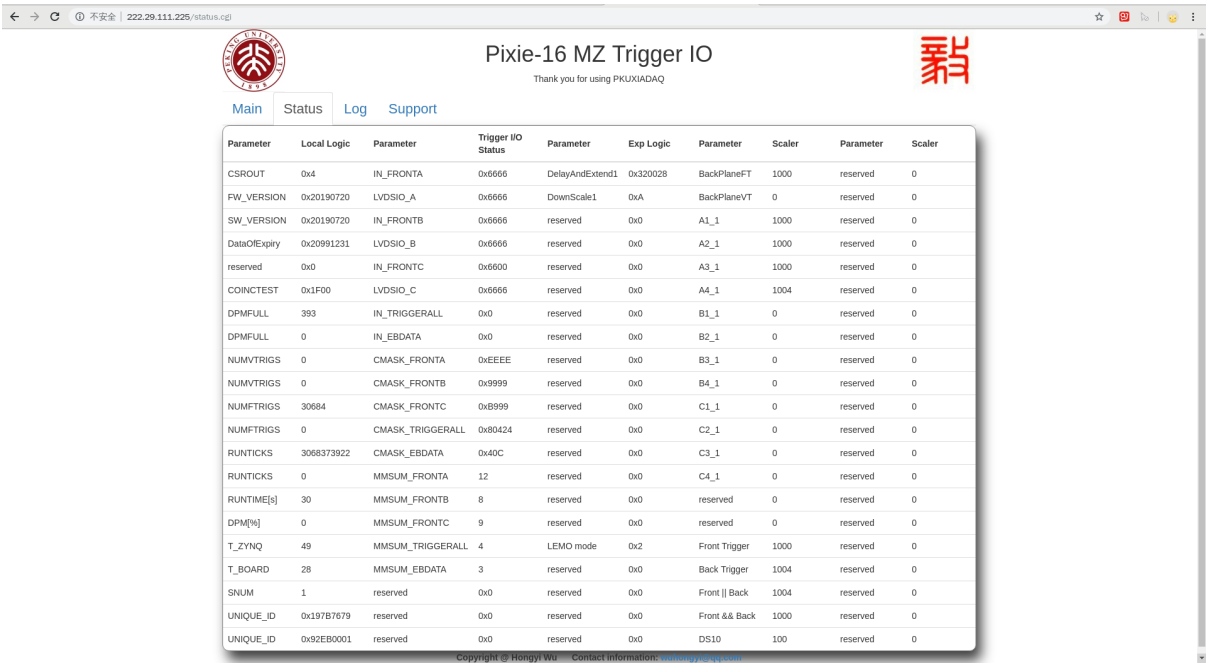
2.2.3 status page

When you aaccess the status page, the page will automatically refresh every 5 second.

There are currently five columns of monitorable parameters on this page.

- The fourth row of the first column indicates the date the solid is allowed to be used.
- The fifteenth line of the first column indicates the running time of the current round of DAQ.
- The first column, line 16, represents the percentage of DPMFULL and total runtime.

The parameters of the third column, the fourth column and the fifth column are determined by the settings of each experiment. For details, please refer to the manual of the specific experiment settings.



Parameter	Local Logic	Parameter	Trigger I/O Status	Parameter	Exp Logic	Parameter	Scaler	Parameter	Scaler
CSROUT	0x4	IN_FRONTA	0x6666	DelayAndExtend1	0x320028	BackPlaneFT	1000	reserved	0
FW_VERSION	0x20190720	LVDSIO_A	0x6666	DownScale1	0xA	BackPlaneVT	0	reserved	0
SW_VERSION	0x20190720	IN_FRONTB	0x6666	reserved	0x0	A1_1	1000	reserved	0
DataOfExpiry	0x20991231	LVDSIO_B	0x6666	reserved	0x0	A2_1	1000	reserved	0
reserved	0x0	IN_FRONTC	0x6600	reserved	0x0	A3_1	1000	reserved	0
COINCTEST	0x1F00	LVDSIO_C	0x6666	reserved	0x0	A4_1	1004	reserved	0
DPMFULL	393	IN_TRIGGERALL	0x0	reserved	0x0	B1_1	0	reserved	0
DPMFULL	0	IN_EBDATA	0x0	reserved	0x0	B2_1	0	reserved	0
NUMVTRIGS	0	CMASK_FRONTA	0xEEEE	reserved	0x0	B3_1	0	reserved	0
NUMVTRIGS	0	CMASK_FRONTB	0x9999	reserved	0x0	B4_1	0	reserved	0
NUMFTRIGS	30684	CMASK_FRONTC	0xB999	reserved	0x0	C1_1	0	reserved	0
NUMFTRIGS	0	CMASK_TRIGGERALL	0x80424	reserved	0x0	C2_1	0	reserved	0
RUNTICKS	3068373922	CMASK_EBDATA	0x40C	reserved	0x0	C3_1	0	reserved	0
RUNTICKS	0	MMSUM_FRONTA	12	reserved	0x0	C4_1	0	reserved	0
RUNTIME[s]	30	MMSUM_FRONTB	8	reserved	0x0	reserved	0	reserved	0
DPM[%]	0	MMSUM_FRONTC	9	reserved	0x0	reserved	0	reserved	0
T_ZYNQ	49	MMSUM_TRIGGERALL	4	LEMO mode	0x2	Front Trigger	1000	reserved	0
T_BOARD	28	MMSUM_EBDATA	3	reserved	0x0	Back Trigger	1004	reserved	0
SNUM	1	reserved	0x0	reserved	0x0	Front || Back	1004	reserved	0
UNIQUE_ID	0x197B7679	reserved	0x0	reserved	0x0	Front && Back	1000	reserved	0
UNIQUE_ID	0x92EB0001	reserved	0x0	reserved	0x0	DS10	100	reserved	0

2.2.4 log page

In development, this page will save the status parameters and read the historical parameters.

2.2.5 support page

This page provides some basic instructions, including XIA instructions, PKU instructions, and more.

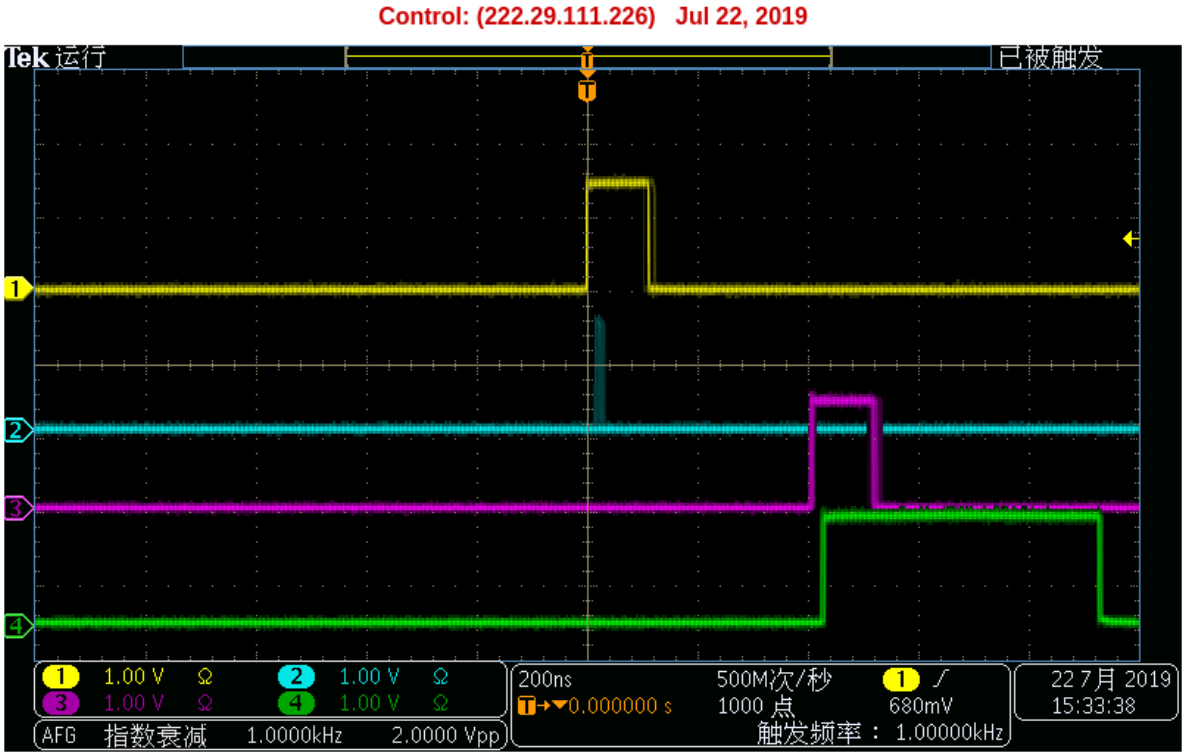
2.3 Oscilloscope

Output signals to the oscilloscope through the MZTIO daughter board.

Most oscilloscopes have only 4 channels, so our monitor settings are set by default for 4 channels. If you want to monitor 8 channels at the same time, you can do it with 2 oscilloscopes.

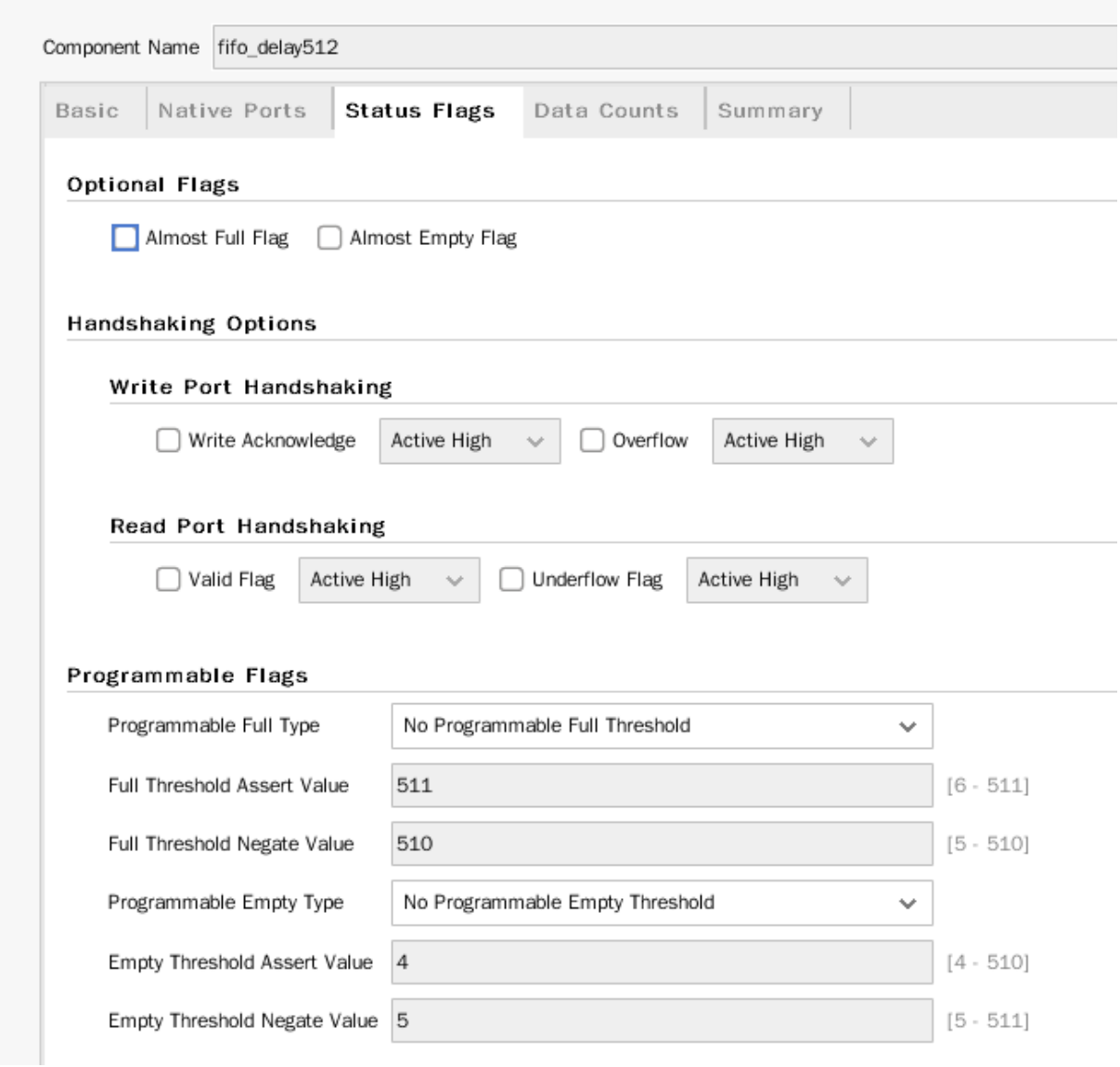
Of course, the monitored signal can be switched by modifying the control register. For instructions on how to monitor different signals, please read the instructions for the specific experiment.

The following figure is an example of oscilloscope monitoring. Line 1 represents the trigger signal, line 2 is the down scale 10, line 3 represents the signal after line 1 is delayed by 400 ns, and line 4 represents line 3 is extend to 500 ns.



2.4 FIFO IP code limits

The figure below shows the settable range of the FIFO IP core parameters.



Due to the limitation of the FIFO IP core, the delay is set to a minimum of 4 clocks.

CHAPTER 3

remote control

3.1 minicom

Connect the USB cable to your computer to get the IP

Serial communication software(minicom) can be used in Linux OS

```
minicom -s
```

```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

- Enter *Serial port setup* ， modify Serial Device to */dev/ttyUSB0* 。Bps/Par/Bits change to *115200 8N1*, the bottom two options are *NO*
- Enter *Modem and dialing* , delete A, B, and K items
- Then select *Save setup as dfl* to save the settings
- Finally, select *Exit* to exit the configuration mode and enter the control mode

```
user: root
password: xia17pxn

The password is the default, so users can log in.
```

Assuming the IP address is 222.29.111.80, you can log in with the following command.

```
ssh -Y root@222.29.111.80
```

3.2 static IP setting

Because Ubuntu 18.04 uses netplan to manage the network. So you can create a file ending in yaml in the /etc/netplan/ directory. For example, the 01-netplan.yaml file.

Then write the following configuration under this file(**You need to modify the IP address and gateway**):

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp3s0:
      dhcp4: no
      addresses: [192.168.1.110/24]
      gateway4:  192.168.1.1
      nameservers:
        addresses: [8.8.8.8, 114.114.114.114]
```

It is important to note that the spaces in each line must be there, otherwise the error will be reported and the setting will fail!

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      addresses: [10.10.6.33/24]
      gateway4: 10.10.6.10
      dhcp4: no
```

The above parameters are the configurations used by the CIAE experiment.

Finally, use *sudo netplan apply* to restart the network service. Use *ip a* to see if your static IP is set up successfully!

CHAPTER 4

ubuntu

4.1 basic configuration

4.1.1 ubuntu 18

If the operating system is the latest version, no additional source configuration is required.

If you want to install CERN ROOT, add the following line to /etc/apt/sources.list

```
deb http://ports.ubuntu.com/ xenial main universe multiverse
```

4.1.2 ubuntu 12

If the operating system version is the previous version, you need to modify the source configuration as follows.

Edit source list file

```
vim /etc/apt/sources.list
```

change into:

```
deb http://old-releases.ubuntu.com/ubuntu vivid main restricted universe multiverse
deb http://old-releases.ubuntu.com/ubuntu vivid-security main restricted universe␣
↪multiverse
deb http://old-releases.ubuntu.com/ubuntu vivid-updates main restricted universe␣
↪multiverse
deb http://old-releases.ubuntu.com/ubuntu vivid-proposed main restricted universe␣
↪multiverse
deb http://old-releases.ubuntu.com/ubuntu vivid-backports main restricted universe␣
↪multiverse
deb-src http://old-releases.ubuntu.com/ubuntu vivid main restricted universe␣
↪multiverse
deb-src http://old-releases.ubuntu.com/ubuntu vivid-security main restricted␣
↪universe multiverse
deb-src http://old-releases.ubuntu.com/ubuntu vivid-updates main restricted␣
↪universe multiverse
deb-src http://old-releases.ubuntu.com/ubuntu vivid-proposed main restricted␣
↪universe multiverse
```

(下页继续)

(续上页)

```
deb-src http://old-releases.ubuntu.com/ubuntu vivid-backports main restricted␣
↪universe multiverse

deb http://mirrors.ustc.edu.cn/ubuntu/ vivid main universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ vivid main universe
```

4.1.3 software upgrade

```
apt-get update
```

```
#install firefox
apt-get install firefox
# install emacs
apt-get install emacs

# ROOT dependent library
apt-get install cmake
apt-get install libx11-dev
apt-get install libxpm-dev
apt-get install libxft-dev
apt-get install libxext-dev
apt-get install gfortran
apt-get install libssl-dev
apt-get install xlibmesa-glu-dev
apt-get install libglew1.5-dev
apt-get install libftgl-dev
apt-get install libmysqlclient-dev
apt-get install libfftw3-dev
apt-get install libcfitsio-dev
apt-get install graphviz-dev
apt-get install libavahi-compat-libdnssd-dev
apt-get install libxml2-dev
apt-get install libkrb5-dev
apt-get install libgsl0-dev
apt-get install libqt4-dev

#install django
apt install python3-pip
pip3 install django==2.2
```

```
apt-get install root-system-bin
```

Ubuntu color configuration, place the color configuration file .dircolors in the personal directory, the file name is .dir_colors in the readhat system.

4.1.4 Time zone select

```
#First check the current system time
date -R
#Check the displayed time zone. If it is not consistent with the local time zone,␣
↪you can modify it in the following ways:

tzselect
#The figure below shows how Chinese users can modify the local time zone. Users in␣
↪other regions can make corresponding selections.
cp /usr/share/zoneinfo/Asia/Shanghai  /etc/localtime
```

(下页继续)

(续上页)

```
#Check if the modification is successful
date -R
```

```
root@ubuntu:/# tzselect
Please identify a location so that time zone rules can be set correctly.
Please select a continent, ocean, "coord", or "TZ".
 1) Africa
 2) Americas
 3) Antarctica
 4) Asia
 5) Atlantic Ocean
 6) Australia
 7) Europe
 8) Indian Ocean
 9) Pacific Ocean
10) coord - I want to use geographical coordinates.
11) TZ - I want to specify the time zone using the Posix TZ format.
#? 4
Please select a country whose clocks agree with yours.
 1) Afghanistan          18) Israel              35) Palestine
 2) Armenia              19) Japan               36) Philippines
 3) Azerbaijan           20) Jordan              37) Qatar
 4) Bahrain              21) Kazakhstan          38) Russia
 5) Bangladesh           22) Korea (North)       39) Saudi Arabia
 6) Bhutan               23) Korea (South)       40) Singapore
 7) Brunei               24) Kuwait              41) Sri Lanka
 8) Cambodia             25) Kyrgyzstan          42) Syria
 9) China                26) Laos                43) Taiwan
10) Cyprus               27) Lebanon             44) Tajikistan
11) East Timor           28) Macau               45) Thailand
12) Georgia              29) Malaysia            46) Turkmenistan
13) Hong Kong            30) Mongolia            47) United Arab Emirates
14) India                31) Myanmar (Burma)     48) Uzbekistan
15) Indonesia            32) Nepal               49) Vietnam
16) Iran                 33) Oman                50) Yemen
17) Iraq                 34) Pakistan
#? 9
Please select one of the following time zone regions.
1) Beijing Time
2) Xinjiang Time
#? 1

The following information has been given:

        China
        Beijing Time

Therefore TZ='Asia/Shanghai' will be used.
Local time is now:      Tue Jan 16 09:29:44 CST 2018.
Universal Time is now:  Tue Jan 16 01:29:44 UTC 2018.
Is the above information OK?
1) Yes
2) No
#? 1

You can make this change permanent for yourself by appending the line
        TZ='Asia/Shanghai'; export TZ
to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you
can use the /usr/bin/tzselect command in shell scripts:
Asia/Shanghai
root@ubuntu:/#
```

4.2 Restore SD card space

In order to speed up the installation speed of the image, only the SD card space of about 8/16G is actually formatted. The 16/32G SD card and the 8/16G space are not used. In order to be able to use, the following operations are performed.

```
fdisk /dev/mmcblk0
# Then enter: d [ENTER],2 [ENTER],n[ENTER] [ENTER],[ENTER],[ENTER],[ENTER],
↪w[ENTER]. Then reboot the OS. If there is a problem, please refer to *Getting␣
↪started with Xillinux for Zynq-7000 EPP*
```

```
# Execute the following command
resize2fs /dev/mmcblk0p2

# Use the following command to view the result
df -h
```

4.3 update the boot files

To mount the SD card boot partition to a folder /mnt/sd, execute

```
mount /dev/mmcblk0p1 /mnt/sd
```

this is useful to update the boot files without removing the SD card. The Pixie-16 MZ-TrigIO has to be rebooted before the new boot files become effective.

So the precedure would be:

- generate FW files on a desktop PC
- copy to shared Linux folder on the SD card (/var/www)
- mount boot partition mount /dev/mmcblk0p1 /mnt/sd (create /mnt/sd if not already there)
- copy files e.g. cp /var/www/xillydemo.bit /mnt/sd
- reboot or power cycle (reboot)

```
scp xillydemo.bit root@222.29.111.157:~
```

4.4 /dev/mmcblk0p1

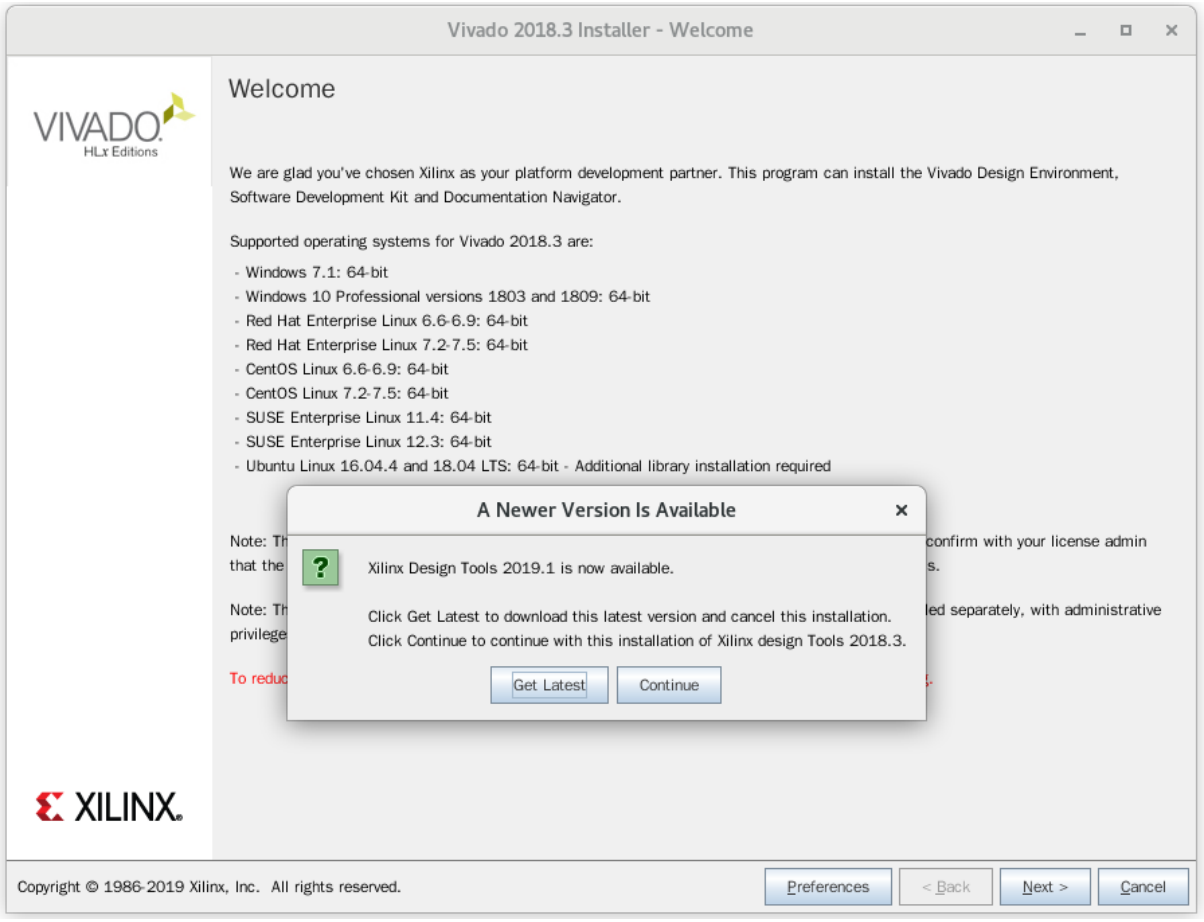
```
boot.bin  devicetree.dtb  uImage  xillydemo.bit
```

CHAPTER 5

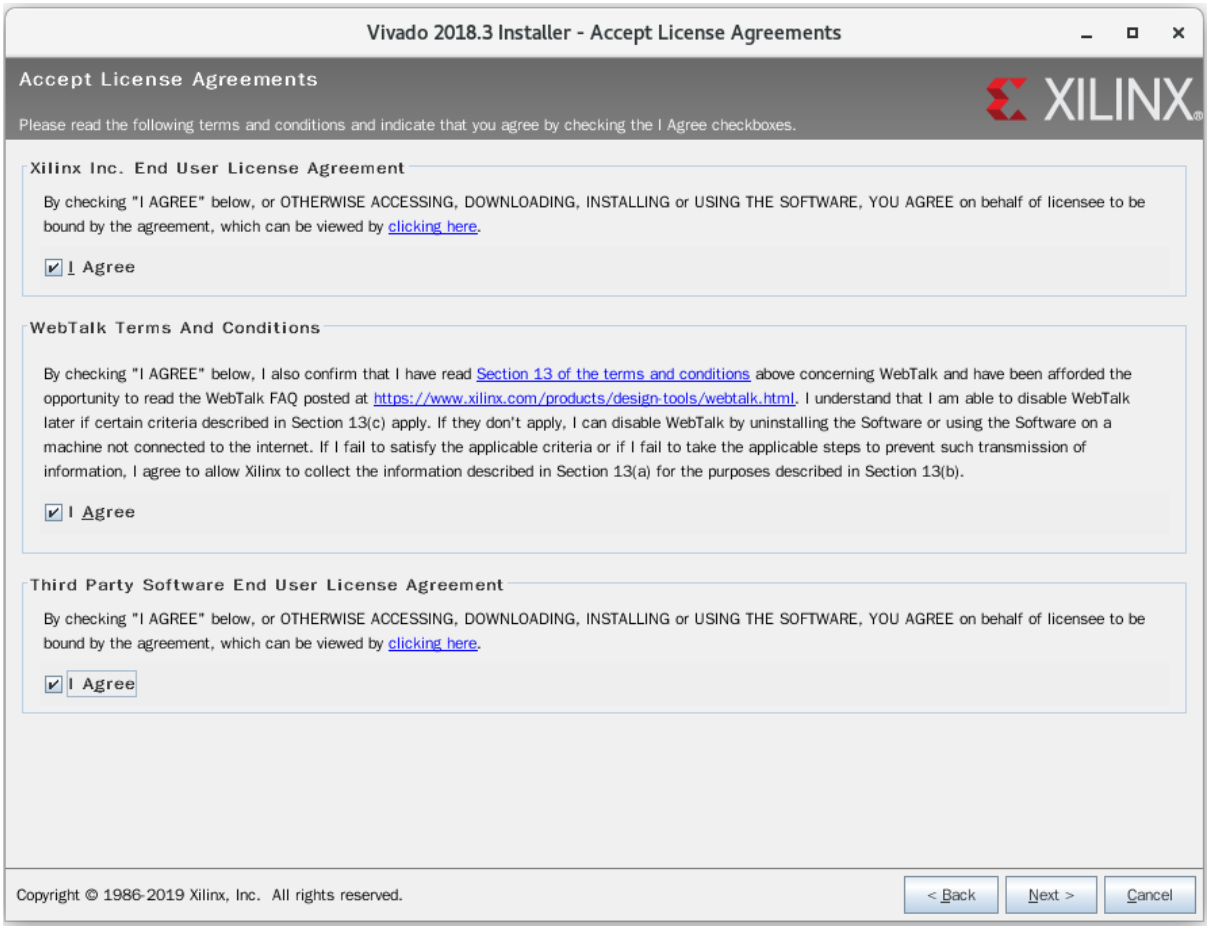
Vivado

5.1 Install

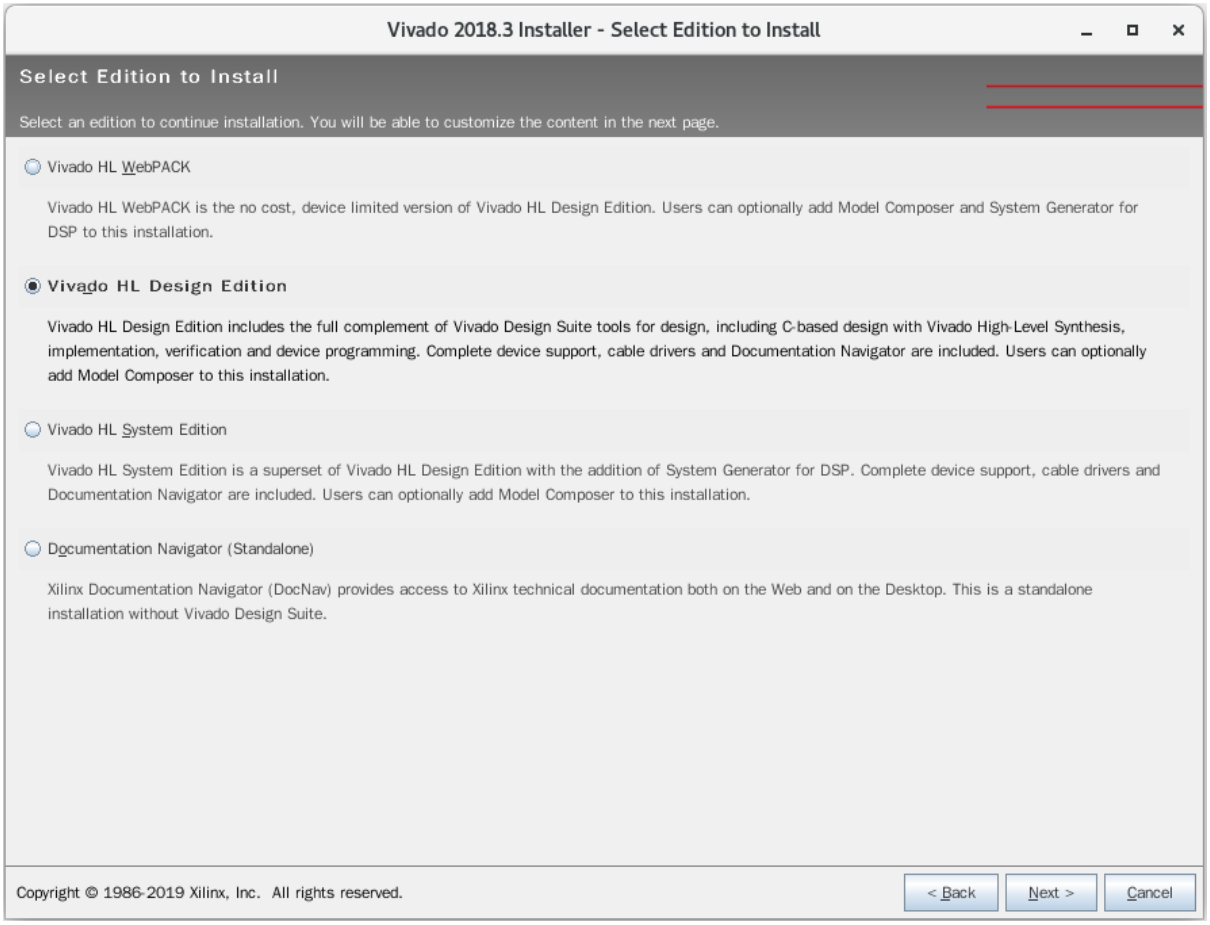
```
tar   -zxvf   Xilinx_Vivado_SDK_2018.3_1207_2324.tar.gz
cd    Xilinx_Vivado_SDK_2018.3_1207_2324
./xsetup
```



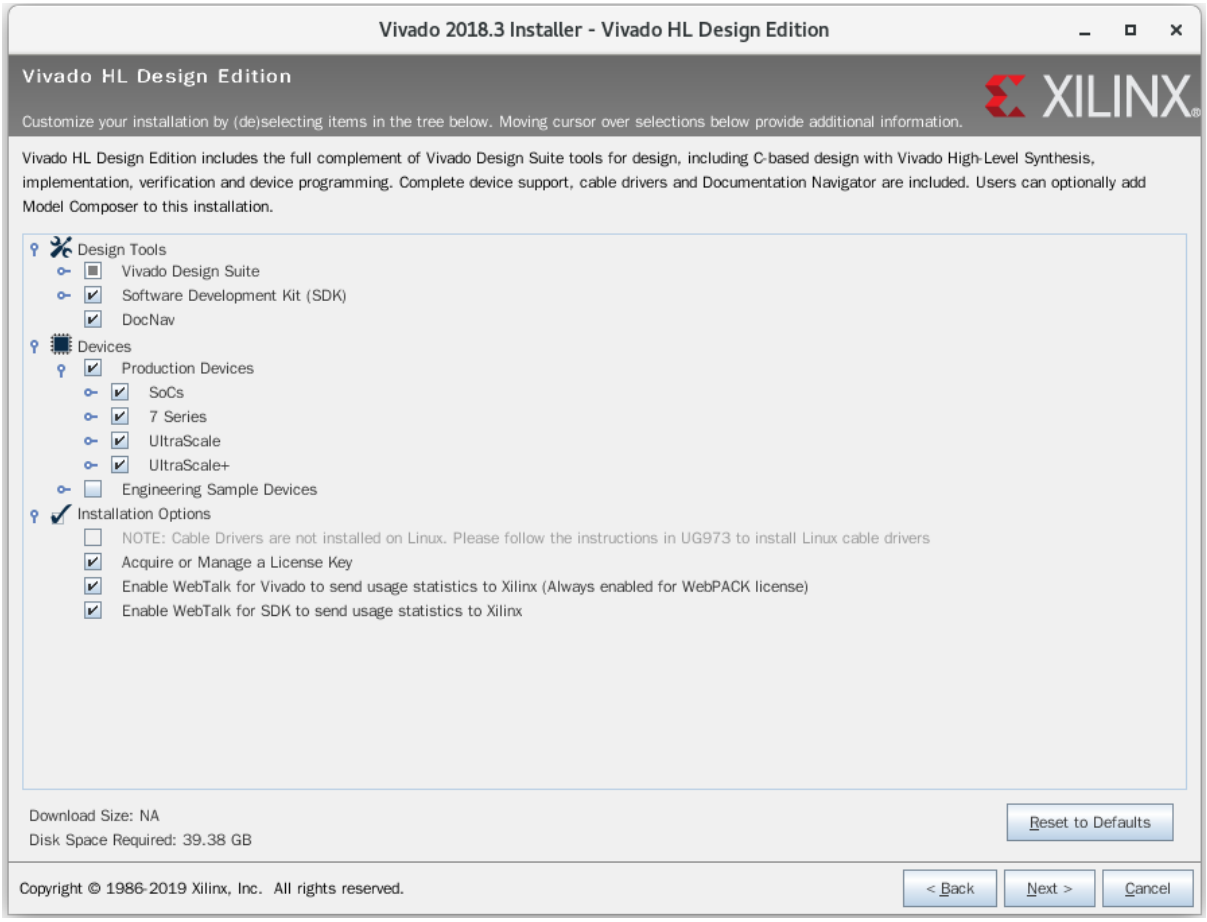
Click “continue” to choose not to download the latest version, then click “Next” to go to the next step



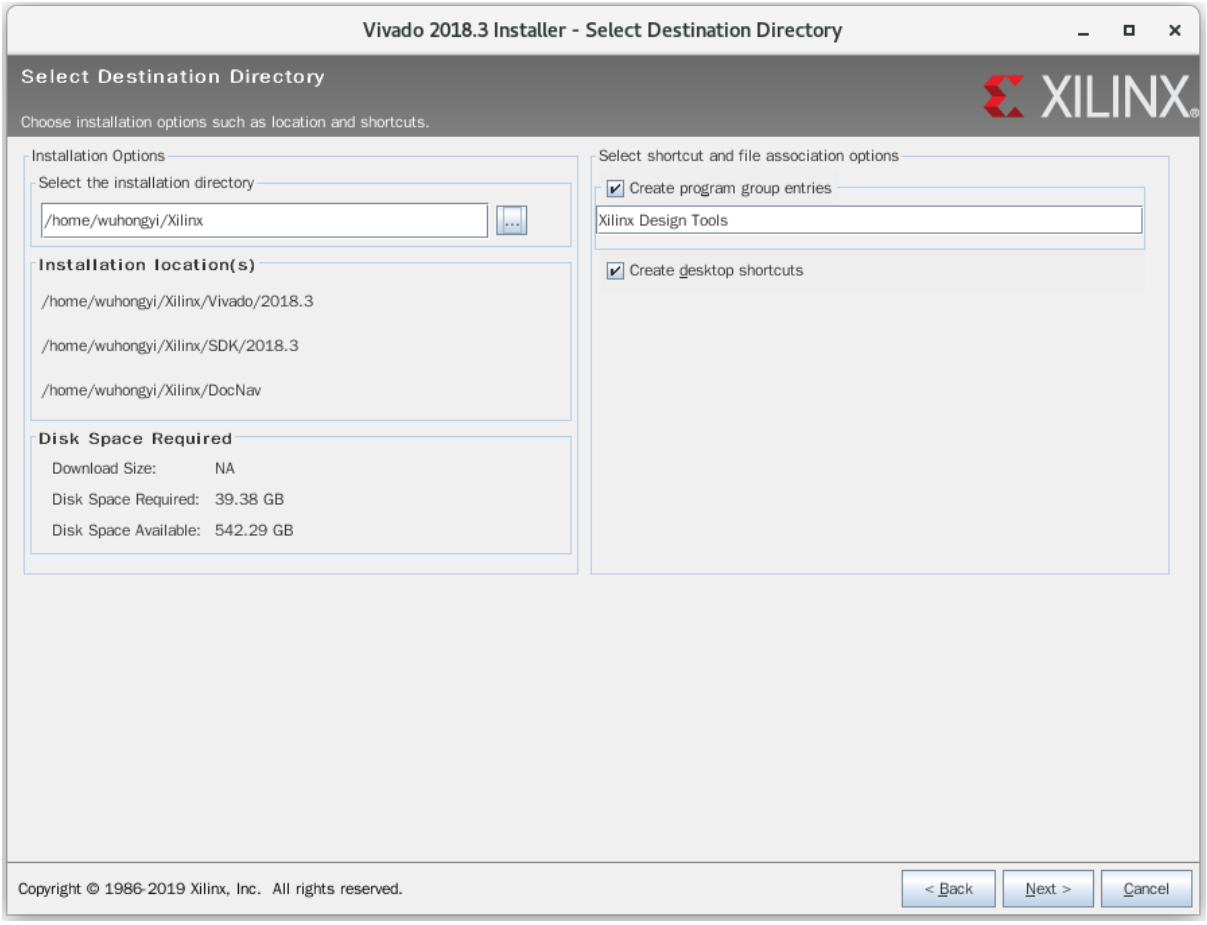
Click on the three optional boxes and then click “Next” to go to the next step



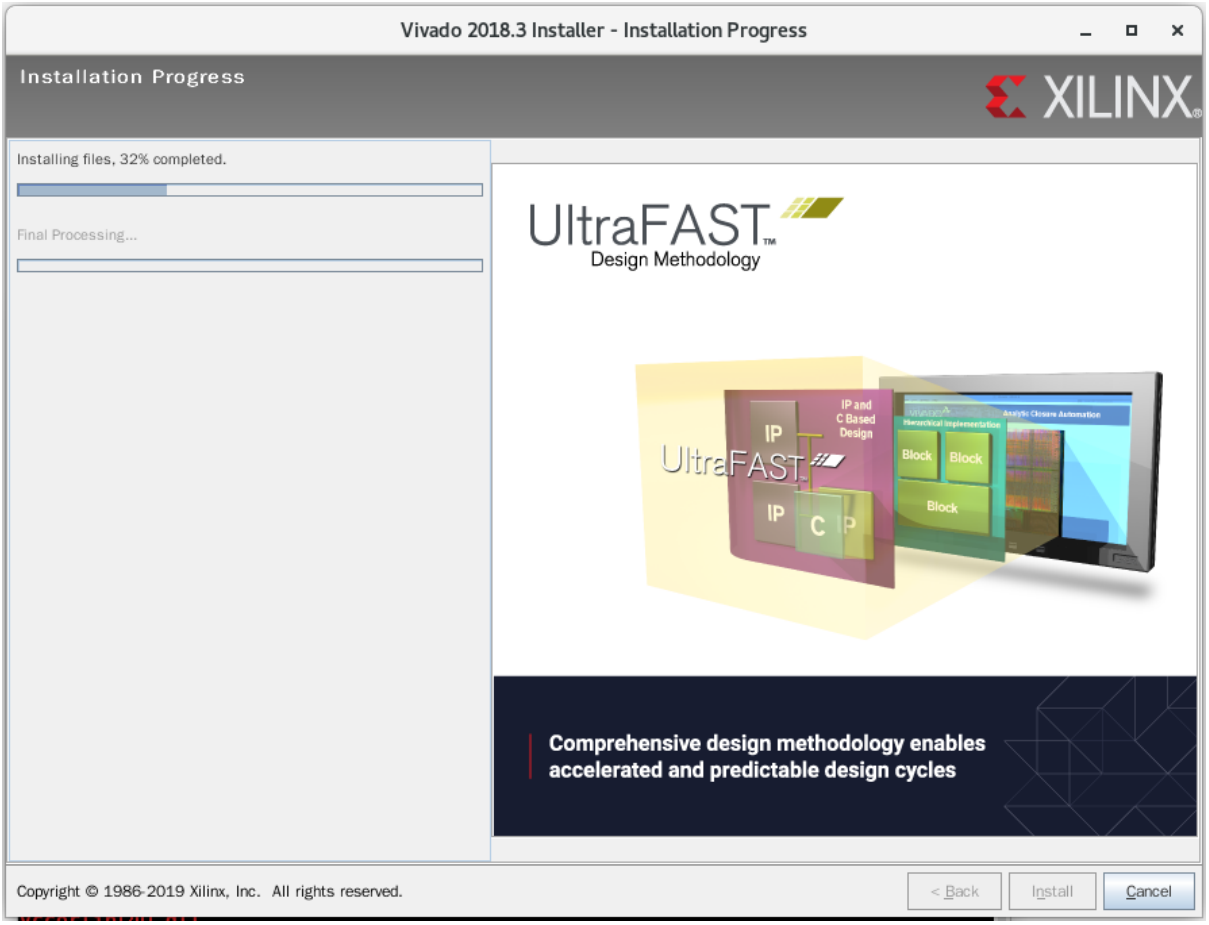
Select “Vinado HL Design Edition” and click “Next” to go to the next step



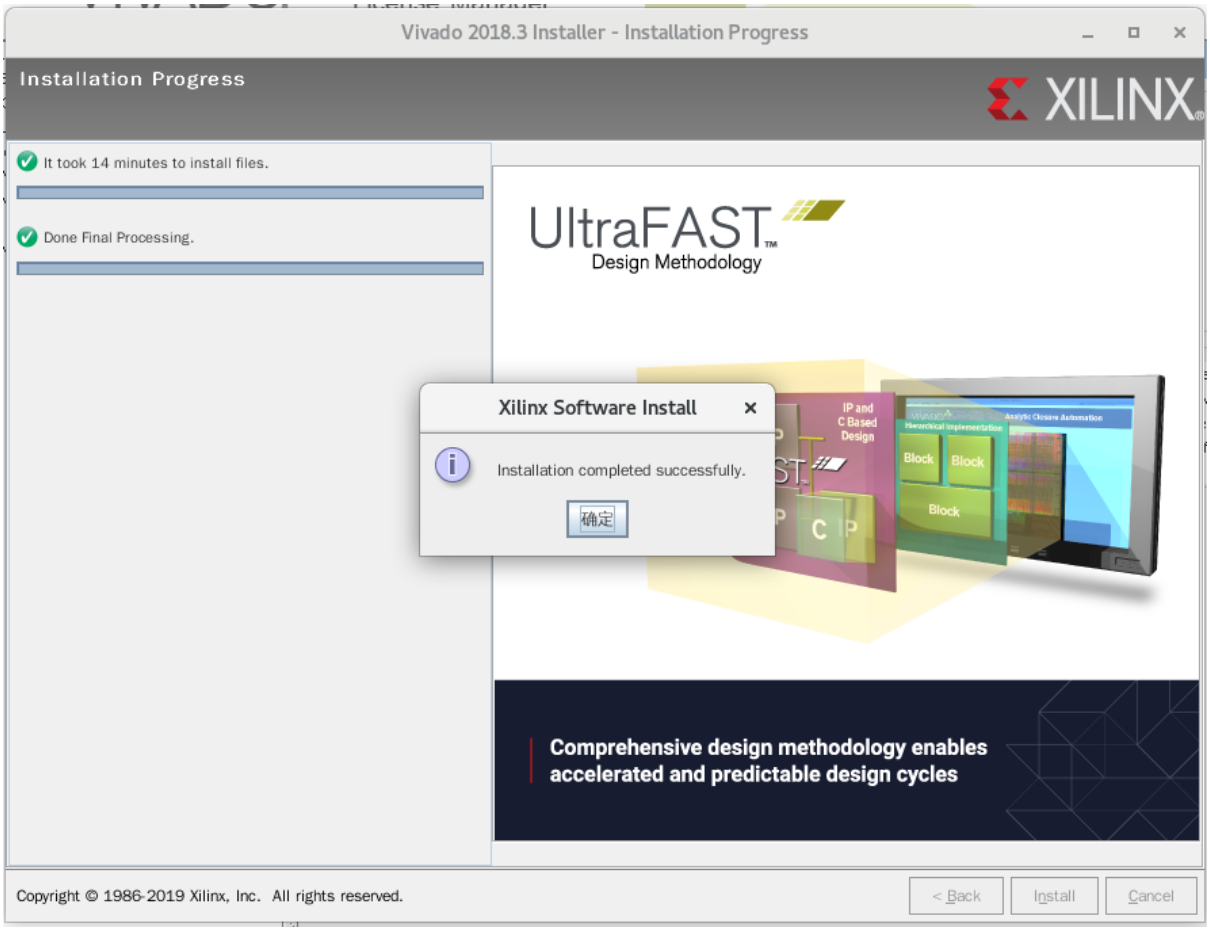
Click “Next” directly to enter the next step



Select the installation directory, here I choose to install to “/home/wuhongyi/Xilinx”, and then click “Next” to enter the next step



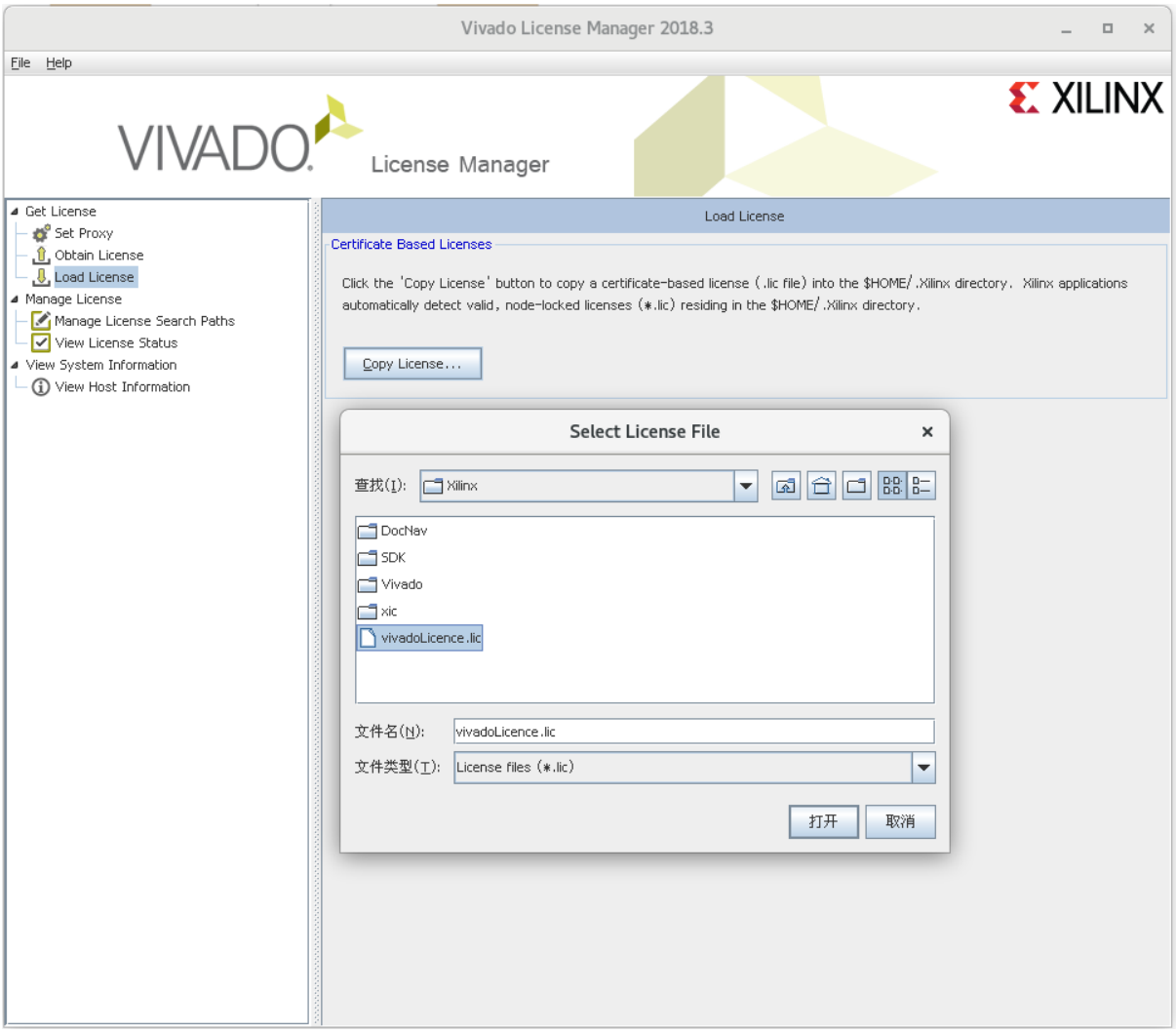
Wait for the installation to complete



The following two steps are not necessary.

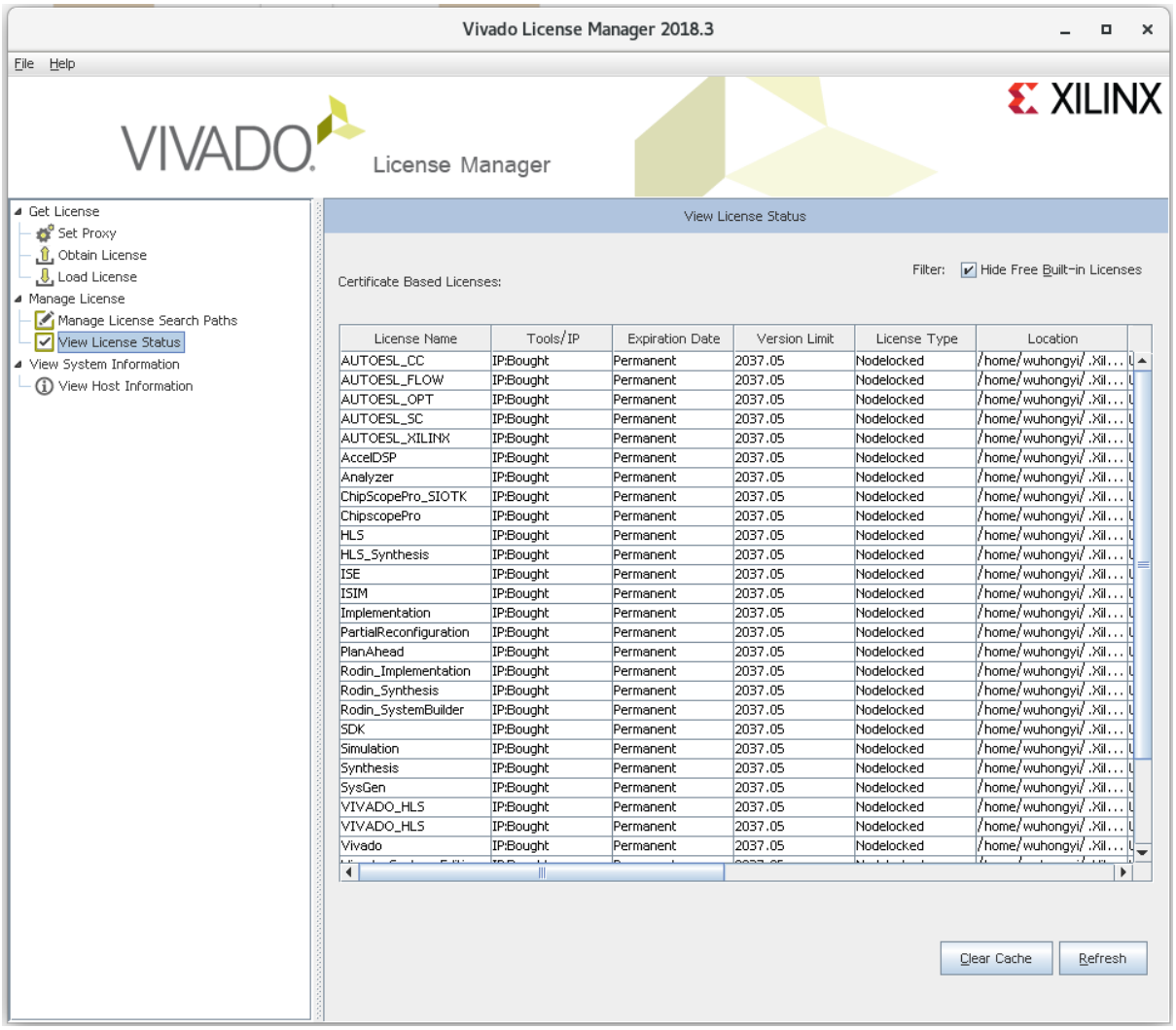
Copy the “vivadoLicence.lic” file to the installation directory, here is “/home/wuhongyi/Xilinx”

After the installation is complete, the following interface will pop up



Click on the “Load License” in the upper left and select our “vivadoLicence.lic” file

Then click “View License Status” in the upper left to view the authorized IP core



5.2 Compile

When you open it for the first time, you need to clear the `P16_MZTIO_FW_0p01/build` folder.

- Open Vivado. Use Tools > Run Tcl Script to run project generating script ⋯/verilog/xillydemo-vivado.tcl. The resulting project file is in ⋯verilogvivado
- There have been cases where the script crashes Vivado, and then the compile has ~100 pin property critical warnings. In such cases, start over.
- Compile demo project (generate bitstream). Ignore warnings and critical warnings.
- Check build/xillydemo.runs/impl_1/xillydemo.bit

5.3 In system debug

Is possible???

CHAPTER 6

experiment

About multiplicity output in RJ45 in PKU firmware

- when setting multiplicity==0, output high level
- when setting multiplicity>=1, the default output is low level, and it is high when triggered.

When the MSRB bit 6 is 1

- the synchronization indication signal can be obtained
- have the DPMFULL output information
- have back plane FT, VT information

6.1 online monitor

After modifying the parameter configuration file `settings.ini`, you need to run the following program to modify the register settings.

```
./progfippi
```

It should be noted that the program is not allowed to be executed when DAQ running

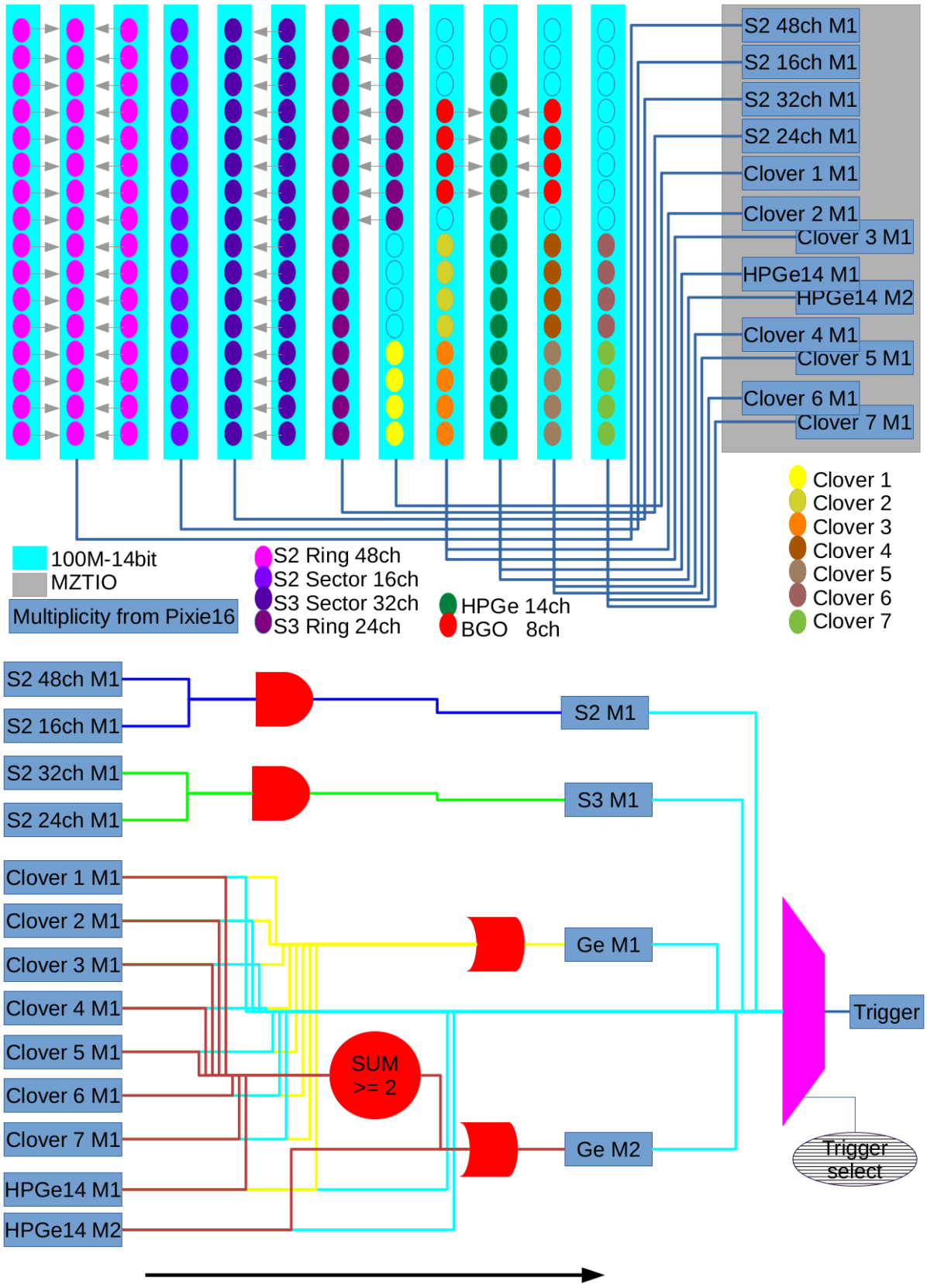
You can view the parameters settings in the web page, and the scaler counter and so on.

6.2 experiment mode

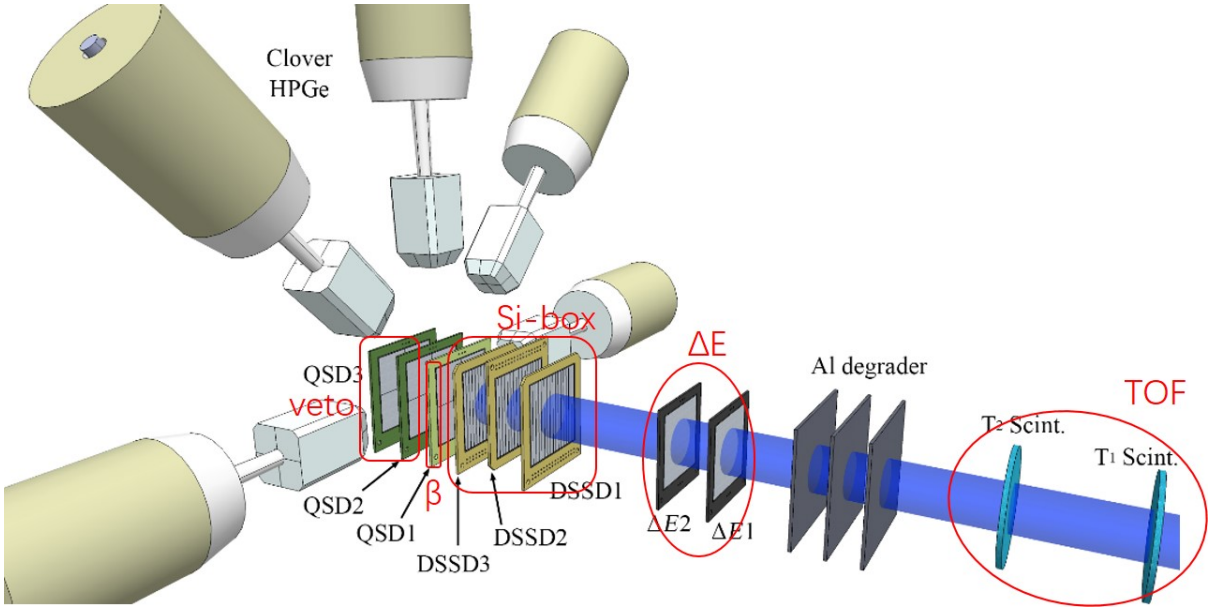
We will provide a common combination of firmware and software for the following four types of experiments.

6.2.1 in beam gamma

designing…


S2 48ch M1
S2 16ch M1
S2 32ch M1
S2 24ch M1
Clover 1 M1
Clover 2 M1
Clover 3 M1
HPGe14 M1
HPGe14 M2
Clover 4 M1
Clover 5 M1
Clover 6 M1
Clover 7 M1
Clover 1
Clover 2
Clover 3
Clover 4
Clover 5
Clover 6
Clover 7
100M-14bit
MZTIO
Multiplicity from Pixie16
S2 Ring 48ch
S2 Sector 16ch
S3 Sector 32ch
S3 Ring 24ch
HPGe 14ch
BGO 8ch
S2 M1
S3 M1
Ge M1
Ge M2
SUM
>= 2
Trigger
Trigger
select

6.2.2 beta decay

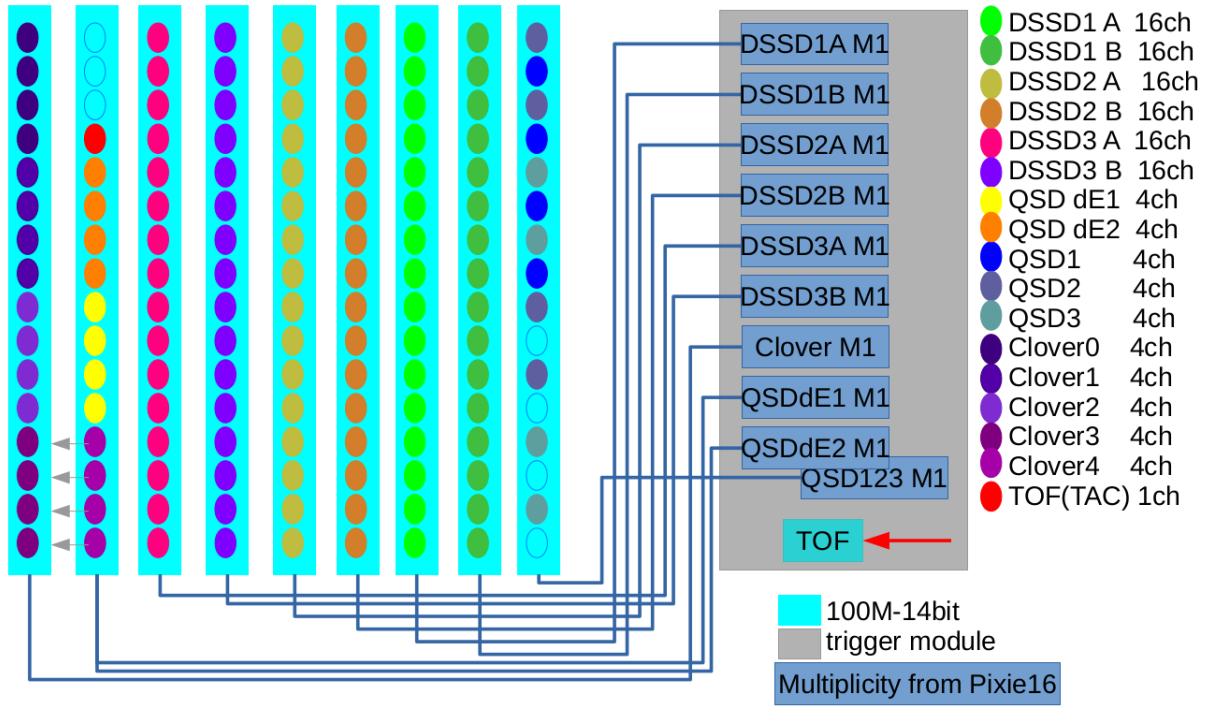


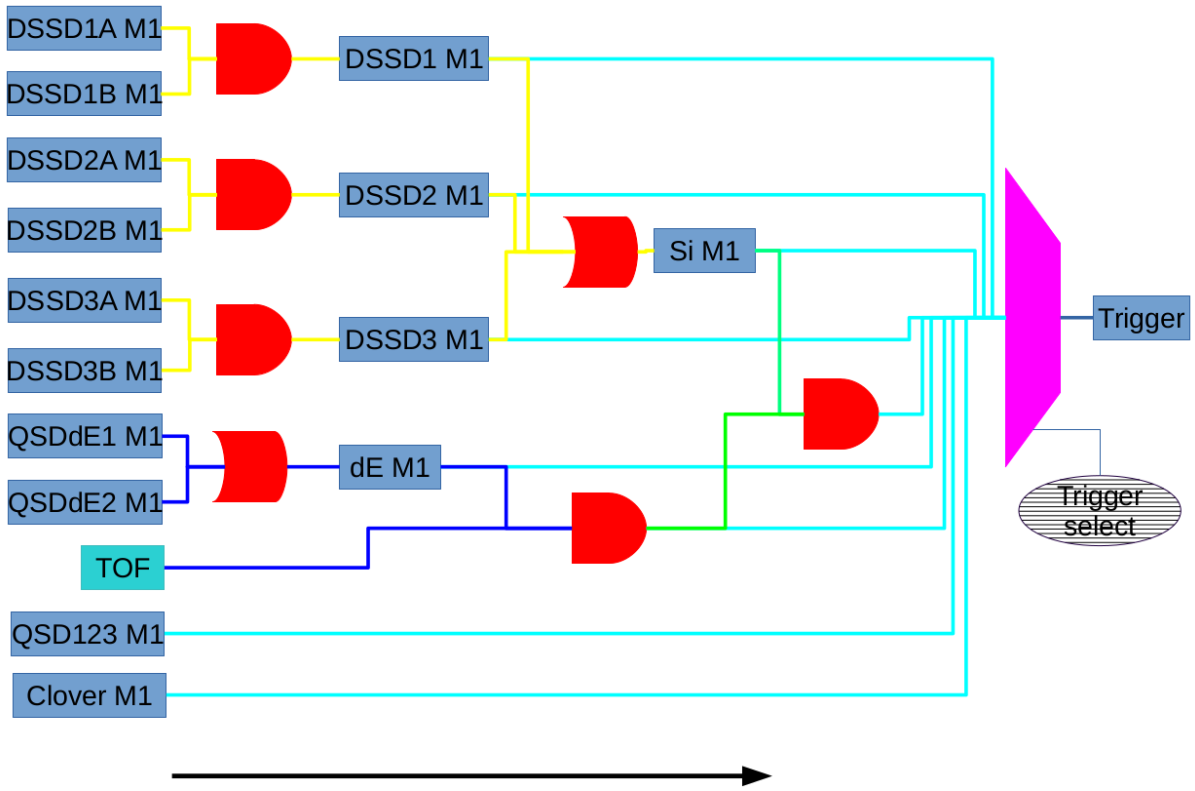
Listed below is the silicon detector information in the detection array:

- **QSDΔE1**
  - MICRON MSQ25, Junction 4, 50.0mm x 50.0mm, 309um
- **QSDΔE2**
  - CIAE Q300, Junction 4, 50.0mm x 50.0mm, 300um
- **DSSD1**
  - MICRON W1, Junction 16, Ohmic 16, 49.5mm x 49.5mm, 142um
- **DSSD2**
  - MICRON W1, Junction 16, Ohmic 16, 49.5mm x 49.5mm, 142um
- **DSSD3**
  - MICRON W1, Junction 16, Ohmic 16, 49.5mm x 49.5mm, 142um
- **QSD1**
  - MICRON MSQ25, Junction 4, 50.0mm x 50.0mm, 1546um
- **QSD2**
  - CIAE Q300, Junction 4, 50.0mm x 50.0mm, 300um
- **QSD3**
  - CIAE Q300, Junction 4, 50.0mm x 50.0mm, 300um

The signals of the plastic scintillator T1 and T2 are converted into pulse amplitude information by TAC, which can be collected using 100MSPS module.

designing…



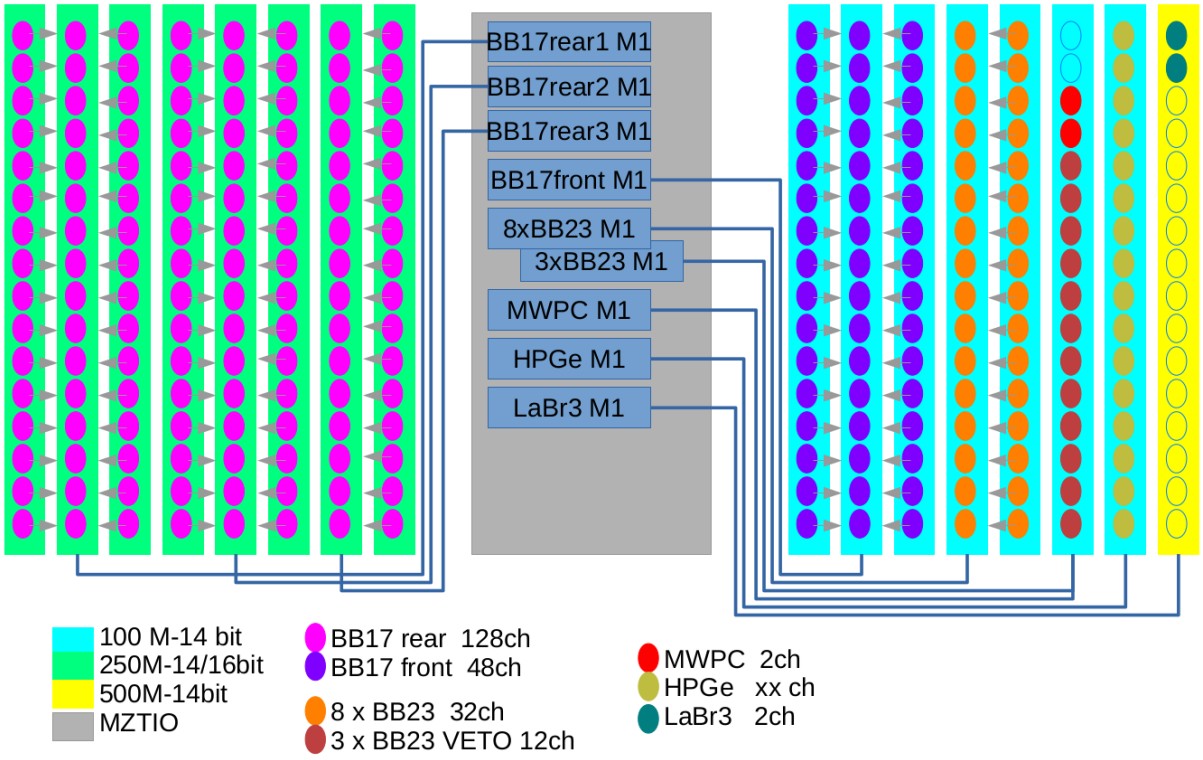


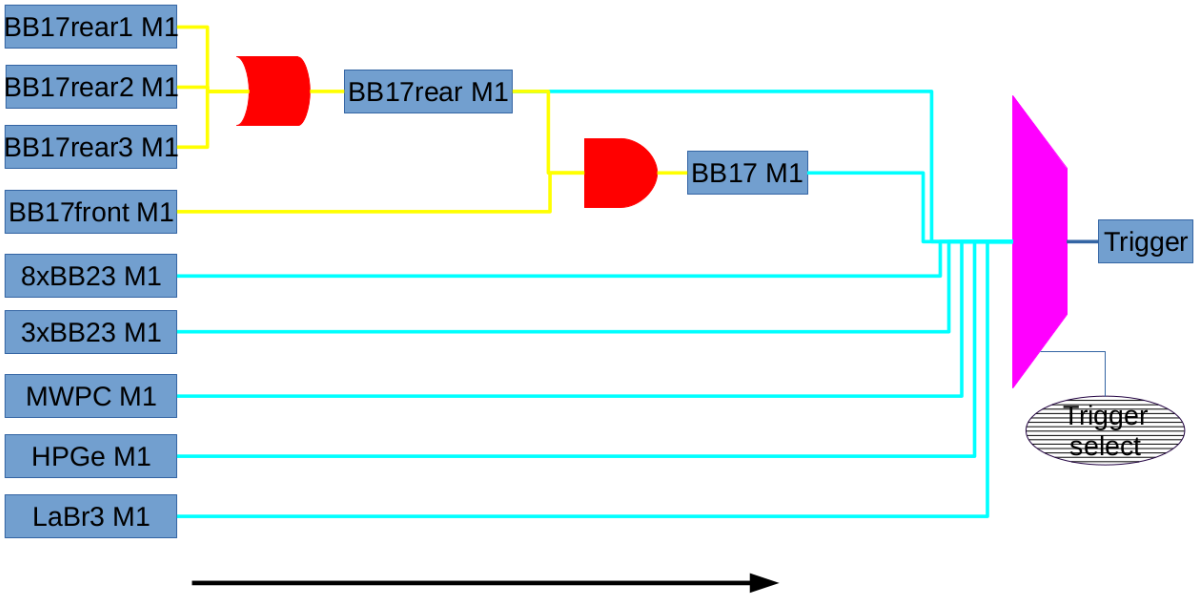
6.2.3 nuclear reaction

designing…

6.2.4 Super heavy nucleus

designing…


BB17rear1 M1
BB17rear2 M1
BB17rear3 M1
BB17front M1
8xBB23 M1
3xBB23 M1
MWPC M1
HPGe M1
LaBr3 M1
100 M-14 bit
250M-14/16bit
500M-14bit
MZTIO
BB17 rear 128ch
BB17 front 48ch
8 x BB23 32ch
3 x BB23 VETO 12ch
MWPC 2ch
HPGe xx ch
LaBr3 2ch


BB17rear1 M1
BB17rear2 M1
BB17rear3 M1
BB17front M1
8xBB23 M1
3xBB23 M1
MWPC M1
HPGe M1
LaBr3 M1
BB17rear M1
BB17 M1
Trigger
Trigger select

CHAPTER 7

Code

7.1 PS code

```
docs    #PKU MZTIO GUIDES
static   # css js
webops

Pixie16_MZTrigIO_Manual.pdf

MZTIOCommon.c
MZTIOCommon.h
MZTIODefs.h
clockprog.c
progfippi.cc
settings.ini
status.c
status.cgi
makefile

pkulogo100.jpg
why.jpg
webopspasswords
index.html
log.html
status.html
support.html
```

7.2 PL code

7.2.1 downscale

```
module downscale
  (
```

(下页继续)

(续上页)

```
   din,
   dout,
   down,
   clk
   );

   parameter DATA_W = 16;
   input [DATA_W-1:0]  down;
   input din;
   output dout;
   reg         dout;
   input clk;
endmodule
```

7.2.2 scaler

```
module scaler
  (
   din,
   dout ,
   endcount,
   clk
   );

   parameter DATA_W = 32;
   output[DATA_W-1:0]  dout;
   reg   [DATA_W-1:0]  dout;

   input din;
   input endcount;
   input clk;
endmodule
```

7.2.3 signaldelay512

```
module signaldelay512
  (
   din,
   dout,
   delay,
   clk
   );

   output dout;
   reg    dout;
   input [9:0] delay;
   input       din;
   input clk;
endmodule
```

7.2.4 signalextend512

```
module signalextend512
  (
   din,
```

(下页继续)

(续上页)

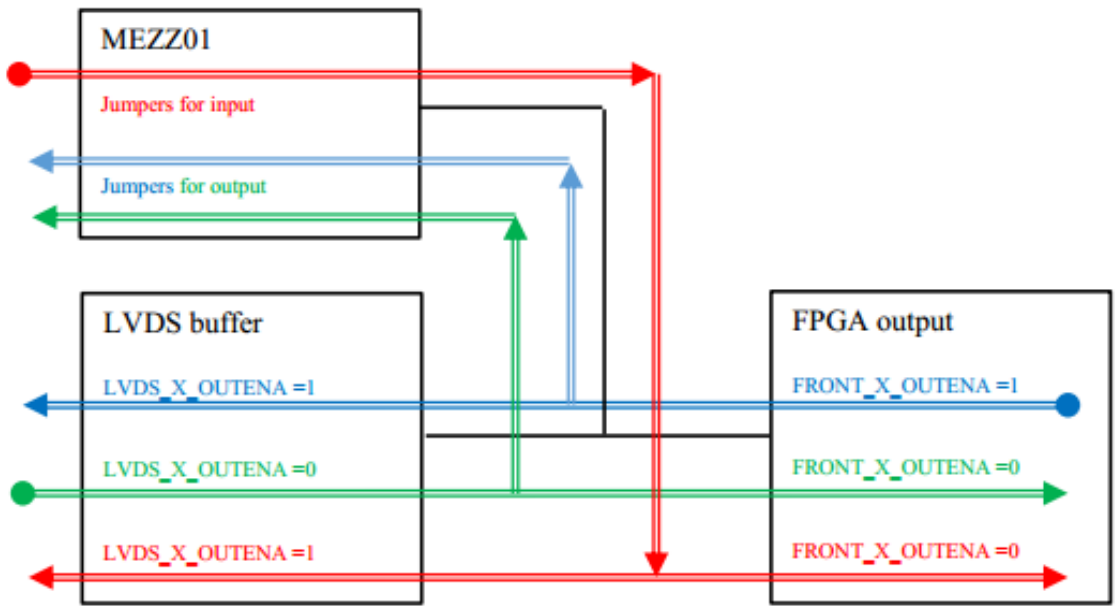
```
    dout,
    extend,
    clk
    );

    input din;
    output dout;
    reg          dout;
    input [9:0] extend;
    input clk;
endmodule
```

7.2.5 IP core

FIFO

```
module fifo_delay512(clk, srst, din, wr_en, rd_en, dout, full, empty,
  data_count)
/* synthesis syn_black_box black_box_pad_pin="clk,srst,din[0:0],wr_en,rd_en,
↪dout[0:0],full,empty,data_count[9:0]" */;
  input clk;
  input srst;
  input [0:0]din;
  input wr_en;
  input rd_en;
  output [0:0]dout;
  output full;
  output empty;
  output [9:0]data_count;
endmodule
```

7.3 xillydemo

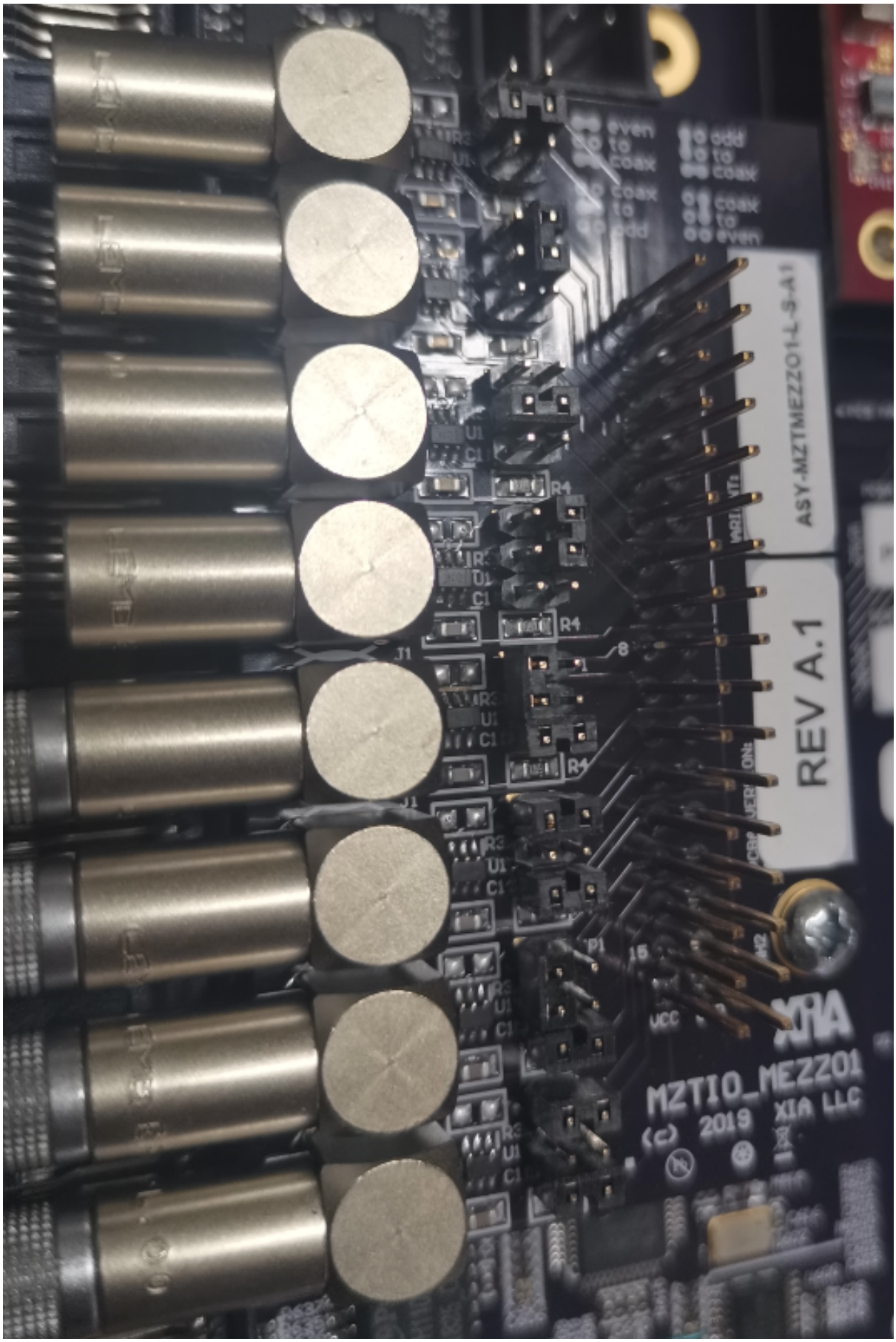


```
// The configuration of the FrontIO_A/B/C is completely flexible. For example, if you connect the RJ-45 of a Pixie-16 to FrontI/O A 0-3 (the upper RJ-45 on the trigger board), signals will connect
// FO5 - Front I/O A 3      FrontIO_Aena==0
// FO1 - Front I/O A 0      FrontIO_Aena==0
// FI5 - Front I/O A 1      FrontIO_Aena==1
// FI1 - Front I/O A 2      FrontIO_Aena==1

// F0  5p/5n  synchronization status / multiplicity result channel 0(pku firmware)
// FO  1p/1n  not used / multiplicity result channel 1(pku firmware)
// FI  5p/5n  external fast trigger
// FI  1p/1n  external validation trigger

// FrontIO_Aout [3] [0]  [7] [4]  [11] [8]  [15] [12]
// FrontIO_Ain  [1] [2]  [5] [6]  [9] [10]  [13] [14]
```

- **FRONT_X_OUTENA**
  - == 1 表示从 MZ 往前面板驱动输出，代码里面操作 out
  - == 0 表示从前面板往 MZ 驱动输入，代码里面操作 in
- **LVDS_X_OUTTENA**
  - == 1 表示驱动网口向外输出
  - == 0 表示驱动网口向里输入

如果 MEZZ01 开启输入模式，则必须设置 `FRONT_X_OUTENA==0 && LVDS_X_OUTTENA==1`，其余模式下，MEZZ01 跳针全部设置成输出模式，此时网口可用于输入或者输出模式。

当前，在前面板 C 口配置一个 MEZZ01 模块，其中前四通道设置为信号输入，分别连接 [1]/[2]/[5]/[6]，后四个通道设置为信号输出，分别连接 [9]/[10]/[13]/[14]。该配置模式下，C 口对应的四个网口仍然可用于多重性的输入，此时参数 FrontIO = 0x6600, LVDSIO = 0x6666。如果不使用 MEZZ01 模块，只连接网口与 P16 模块，则参数 FrontIO/LVDSIO 均设置为 0x6666。


REV A.1
ASY-MZTMEZZ01-L-S-A1
MZTIO_MEZZ01
(c) 2019 XIA LLC
even to coax
odd to coax
coax to odd
coax to even
J1
P1
VCC

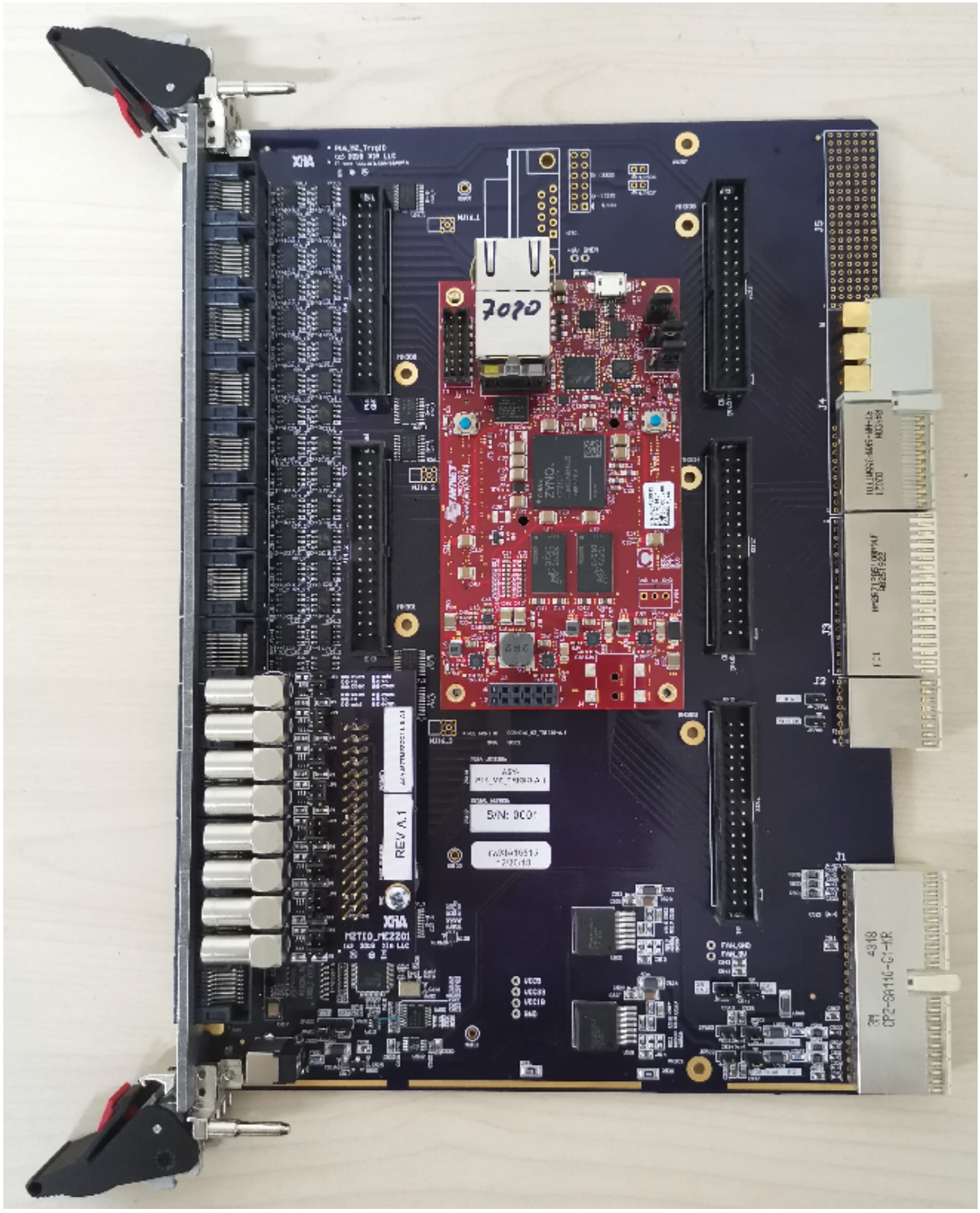
CHAPTER 8

demo version 01

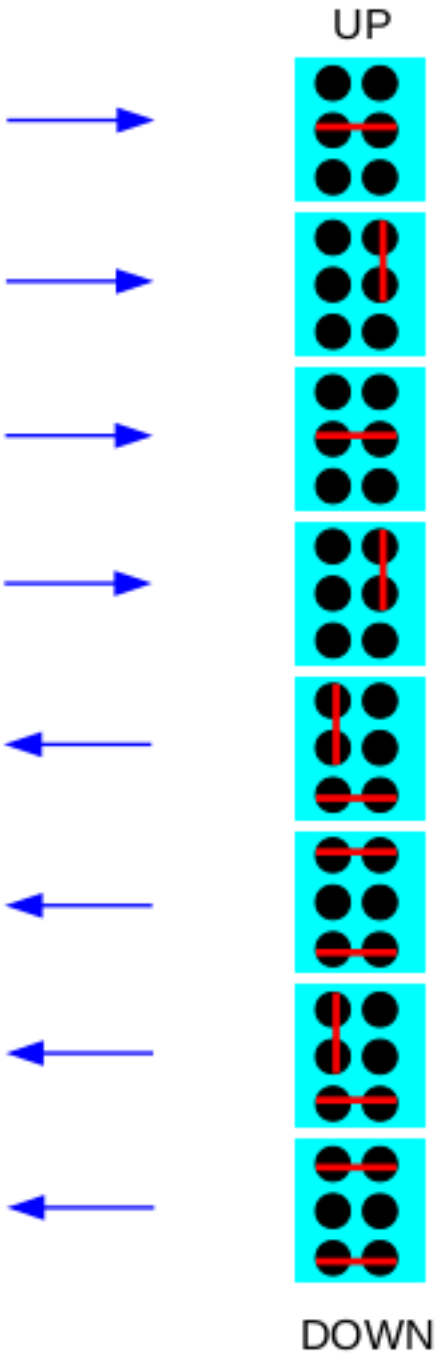
In order to facilitate GDDAQ users to be familiar with the logic functions of Pixie-16 module and the characteristics of PKU firmware, this firmware was specially developed for teaching. Users can download the corresponding version firmware and web control program at https://github.com/wuhongyi/MZTIO/ .

The *version/01* folder contains the firmware *xillydemo.bit* and the control web *www* folder. **This firmware and its supporting control program can only be used for learning, please contact Hongyi Wu get the experimental version.**

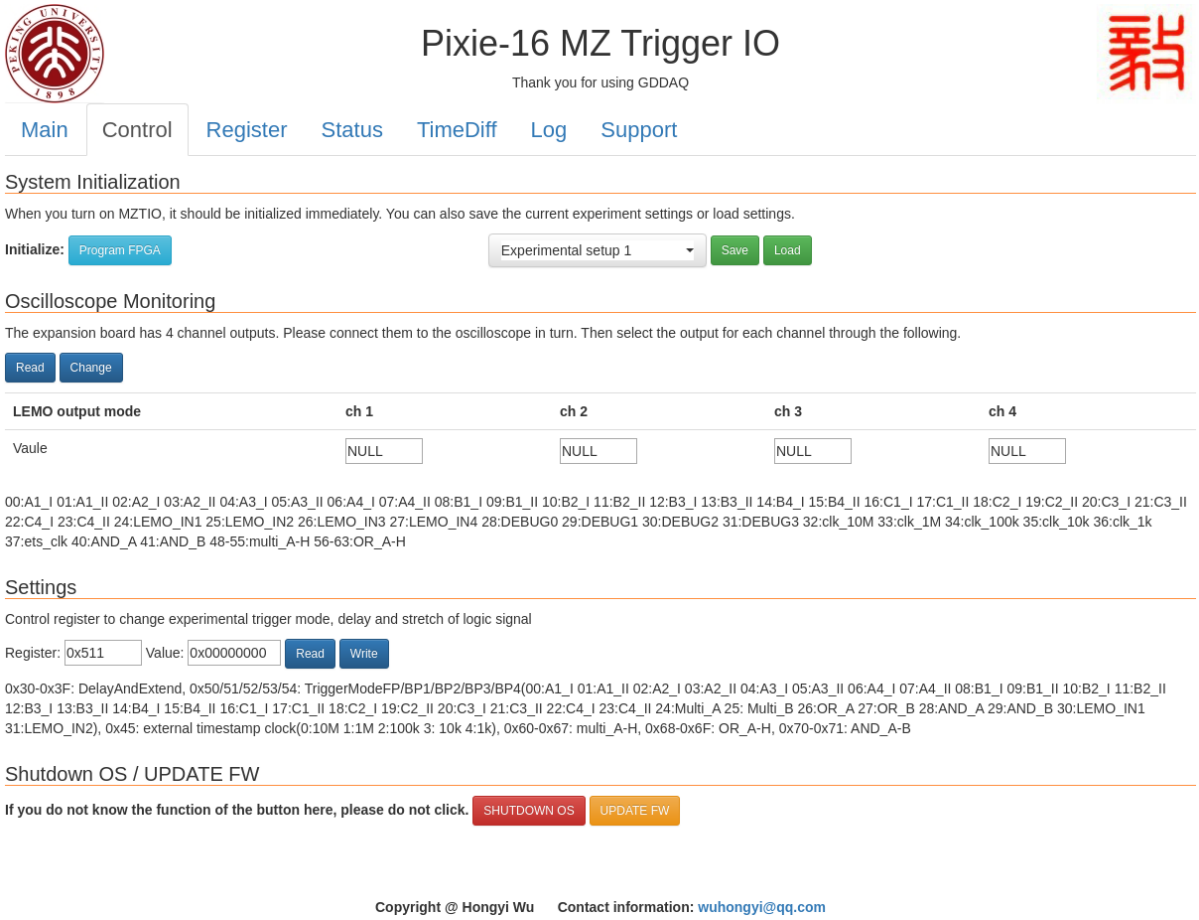
The top 12 RJ-45 connectors on the front panel of the MZTIO are represented by the following symbols from top to bottom: A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4. The RJ-45 connector on the Pixie-16 module in the PKU firmware outputs the multiplicity logic signals of channel 0 and channel 1, which are denoted by _I and _II, respectively. Then A1_I represents the multiplicity logic of channel 0 in the Pixie-16 module connected to the first RJ-45 port of MZTIO.



This firmware requires the use of an 8-channel LEMO port daughter board. Set the first 4 channels as input and the next 4 channels as output by jumper. The jumper settings are shown below:


UP
DOWN

8.1 Control register



The button “Program FPGA” is used to initialize the system configuration. When the operating system is powered on, click this button to complete the system initialization.

Five experimental setup parameters can be saved, which are “Experimental setup 1-5”. After modifying the register to configure the experiment logic, user can click the “Save” button to save, and the current FPGA register parameters will be saved to the selected experiment configuration. The button “Load” is used to load the selected experiment configuration into the FPGA.

The four LEMO input channels are named LEMO1, LEMO2, LEMO3, LEMO4. The LEMO4 is connected to the “run inhibit” signal on the back panel of the chassis. When the input signal is high, the acquisition system stops running.

The monitoring part of the oscilloscope is used to select the output signals of the four LEMO output channels. The following table lists all currently available options. Click the “Read” button to read the current setting parameters. The button “Change” is used to write the parameters of the current input box to the FPGA.

表 1: 4 channels LEMO output

vaule	signal
00	A1_I
01	A1_II
02	A2_I
03	A2_II
04	A3_I
05	A3_II
06	A4_I
07	A4_II

下页继续

表 1 – 续上页

vaule	signal
08	B1_I
09	B1_II
10	B2_I
11	B2_II
12	B3_I
13	B3_II
14	B4_I
15	B4_II
16	C1_I
17	C1_II
18	C2_I
19	C2_II
20	C3_I
21	C3_II
22	C4_I
23	C4_II
24	LEMO input 1
25	LEMO input 2
26	LEMO input 3
27	LEMO input 4
28	DEBUG0
29	DEBUG1
30	DEBUG2
31	DEBUG3
32	10M clock
33	1M clock
34	100k clock
35	10k clock
36	1k clock
37	ets clock
40	AND_A
41	AND_B
48	multi_A
49	multi_B
50	multi_C
51	multi_D
52	multi_E
53	multi_F
54	multi_G
55	multi_H
56	OR_A
57	OR_B
58	OR_C
59	OR_D
60	OR_E
61	OR_F
62	OR_G
63	OR_H

The register setting part is used to read or modify register setting parameters. When reading the register, user need to enter the address of the register to be read, and then click the button “Read”; when modifying the register, input the address and parameter value of the register to be modified, and then click the button “Write”.

表 2: control register

vaule	function
0x30	DelayAndExtend1(not used now [15:0]delay [31:16]stretch)
0x31	DelayAndExtend2(not used now [15:0]delay [31:16]stretch)
0x32	DelayAndExtend3(not used now [15:0]delay [31:16]stretch)
0x33	DelayAndExtend4(not used now [15:0]delay [31:16]stretch)
0x34	DelayAndExtend5(not used now [15:0]delay [31:16]stretch)
0x35	DelayAndExtend6(not used now [15:0]delay [31:16]stretch)
0x36	DelayAndExtend7(not used now [15:0]delay [31:16]stretch)
0x37	DelayAndExtend8(not used now [15:0]delay [31:16]stretch)
0x38	DelayAndExtend9(not used now [15:0]delay [31:16]stretch)
0x39	DelayAndExtend10(not used now [15:0]delay [31:16]stretch)
0x3A	DelayAndExtend11(not used now [15:0]delay [31:16]stretch)
0x3B	DelayAndExtend12(not used now [15:0]delay [31:16]stretch)
0x3C	DelayAndExtend13(not used now [15:0]delay [31:16]stretch)
0x3D	DelayAndExtend14(not used now [15:0]delay [31:16]stretch)
0x3E	DelayAndExtend15(not used now [15:0]delay [31:16]stretch)
0x3F	DelayAndExtend16(not used now [15:0]delay [31:16]stretch)
0x45	external timestamp clock(0:10M 1:1M 2:100k 3: 10k 4:1k)
0x50	TriggerModeFP(00:A1_I 01:A1_II 02:A2_I 03:A2_II 04:A3_I 05:A3_II 06:A4_I 07:A4_II 08:B1_I 09:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II)
0x51	TriggerModeBP1(not used now)
0x52	TriggerModeBP2(not used now)
0x53	TriggerModeBP3(not used now)
0x54	TriggerModeBP4(not used now)
0x60	multi_A([23:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II [31:24] multi)
0x61	multi_B([23:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II [31:24] multi)
0x62	multi_C([23:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II [31:24] multi)
0x63	multi_D([23:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II [31:24] multi)
0x64	multi_E([23:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II [31:24] multi)
0x65	multi_F([23:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II [31:24] multi)
0x66	multi_G([23:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II [31:24] multi)
0x67	multi_H([23:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II [31:24] multi)
0x68	OR_A([31:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II 24:multi_A 25:multi_B 26:multi_C 27:multi_D 28:multi_E 29:multi_F 30:multi_G 31:multi_H)

下页继续

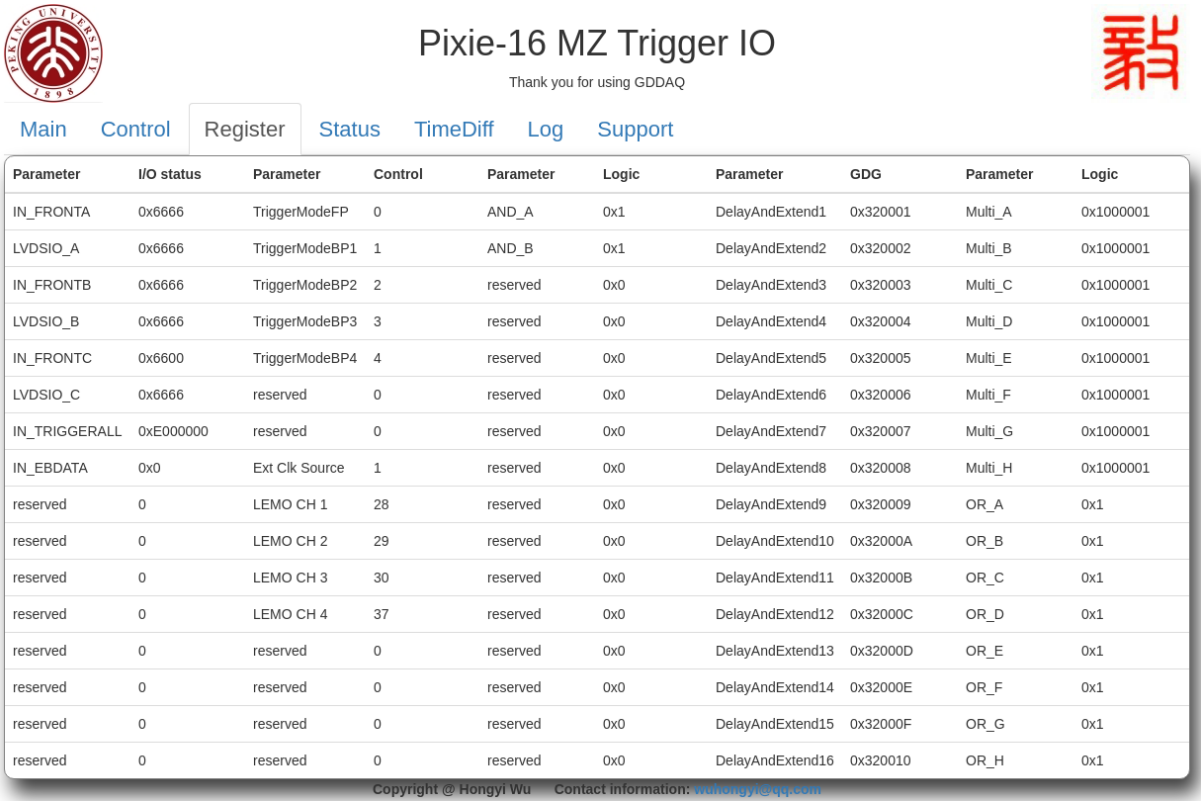
表 2 – 续上页

vaule	function
0x69	OR_B([31:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II 24:multi_A 25:multi_B 26:multi_C 27:multi_D 28:multi_E 29:multi_F 30:multi_G 31:multi_H)
0x6A	OR_C([31:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II 24:multi_A 25:multi_B 26:multi_C 27:multi_D 28:multi_E 29:multi_F 30:multi_G 31:multi_H)
0x6B	OR_D([31:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II 24:multi_A 25:multi_B 26:multi_C 27:multi_D 28:multi_E 29:multi_F 30:multi_G 31:multi_H)
0x6C	OR_E([31:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II 24:multi_A 25:multi_B 26:multi_C 27:multi_D 28:multi_E 29:multi_F 30:multi_G 31:multi_H)
0x6D	OR_F([31:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II 24:multi_A 25:multi_B 26:multi_C 27:multi_D 28:multi_E 29:multi_F 30:multi_G 31:multi_H)
0x6E	OR_G([31:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II 24:multi_A 25:multi_B 26:multi_C 27:multi_D 28:multi_E 29:multi_F 30:multi_G 31:multi_H)
0x6F	OR_H([31:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II 24:multi_A 25:multi_B 26:multi_C 27:multi_D 28:multi_E 29:multi_F 30:multi_G 31:multi_H)
0x70	AND_A([31:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II 24:OR_A 25:OR_B 26:OR_C 27:OR_D 28:OR_E 29:OR_F 30:OR_G 31:OR_H)
0x71	AND_B([31:0] bit mask 0:A1_I 1:A1_II 2:A2_I 3:A2_II 4:A3_I 5:A3_II 6:A4_I 7:A4_II 8:B1_I 9:B1_II 10:B2_I 11:B2_II 12:B3_I 13:B3_II 14:B4_I 15:B4_II 16:C1_I 17:C1_II 18:C2_I 19:C2_II 20:C3_I 21:C3_II 22:C4_I 23:C4_II 24:OR_A 25:OR_B 26:OR_C 27:OR_D 28:OR_E 29:OR_F 30:OR_G 31:OR_H)

The webpage can also be used to shut down the LINUX OS in MZTIO. Clicking the red button "SHUTDOWN OS" will pop up a password input box. After entering the correct password, the OS will be shut down immediately. After that, user will not be able to access the webpage. User need to power on again to start the operating system before use. This button is only used to shut down the MZTIO operating system before shutting down the chassis.

The yellow button "UPDATE FW" is used to upgrade the firmware and restart the operating system. The firmware to be upgraded needs to be placed in the */root* directory, and then click the button. If the firmware upgrade is successful, the web page will prompt that the operating system will restart after one minute, if the upgrade fails, it prompts that the firmware file cannot be found.

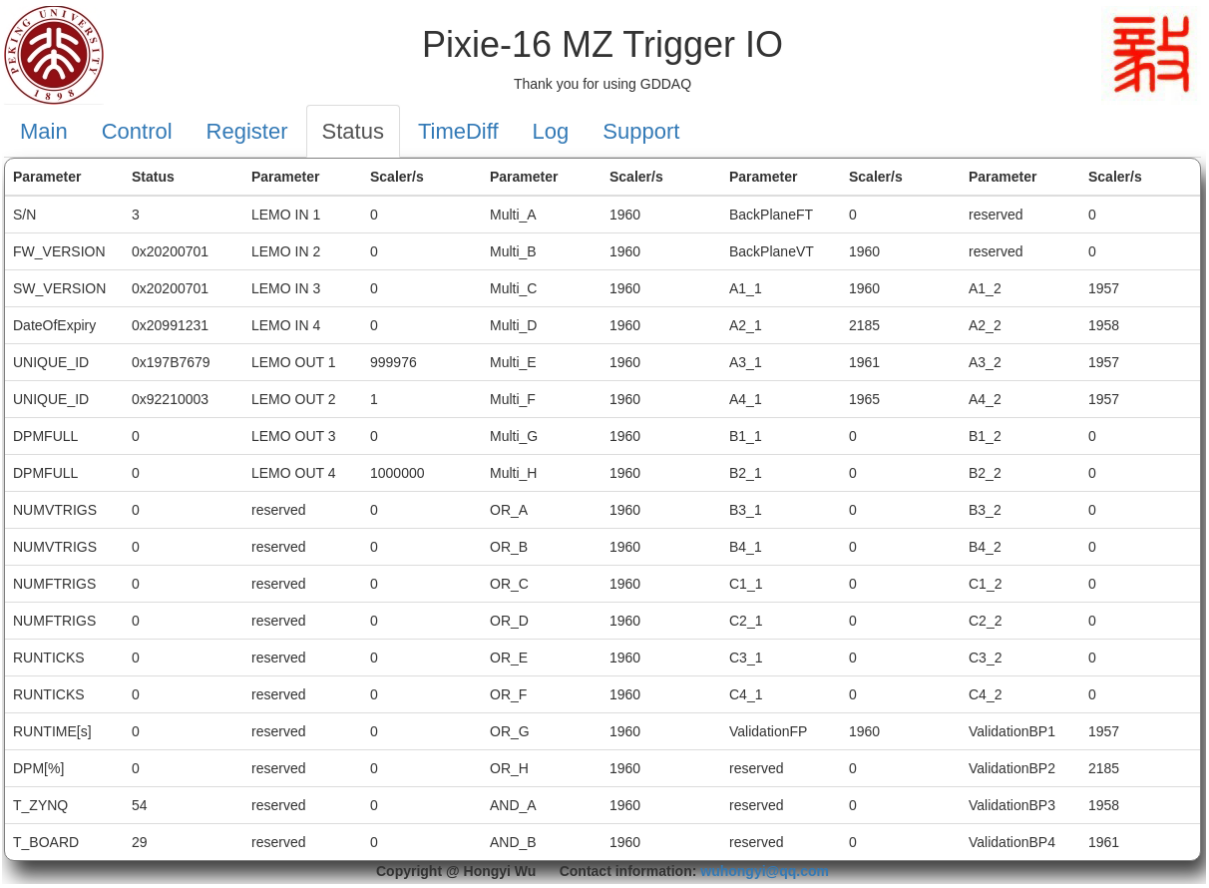
8.2 Register status



Parameter	I/O status	Parameter	Control	Parameter	Logic	Parameter	GDG	Parameter	Logic
IN_FRONTA	0x6666	TriggerModeFP	0	AND_A	0x1	DelayAndExtend1	0x320001	Multi_A	0x1000001
LVDSIO_A	0x6666	TriggerModeBP1	1	AND_B	0x1	DelayAndExtend2	0x320002	Multi_B	0x1000001
IN_FRONTB	0x6666	TriggerModeBP2	2	reserved	0x0	DelayAndExtend3	0x320003	Multi_C	0x1000001
LVDSIO_B	0x6666	TriggerModeBP3	3	reserved	0x0	DelayAndExtend4	0x320004	Multi_D	0x1000001
IN_FRONTC	0x6600	TriggerModeBP4	4	reserved	0x0	DelayAndExtend5	0x320005	Multi_E	0x1000001
LVDSIO_C	0x6666	reserved	0	reserved	0x0	DelayAndExtend6	0x320006	Multi_F	0x1000001
IN_TRIGGERALL	0xE000000	reserved	0	reserved	0x0	DelayAndExtend7	0x320007	Multi_G	0x1000001
IN_EBDATA	0x0	Ext Clk Source	1	reserved	0x0	DelayAndExtend8	0x320008	Multi_H	0x1000001
reserved	0	LEMO CH 1	28	reserved	0x0	DelayAndExtend9	0x320009	OR_A	0x1
reserved	0	LEMO CH 2	29	reserved	0x0	DelayAndExtend10	0x32000A	OR_B	0x1
reserved	0	LEMO CH 3	30	reserved	0x0	DelayAndExtend11	0x32000B	OR_C	0x1
reserved	0	LEMO CH 4	37	reserved	0x0	DelayAndExtend12	0x32000C	OR_D	0x1
reserved	0	reserved	0	reserved	0x0	DelayAndExtend13	0x32000D	OR_E	0x1
reserved	0	reserved	0	reserved	0x0	DelayAndExtend14	0x32000E	OR_F	0x1
reserved	0	reserved	0	reserved	0x0	DelayAndExtend15	0x32000F	OR_G	0x1
reserved	0	reserved	0	reserved	0x0	DelayAndExtend16	0x320010	OR_H	0x1

This page is used to view the user setting register parameters.

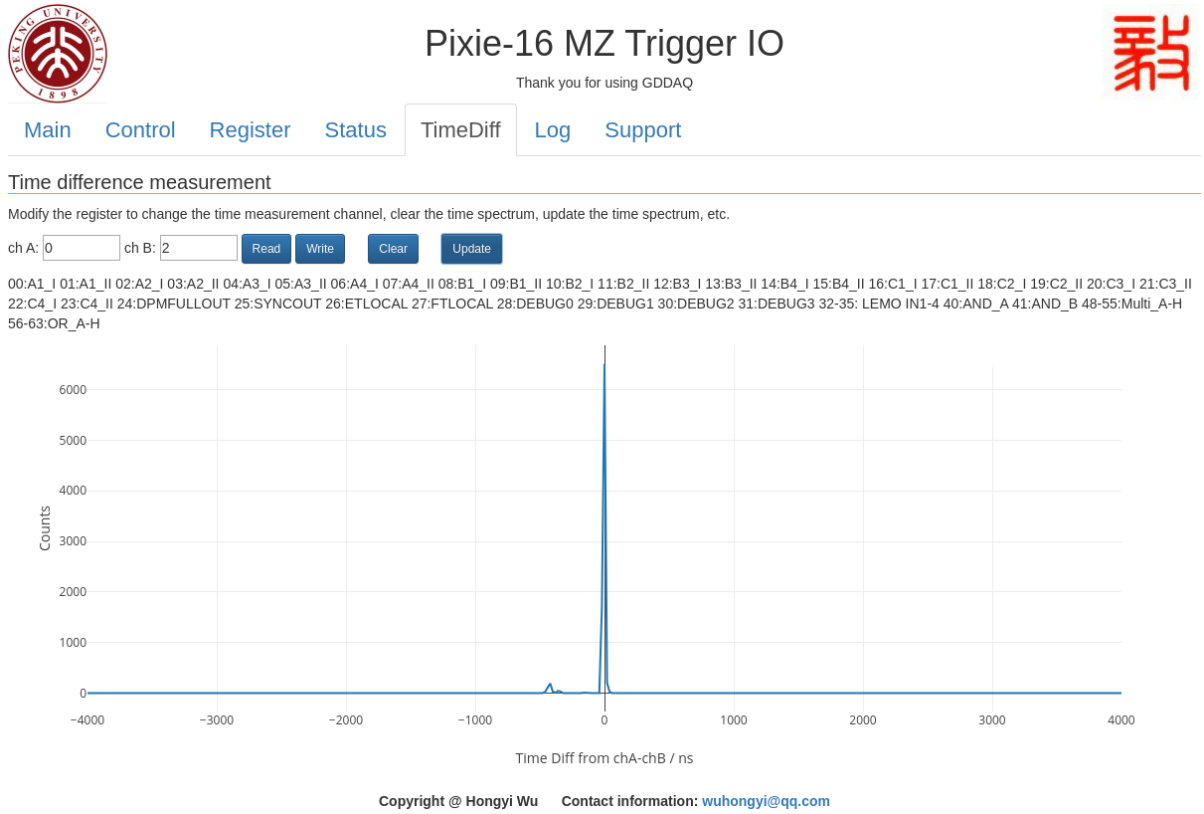
8.3 Trigger rate



Parameter	Status	Parameter	Scaler/s	Parameter	Scaler/s	Parameter	Scaler/s	Parameter	Scaler/s
S/N	3	LEMO IN 1	0	Multi_A	1960	BackPlaneFT	0	reserved	0
FW_VERSION	0x20200701	LEMO IN 2	0	Multi_B	1960	BackPlaneVT	1960	reserved	0
SW_VERSION	0x20200701	LEMO IN 3	0	Multi_C	1960	A1_1	1960	A1_2	1957
DateOfExpiry	0x20991231	LEMO IN 4	0	Multi_D	1960	A2_1	2185	A2_2	1958
UNIQUE_ID	0x197B7679	LEMO OUT 1	999976	Multi_E	1960	A3_1	1961	A3_2	1957
UNIQUE_ID	0x92210003	LEMO OUT 2	1	Multi_F	1960	A4_1	1965	A4_2	1957
DPMFULL	0	LEMO OUT 3	0	Multi_G	1960	B1_1	0	B1_2	0
DPMFULL	0	LEMO OUT 4	1000000	Multi_H	1960	B2_1	0	B2_2	0
NUMVTRIGS	0	reserved	0	OR_A	1960	B3_1	0	B3_2	0
NUMVTRIGS	0	reserved	0	OR_B	1960	B4_1	0	B4_2	0
NUMFTRIGS	0	reserved	0	OR_C	1960	C1_1	0	C1_2	0
NUMFTRIGS	0	reserved	0	OR_D	1960	C2_1	0	C2_2	0
RUNTICKS	0	reserved	0	OR_E	1960	C3_1	0	C3_2	0
RUNTICKS	0	reserved	0	OR_F	1960	C4_1	0	C4_2	0
RUNTIME[s]	0	reserved	0	OR_G	1960	ValidationFP	1960	ValidationBP1	1957
DPM[%]	0	reserved	0	OR_H	1960	reserved	0	ValidationBP2	2185
T_ZYNQ	54	reserved	0	AND_A	1960	reserved	0	ValidationBP3	1958
T_BOARD	29	reserved	0	AND_B	1960	reserved	0	ValidationBP4	1961

This page is used for real-time count rate monitoring. The current version includes the count rate of 4 LEMO input channels, 4 LEMO output channels, Multi_A-H, OR_A-H, and 12 RJ-45 connectors input (Pixie-16 output multiplicity).

8.4 Time difference measurement



This page implements the time difference measurement of any two logic signals (chA-chB, a time difference greater than 0 means that the chA signal is later than the chB signal). The button “Read” is used to read the signal source parameters; the button “Write” is used to change the signal source; the button “Clear” is used to clear the time difference spectrum in the FPGA. When the signal source is changed, the time difference spectrum in the FPGA must be cleared. The button “Update” can be used to read the current time difference spectrum from the FPGA and display it on the web page.

表 3: time difference meaurement sources

vaule	signal
00	A1_I
01	A1_II
02	A2_I
03	A2_II
04	A3_I
05	A3_II
06	A4_I
07	A4_II
08	B1_I
09	B1_II
10	B2_I
11	B2_II
12	B3_I
13	B3_II
14	B4_I
15	B4_II
16	C1_I
17	C1_II

下页继续

表 3 – 续上页

vaule	signal
18	C2_I
19	C2_II
20	C3_I
21	C3_II
22	C4_I
23	C4_II
24	DPMFULLOUT
25	SYNCOUT
26	ETLOCAL
27	FTLOCAL
28	DEBUG0
29	DEBUG1
30	DEBUG2
31	DEBUG3
32	LEMO input 1
33	LEMO input 2
34	LEMO input 3
35	LEMO input 4
40	AND_A
41	AND_B
48	multi_A
49	multi_B
50	multi_C
51	multi_D
52	multi_E
53	multi_F
54	multi_G
55	multi_H
56	OR_A
57	OR_B
58	OR_C
59	OR_D
60	OR_E
61	OR_F
62	OR_G
63	OR_H