MZTIO

发布 *0.0*

Hongyi Wu(吴鸿毅)

2020 年 07 月 01 日

Contents:

CHAPTER 1

README

如果您需要固件，请联系吴鸿毅 (wuhongyi@qq.com)

如果您想了解 PKU 如何使用 MZTIO，请点击以下链接: PKUMZTIO

XIA SUPPORT: XIA Blog

Pixie-16 MZ-TrigIO 设计用于将信号从背板（后连接器）连接到前面板（前连接器），并在 FPGA 架构中实现逻辑组合。它具有以下功能和特性：

- 用于 Pixie-16 的以太网可编程触发/符合控制模块
- 48+ Pixie-16 背板触发连接到本地 Zynq 处理器
- 48 个前面板 LVDS 连接到本地 Zynq 处理器
- 带嵌入式 Linux 的 MicroZed Zynq 处理器，作为独立 PC，内置 SD 卡驱动器，USB 主机，10/100 以太网，网络服务器等
- 1588 PTP 和 SyncE 时钟同步
- 开源用户访问软件和固件
- 用作独立桌面设备或 6U PXI 机箱
- 通过子卡自定义 I/O 标准

1.1 安全须知

请花点时间查看这些安全预防措施。它们既可以保护您，也可以防止损坏 Pixie 模块和连接的设备。此安全信息适用于所有操作员和维修人员。

- 电源
  - Pixie-16 MZ-TrigIO 模块通过 AC/DC 适配器或 PXI 背板供电。默认适配器具有适用于不同地区的各种 AC 插头附件。
  - 在从 Pixie-16 MZ-TrigIO 拔下电源插头或关闭 PXI 机箱电源之前，请记得关闭 Linux 操作系统。
- 用户调整/反汇编

  - 为避免人身伤害和/或损坏，在进入模块内部之前，请务必断开电源 。有一些与有经验的用户可能想要使用的电路板上的时钟相关的跳线。

- 电压额定值
  - 输入和输出信号不得超过 ±3.3V。在进行任何连接之前，请查看附录中的引脚分配。
- 子卡
  - 子卡可用作前面板和背面输入的替代品，这需要小心避免 FPGA 输出和标准连接器输入的冲突。
- 维修和清洁
  - 为避免人身伤害和/或损坏 Pixie 模块或连接的设备，请勿尝试修理或清洁这些设备的内部。
- Linux 密码
  - Pixie-16 MZ-TrigIO Linux 操作系统附带默认用户 ID 和密码，用于 1）SSH 登录，2）SMB 文件共享，以及 3）Web 操作，如下所述。用户应立即更改这些密码，尤其是当 Pixie-16 MZ-TrigIO 连接到外部网络时。不要让黑客接管你的 Pixie-16 MZ-TrigIO！
- Linux 备份
  - Pixie-16 MZ-TrigIO Linux OS 存储在可移动 SD 卡上。SD 卡的文件系统可能会损坏，这会使 Linux 系统崩溃并使 Pixie-16 MZ-TrigIO 无法运行。因此，建议定期备份 SD 卡，例如使用 Win32DiskImager。（需要一个字节一个字节的复制）。
  - 请注意，所有 Linux 密码都存储在 SD 卡上。

1.2 逻辑编程

为了适应中低能实验核物理的需求，我们发展了以下基本功能：

- 信号延迟
- 信号展宽
- 符合
- 多重性选择
- scaler 计数器
- down scale 分除
- 远程参数调节
- 。。。。。。

CHAPTER 2

网页控制界面

2.1 寄存器

用户可以通过修改 settings.ini 文件中的控制寄存器来轻松调整实验逻辑。
当然，对于不同类型的实验，我们有专门的软件，有关特定的寄存器控制方法，请参阅实验手册。

settings.ini - Hongyi Wu @ Peking University (于 PixieNet)

File Edit Options Buffers Tools Conf Help

```
 1 0x000   0                         CSR[15:0]                          (R)
 2 0x001   0                         VERSION                            (R)
 3 0x002   0                         D18[2:0]                           (W/R)
 4 0x003   0                         outblock[1:0]                      (W/R)
 5 0x00A   0                         numtrig                            (R)
 6 0x00B   0                         numtrig                            (R)
 7 0x00C   0                         runticks                           (R)
 8 0x00D   0                         runticks                           (R)
 9 0x100   0x6666                    FrontIO_Aena                       (W/R)
10 0x105   0x6666                    LVDSIO_Aena                        (W/R)
11 0x101   0x6666                    FrontIO_Bena                       (W/R)
12 0x106   0x6666                    LVDSIO_Bena                        (W/R)
13 0x102   0x6600                    FrontIO_Cena                       (W/R)
14 0x107   0x6666                    LVDSIO_Cena                        (W/R)
15 0x103   0x00000000                TriggerAllena                      (W/R)
16 0x104   0x0000                    EB_Dataena                         (W/R)
17 0x108   0xFFFF                    frontA_coincidence_mask            (W/R)
18 0x109   0xFFFF                    frontB_coincidence_mask            (W/R)
19 0x10A   0xFFFF                    frontC_coincidence_mask            (W/R)
20 0x10B   0xFFFFFFFF                TriggerAll_coincidence_mask        (W/R)
21 0x10C   0xFFFF                    EB_Data_coincidence_mask           (W/R)
22 0x110   0xFFFF                    frontA_multiplicity_mask           (W/R)
23 0x111   0xFFFF                    frontB_multiplicity_mask           (W/R)
24 0x112   0xFFFF                    frontC_multiplicity_mask           (W/R)
25 0x113   0xFFFFFFFF                TriggerAll_multiplicity_mask       (W/R)
26 0x114   0xFFFF                    EB_Data_multiplicity_mask          (W/R)
27 0x118   0x0000                    frontA_coincidence_pattern         (W/R)
28 0x119   0x0000                    frontB_coincidence_pattern         (W/R)
29 0x11A   0x0000                    frontC_coincidence_pattern         (W/R)
30 0x11B   0x00000000                TriggerAll_coincidence_pattern     (W/R)
31 0x11C   0x0000                    EB_Data_coincidence_pattern        (W/R)
32 0x120   2                         frontA_multiplicity_threshold      (W/R)
33 0x121   2                         frontB_multiplicity_threshold      (W/R)
34 0x122   2                         frontC_multiplicity_threshold      (W/R)
35 0x123   2                         TriggerAll_multiplicity_threshold  (W/R)
36 0x124   2                         EB_Data_multiplicity_threshold     (W/R)
37 0x128   0                         frontA_output_select               (W/R)
38 0x129   0                         frontB_output_select               (W/R)
39 0x12A   0                         frontC_output_select               (W/R)
40 0x12B   0                         TriggerAll_output_select           (W/R)
41 0x12C   0                         EB_Data_output_select              (W/R)
42 0x030   0x00320028                DelayAndExtend1                    (W/R)
43 0x031   0x000A                    DownScale1                         (W/R)
44 0x040   0                         LEMO output mode                   (W/R)
```

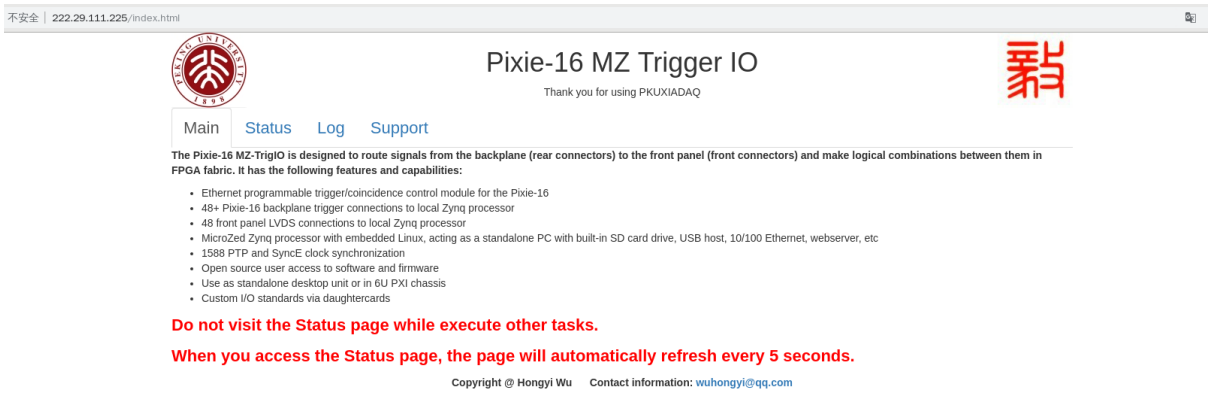
1 -:--- settings.ini All (1,0) (Conf[Space]) 07:49 0.20

Package assoc is obsolete!

2.2 网页

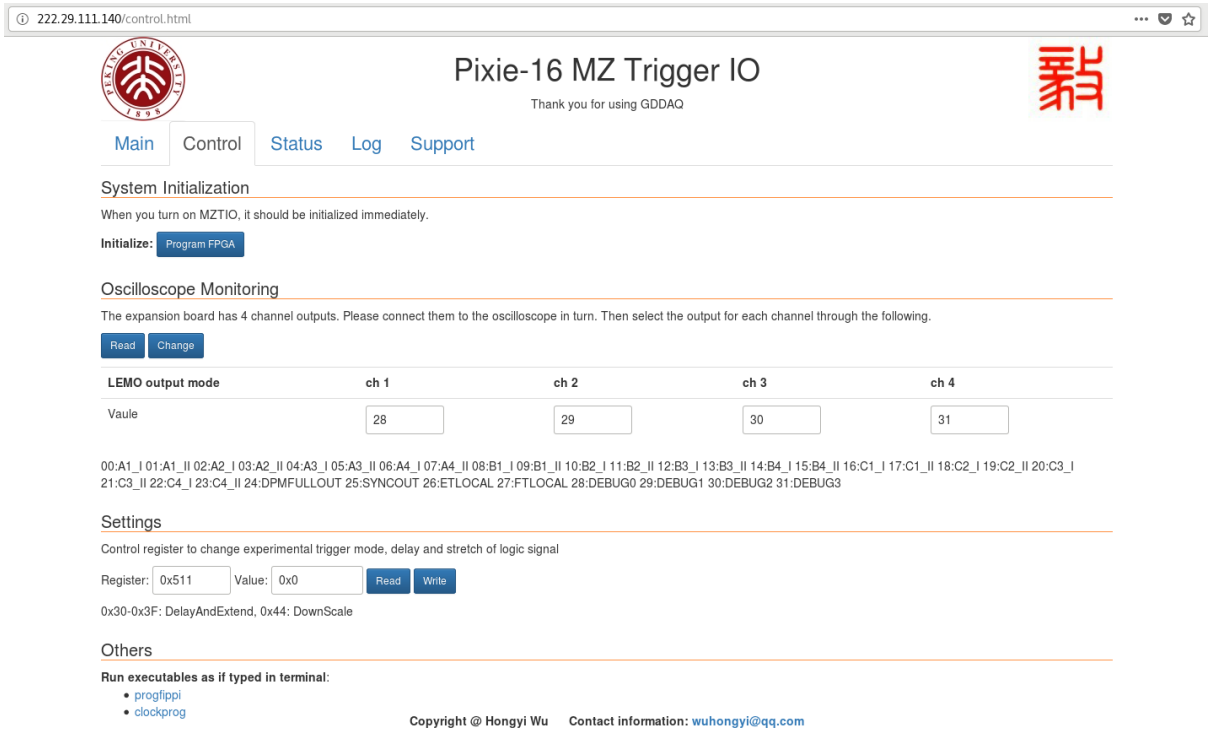
2.2.1 main page

主页，它将提供该模块的基本信息和注意事项。



2.2.2 control page

通过控制寄存器来改变实验触发模式，逻辑信号的延迟与展宽等。

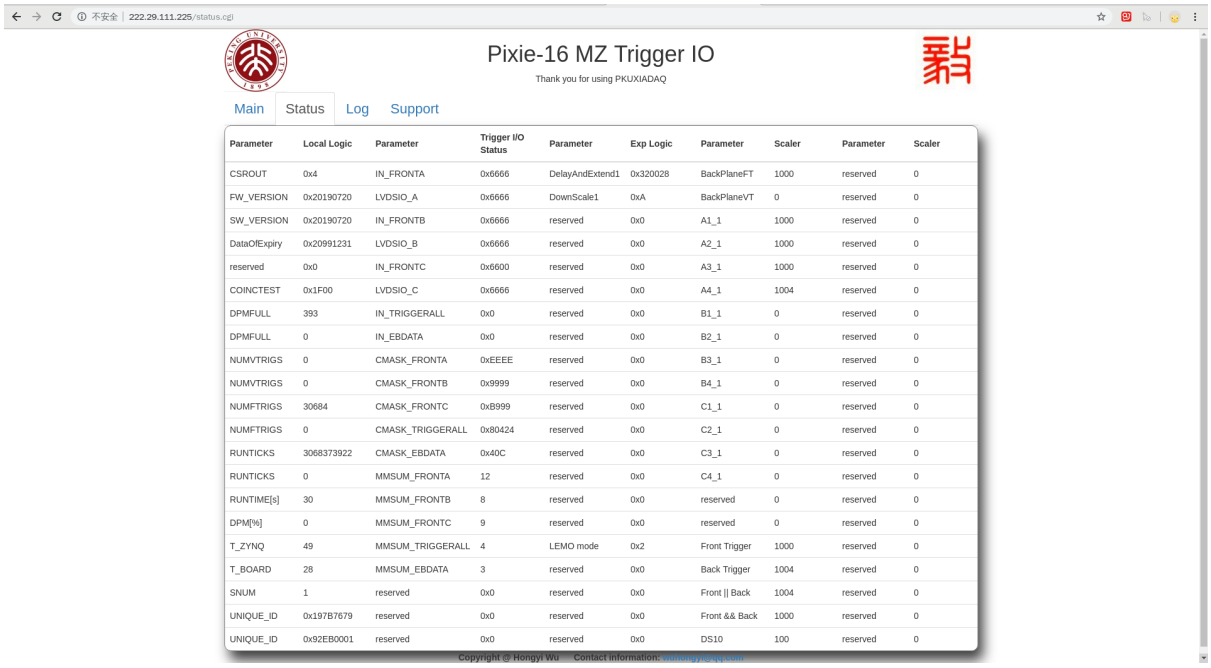


2.2.3 status page

当您访问状态页面时，该页面将每 5 秒钟自动刷新一次。

当前，此页面上有五列可监视参数。

- 第一列的第四行表示允许使用该固体的日期。
- 第一列的第十五行指示当前 DAQ 的运行时间。
- 第一列第 16 行代表 DPMFULL 和总运行时间的百分比。



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

在开发中，此页面将保存状态参数并读取历史参数。

2.2.5 support page

该页面提供了一些基本说明，包括 XIA 说明，PKU 说明等。

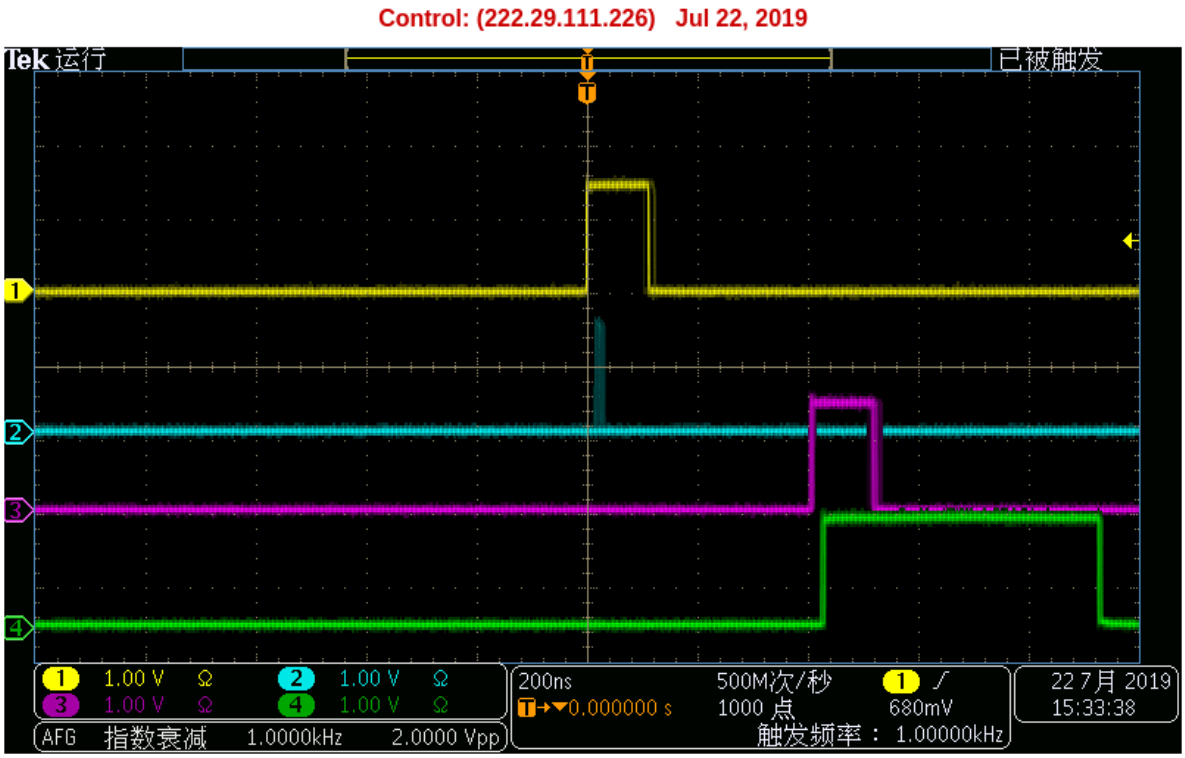
2.3 示波器监视

通过 MZTIO 子板将信号输出到示波器。

大多数示波器只有 4 个通道，因此我们的监视器设置默认设置为 4 个通道。如果要同时监视 8 个通道，则可以用 2 台示波器完成。

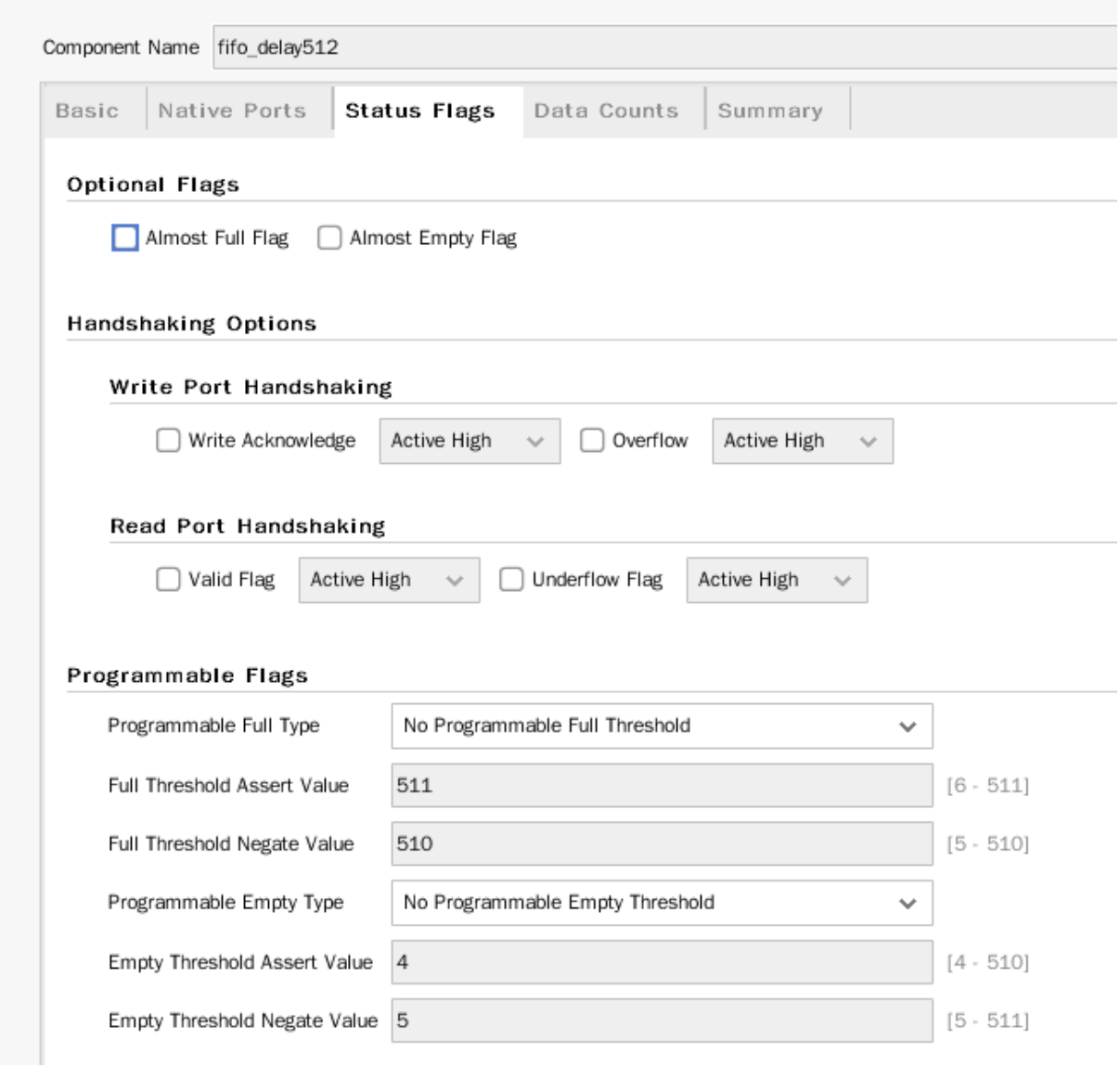
当然，可以通过修改控制寄存器来切换监视信号。有关如何监视不同信号的说明，请阅读特定实验的说明。

下图是示波器监视的示例。线 1 表示触发逻辑信号，线 2 表示触发逻辑信号 10 倍分除之后的结果，线 3 表示线 1 延迟 400 ns 后的信号，线 4 代表线 3 展宽到 500 ns。



2.4 FIFO IP 核的限制

下图显示了 FIFO IP 核参数的可设置范围。



由于 FIFO IP 核的限制，延迟设置为最少 4 个时钟。

CHAPTER 3

远程控制

3.1 minicom

将 USB 线连接电脑，获取系统 IP

在 linux 中可以采用串口通讯软件 minicom

```
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         |
+---------------------------+
```

- 选择 *Serial port setup*，修改 Serial Device 为 */dev/ttyUSB0* 。Bps/Par/Bits 修改为 *115200 8N1* ，底端最后两个选项为 *NO*
- 选择 *Modem and dialing* ，删除 A, B, K 条的内容
- 再然后，选择 *Save setup as dfl* 保存该修改设置
- 最后，选择 *Exit* 来退出配置模式，进入控制模式

```
user: root
password: xia17pxn

密码采用默认的，方便使用者都能登陆
```

假设该模块的 IP 地址为 222.29.111.80，您可以通过以下命令远程登陆。

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

3.2 静态 IP 设置

因为 Ubuntu18.04 采用的是 netplan 来管理 network。所以可以在 /etc/netplan/ 目录下创建一个以 yaml 结尾的文件。比如 01-netplan.yaml 文件。

然后在此文件下写入以下配置 (你需要修改 IP 地址及网关):

```
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]
```

特别要注意的是这里的每一行的空格一定要有的，否则会报错误而设置失败!

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

以上参数为 CIAE 实验使用的配置。

最后使用 *sudo netplan apply* 来重启网络服务就可以了。使用 *ip a* 查看你的静态 IP 是否设置成功了!

CHAPTER 4

ubuntu

4.1 基础配置

4.1.1 ubuntu 18

如果操作系统是当前最新版本，则不需要进行额外的源配置。

使用国内镜像

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic main universe␣
↪multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-updates main␣
↪universe multiverse
```

如果要安装 CERN ROOT，则在 /etc/apt/sources.list 中添加以下行

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

4.1.2 ubuntu 12

如果操作系统版本是之前的老版本，则需要按照以下进行源的修改配置。

编辑源列表文件

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

修改为：

```
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
```

(下页继续)

(续上页)

```
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
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 软件升级

```
apt-get update
```

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

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

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

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

ubuntu 颜色配置，个人目录下放置颜色配置文件.dircolors，该文件在 readhat 系统中文件名为.dir_colors

4.1.4 时区选择

```
# 先查看当前系统时间
date -R
# 查看结果显示的时区，如果与当地时区不一致，则可以通过以下方式进行修改

tzselect
# 下图中展示了中国用户如何修改成当地的时区，其它地区用户进行对应的选择即可
cp /usr/share/zoneinfo/Asia/Shanghai  /etc/localtime

# 查看是否修改成功
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 恢复 SD 卡原始空间

为了加快镜像装载速度，实际上只格式化了 8/16G 左右的 SD 卡空间，我 16/32G 的 SD 卡还有 8/16G 多的空间都没用到，为了能够进行使用进行如下操作

```
fdisk /dev/mmcblk0
# 然后分别输入: d [ENTER],2 [ENTER],n[ENTER] [ENTER],[ENTER],[ENTER],[ENTER],w[ENTER],
 若中间出现问题详细参考 Getting started with Xillinux for Zynq-7000 EPP ， 然后重启
linux 开机后
```

```
# 执行以下命令
resize2fs /dev/mmcblk0p2

# 使用以下命令查看追加的结果
df -h
```

4.3 升级启动文件

要将 SD 卡启动分区挂载到 /mnt/sd 文件夹，请执行

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

这在不删除 SD 卡的情况下更新启动文件很有用。在新的启动文件生效之前，必须重新启动 Pixie-16 MZ-TrigIO。

操作流程如下：

- 在台式机上生成固件文件
- 复制文件到 SD 卡上的文件夹（/var/www）
- 挂载启动分区 /dev/mmcblk0p1 到 /mnt/sd（如果尚未创建 /mnt/sd，则创建该目录）
- 复制文件，例如 cp /var/www/xillydemo.bit /mnt/sd
- 重新启动或关机后再开机（重新启动）

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

4.4 /dev/mmcblk0p1

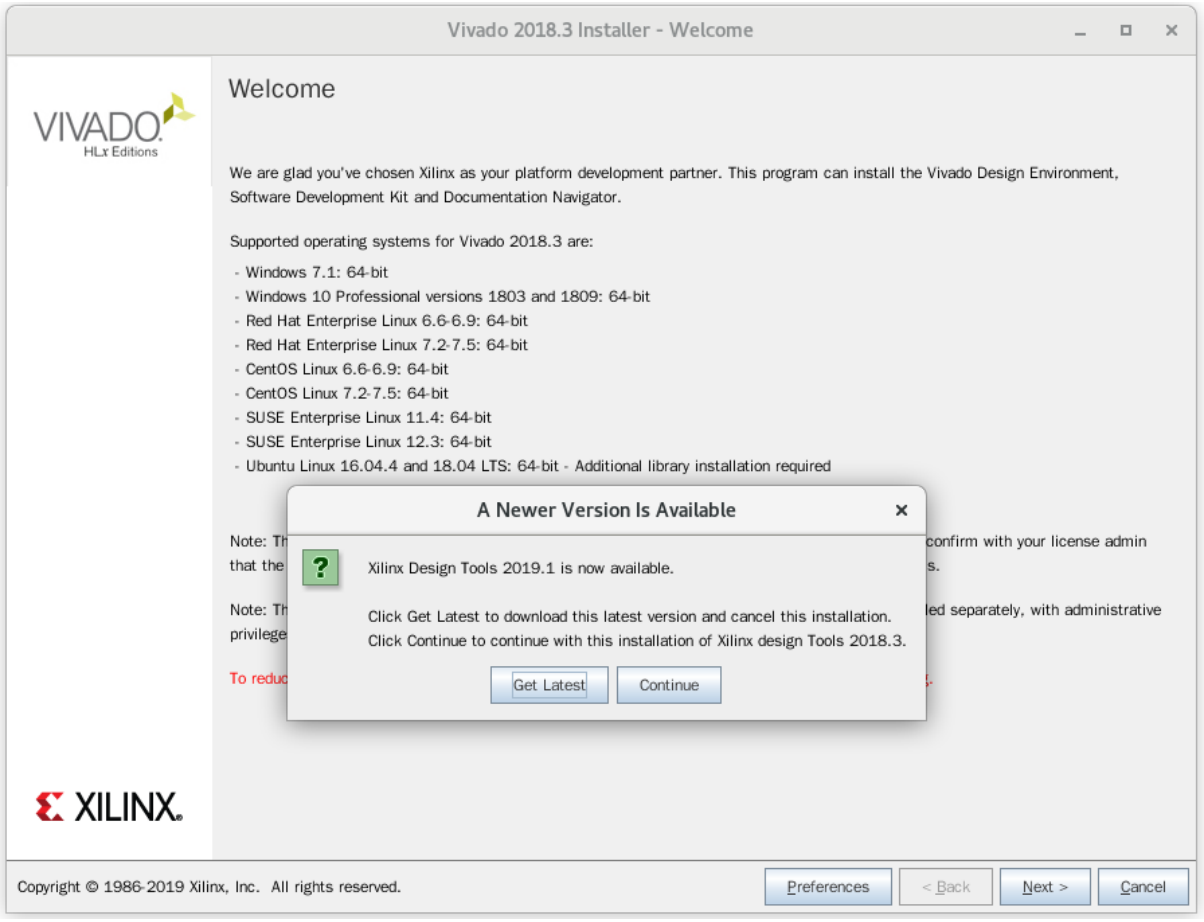
```
boot.bin  devicetree.dtb  uImage  xillydemo.bit
```

CHAPTER 5

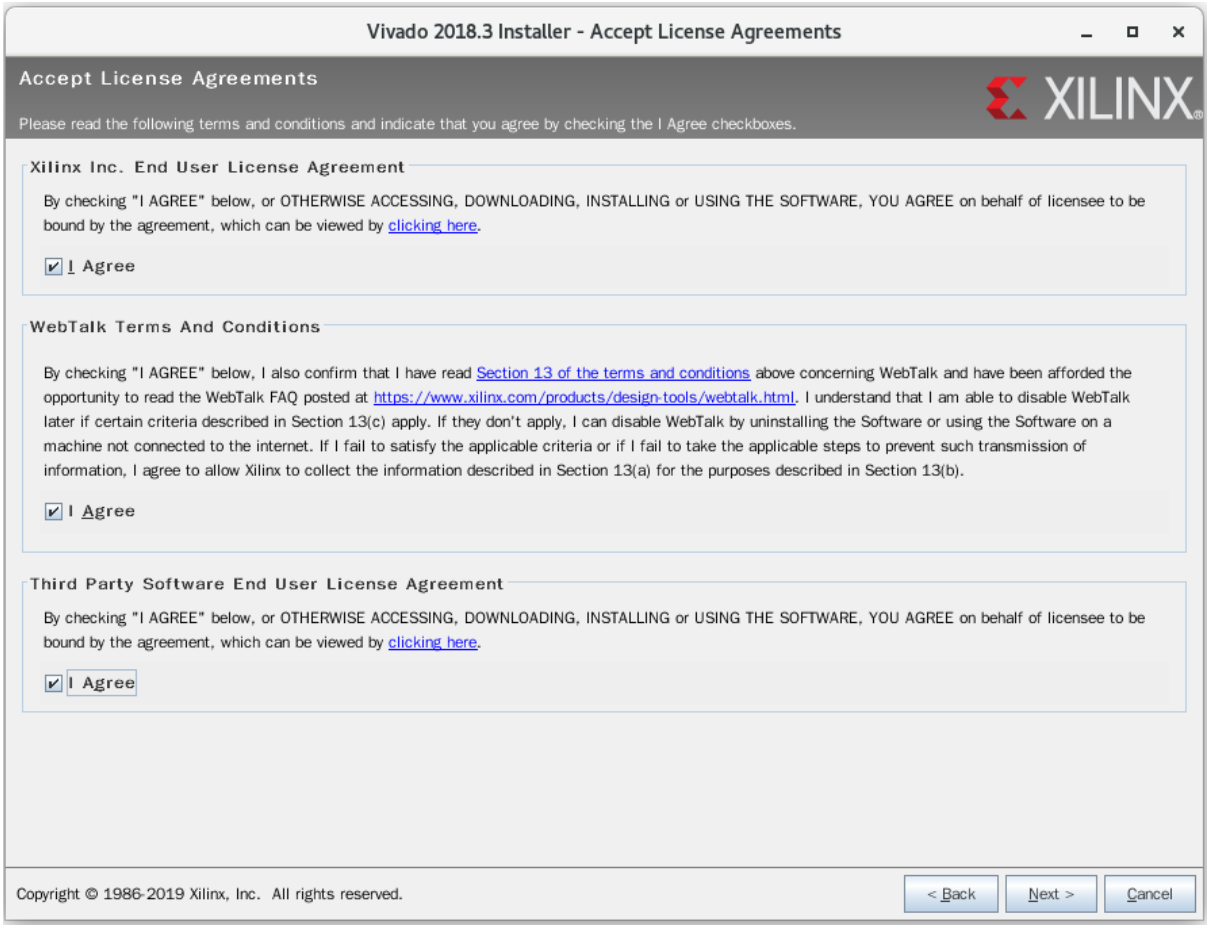
Vivado

5.1 安装

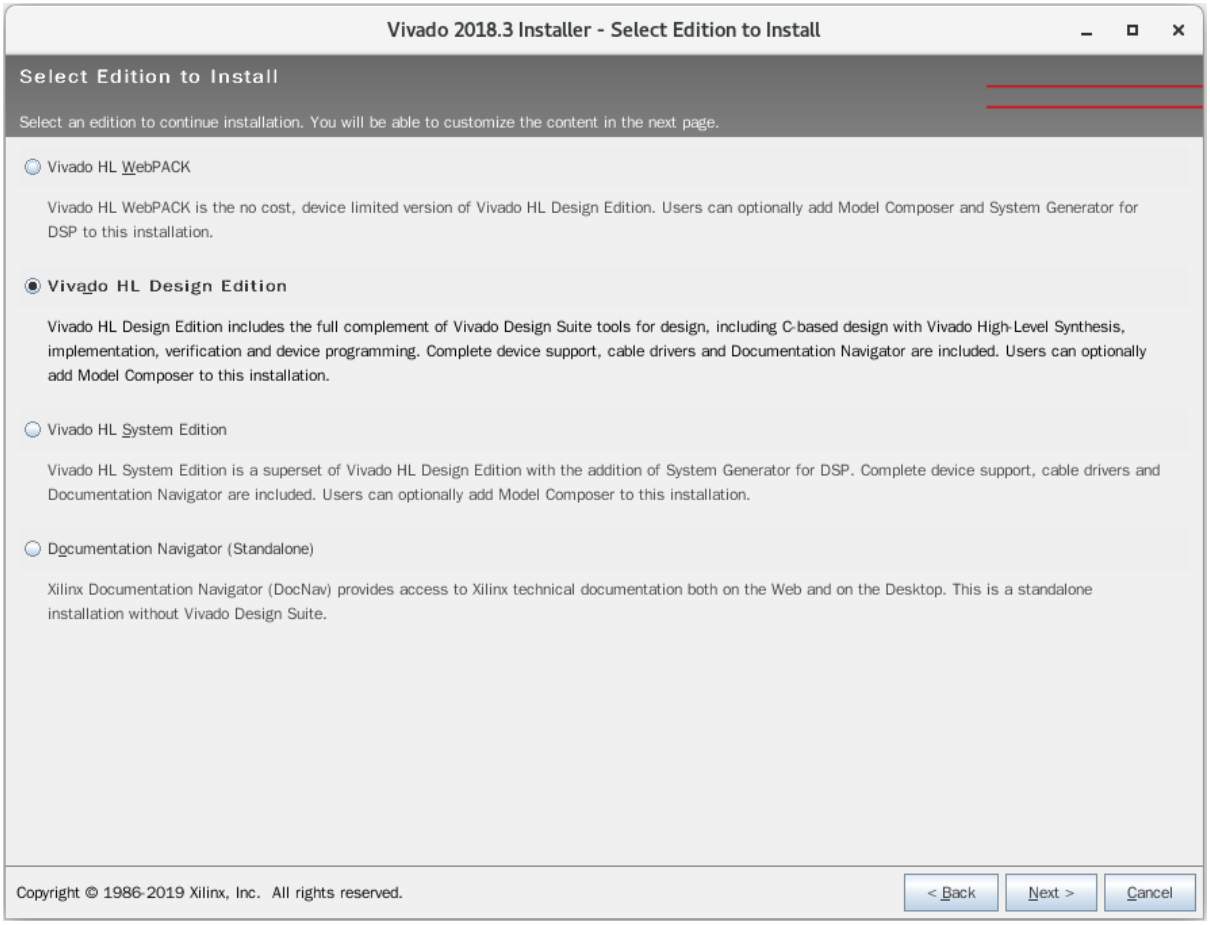
```
tar   -zxvf   Xilinx_Vivado_SDK_2018.3_1207_2324.tar.gz
cd    Xilinx_Vivado_SDK_2018.3_1207_2324
./xsetup
```



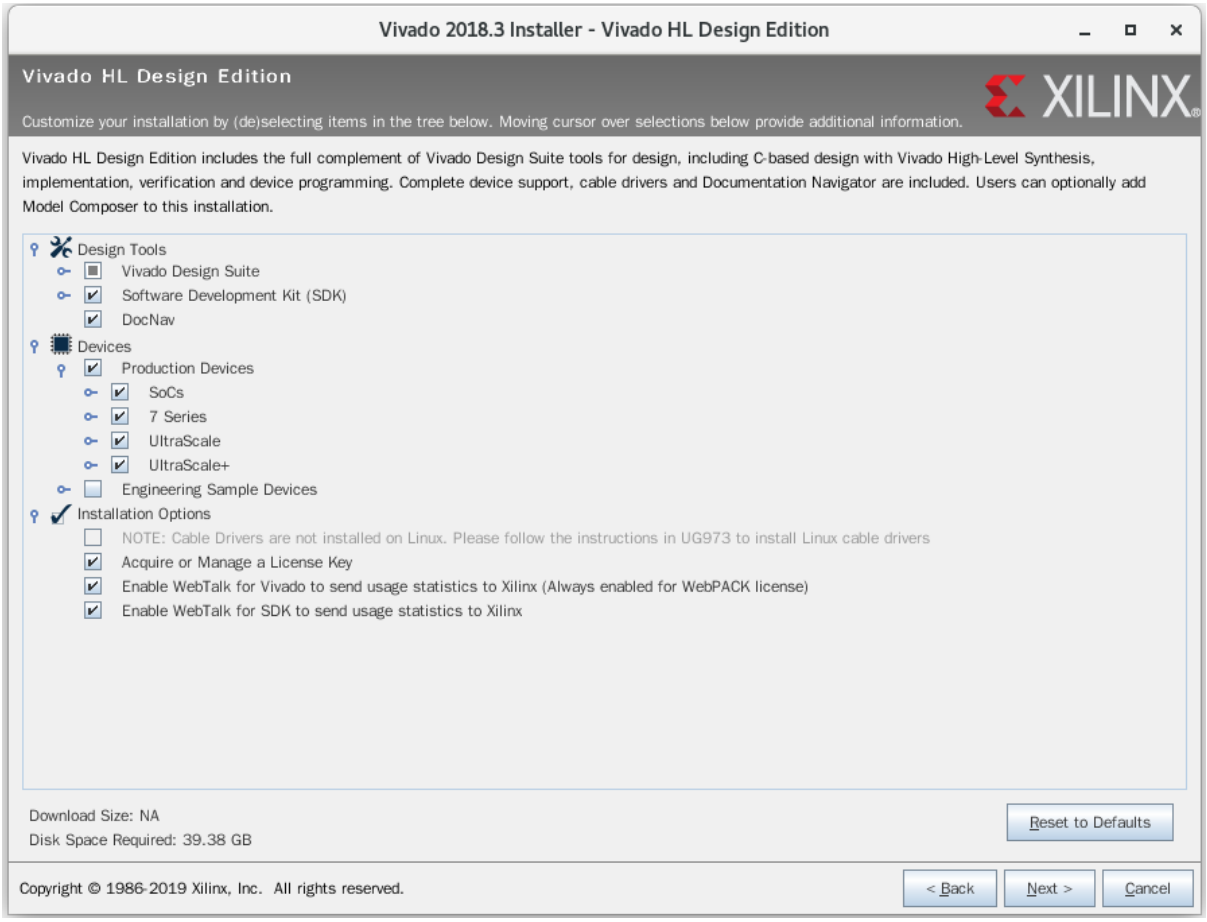
点击 continue 选择不下载最新版本，然后点击 Next 进入下一步



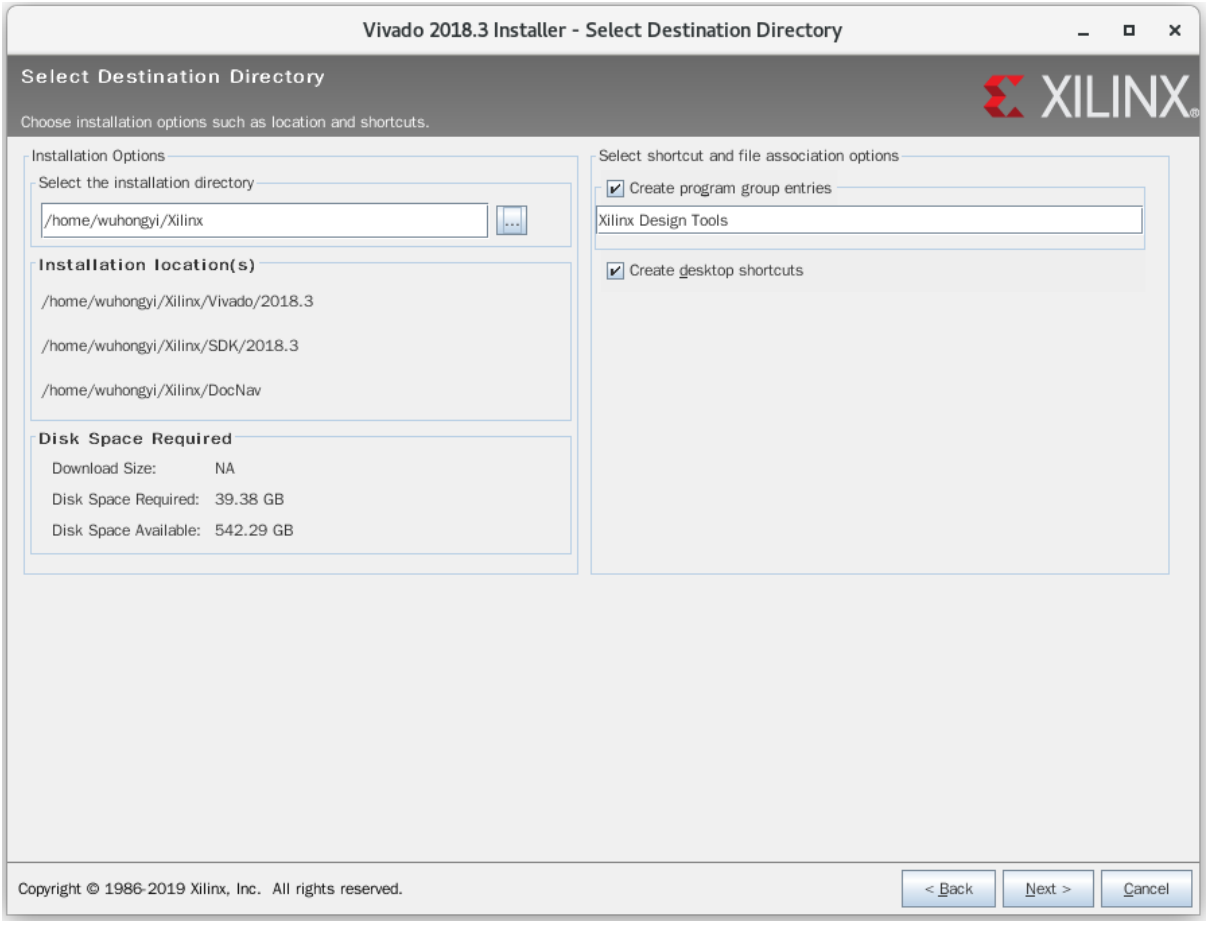
点击三个可选框，然后点击 Next 进入下一步



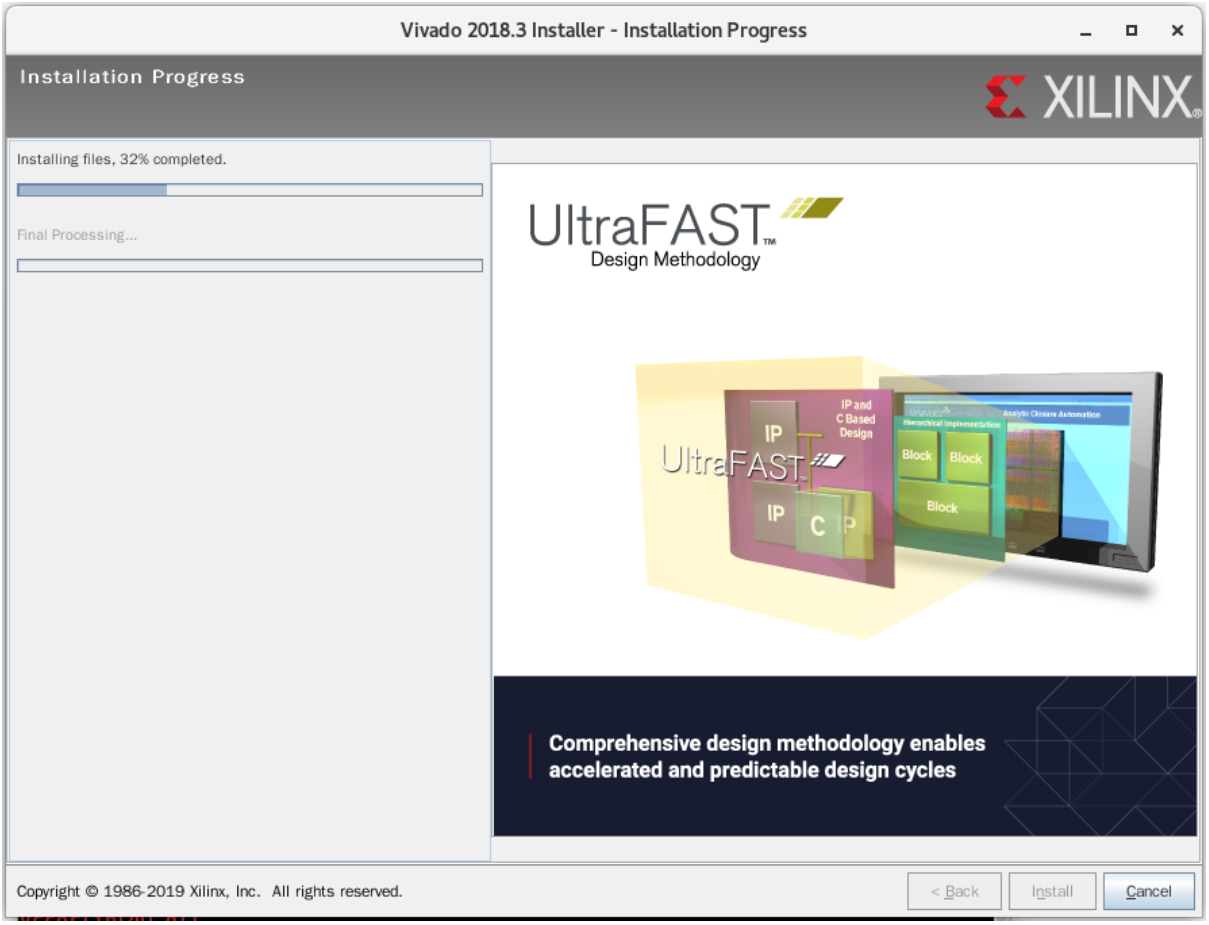
选择 Vinado HL Design Edition，然后点击 Next 进入下一步



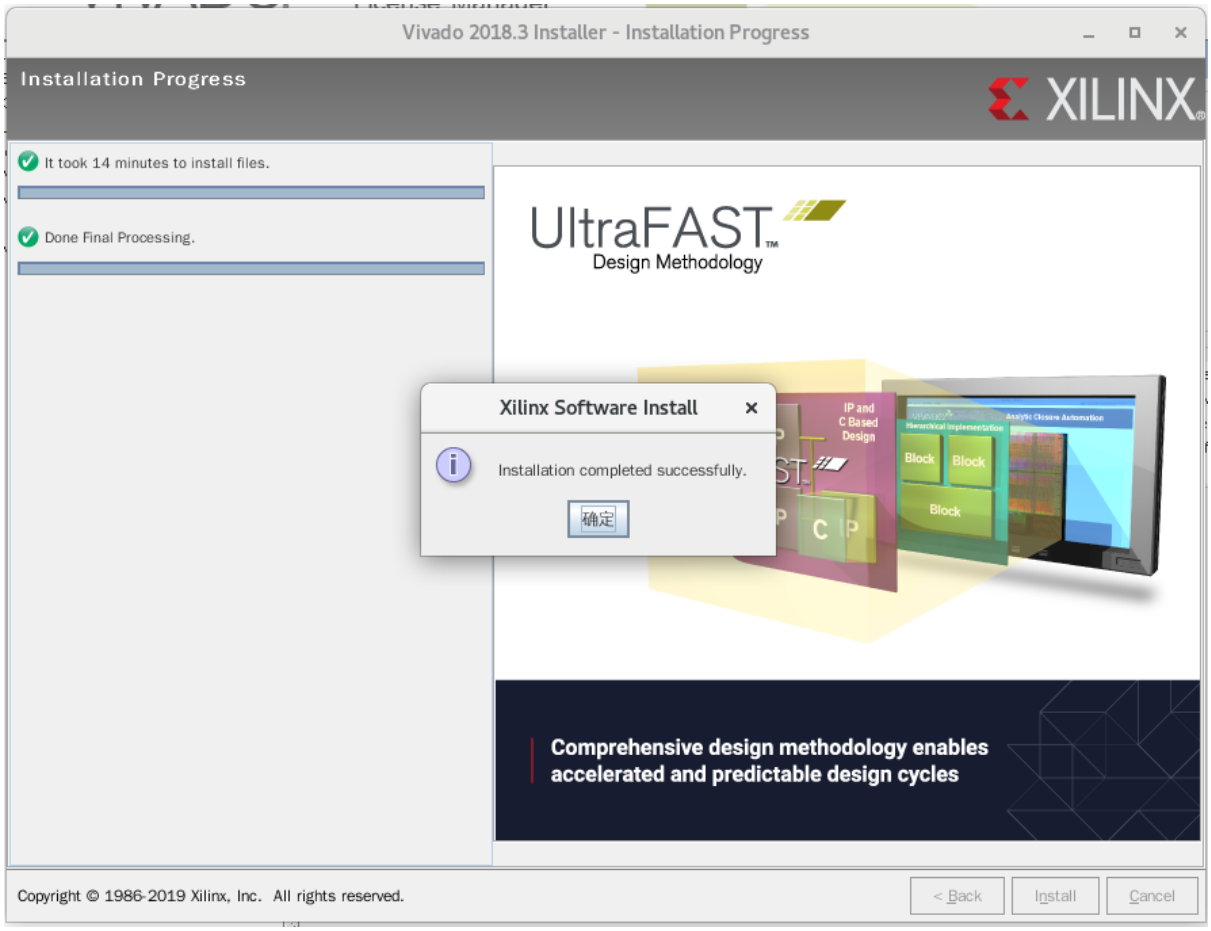
直接点击 Next 进入下一步



选择安装目录，这里我选择安装到 /home/wuhongyi/Xilinx ， 然后点击 Next 进入下一步


Vivado 2018.3 Installer - Installation Progress
Installation Progress
XILINX
Installing files, 32% completed.
Final Processing...
UltraFAST
Design Methodology
IP and C Based Design
IP
IP
C
IP
Block
Block
Block
UltraFAST
Comprehensive design methodology enables accelerated and predictable design cycles
Copyright © 1986-2019 Xilinx, Inc. All rights reserved.
< Back
Install
Cancel

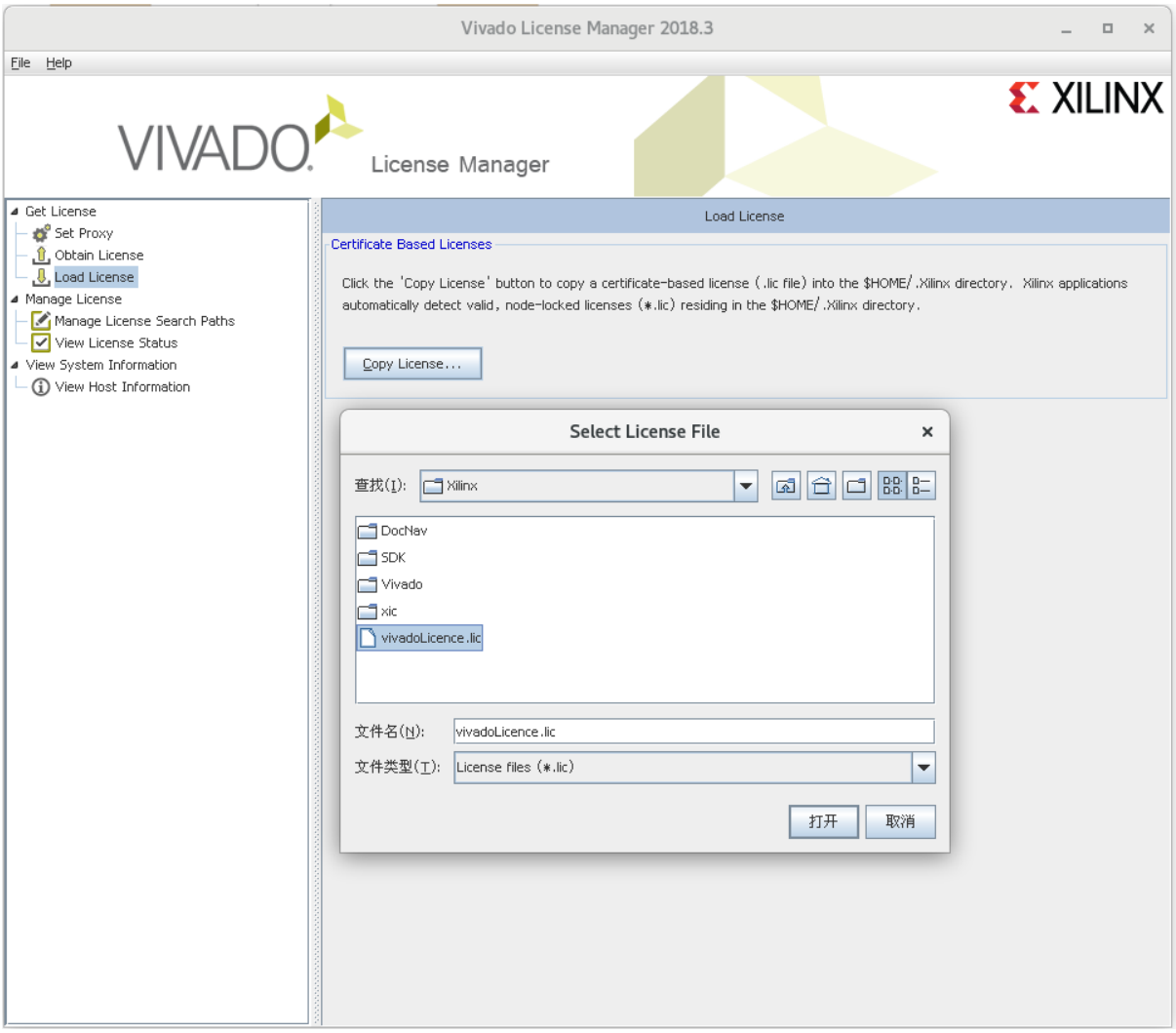
等待安装完成



以下两个步骤不是必须的。

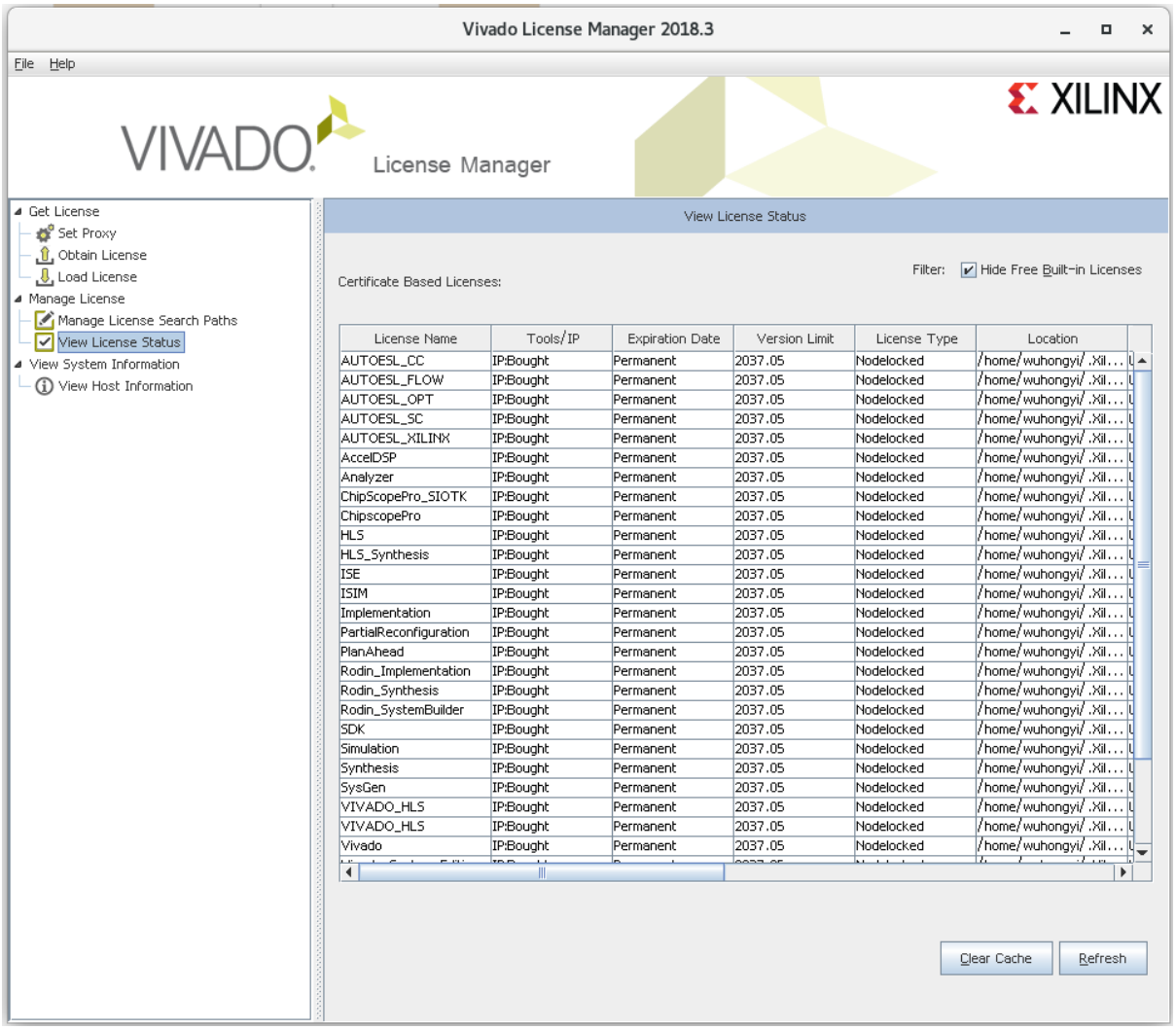
将 vivadoLicence.lic 文件复制到安装目录，这里为 /home/wuhongyi/Xilinx

安装完成之后会弹出以下界面



点击左上方的 Load License，选择我们的 vivadoLicence.lic 文件

然后点击左上方的 View License Status 可查看破解的 IP 核



5.2 编译

首次打开时，需要清空 `P16_MZTIO_FW_0p01/build` 文件夹

- 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

实验

关于 PKU 固件从前面板网口 RJ45 输出多重性选择的结果

- 当设置 multiplicity==0, 输出高电平
- 当设置 multiplicity>=1, 默认输出低电平，只有满足多重性条件时才有高电平。

MSRB bit6 为 1 时

- 才能有同步指示信号
- 才能 DPM 的输出信息
- 才有 FT，VT 信息

6.1 在线监视

在修改参数文件 `settings.ini` 之后，你需要运行以下程序来修改寄存器的设置。

```
./progfippi
```

需要注意的是，运行 DAQ 时不允许执行该程序

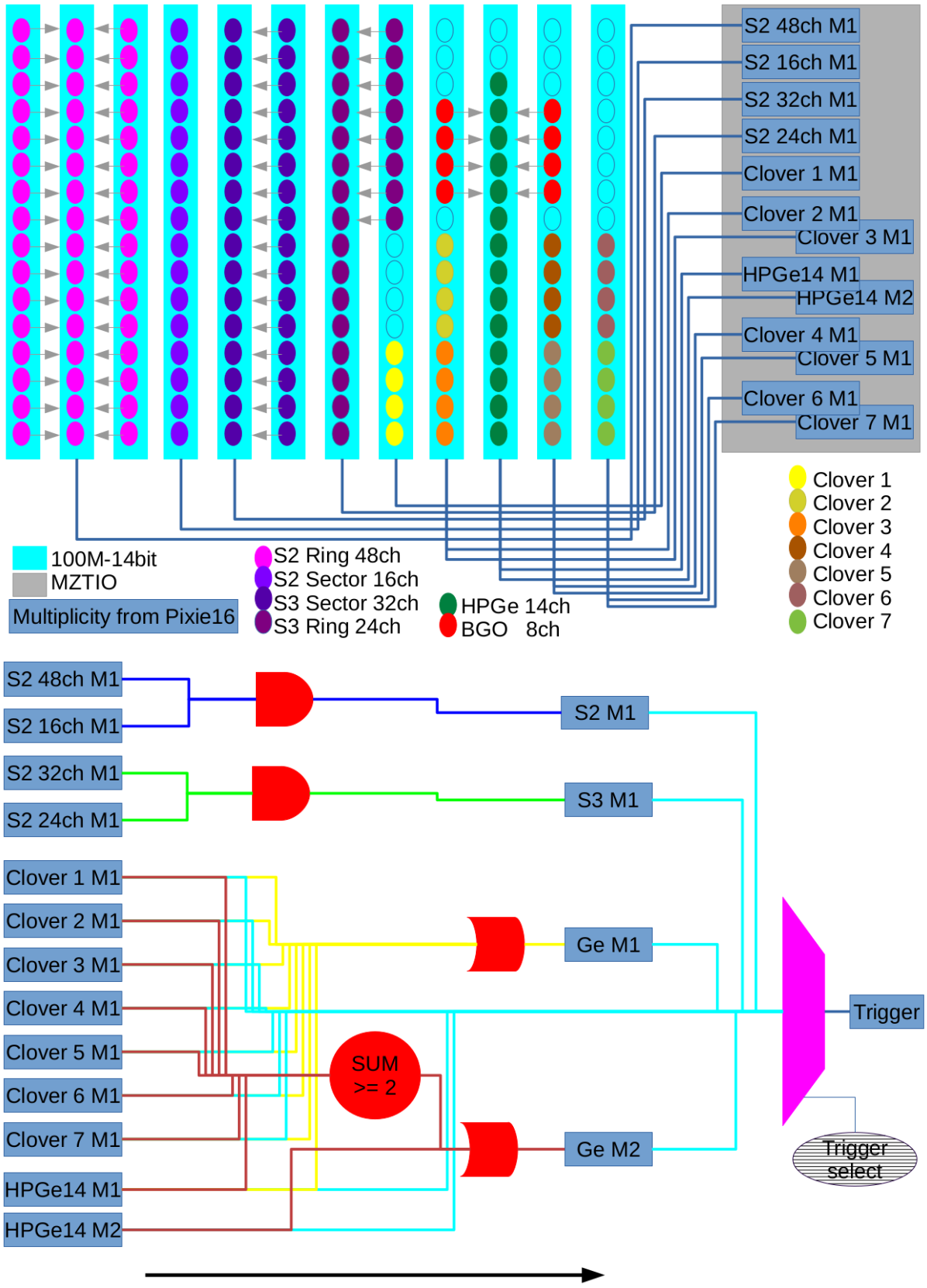
您可以在网页中查看参数设置，以及计数器等情况。

6.2 实验模式

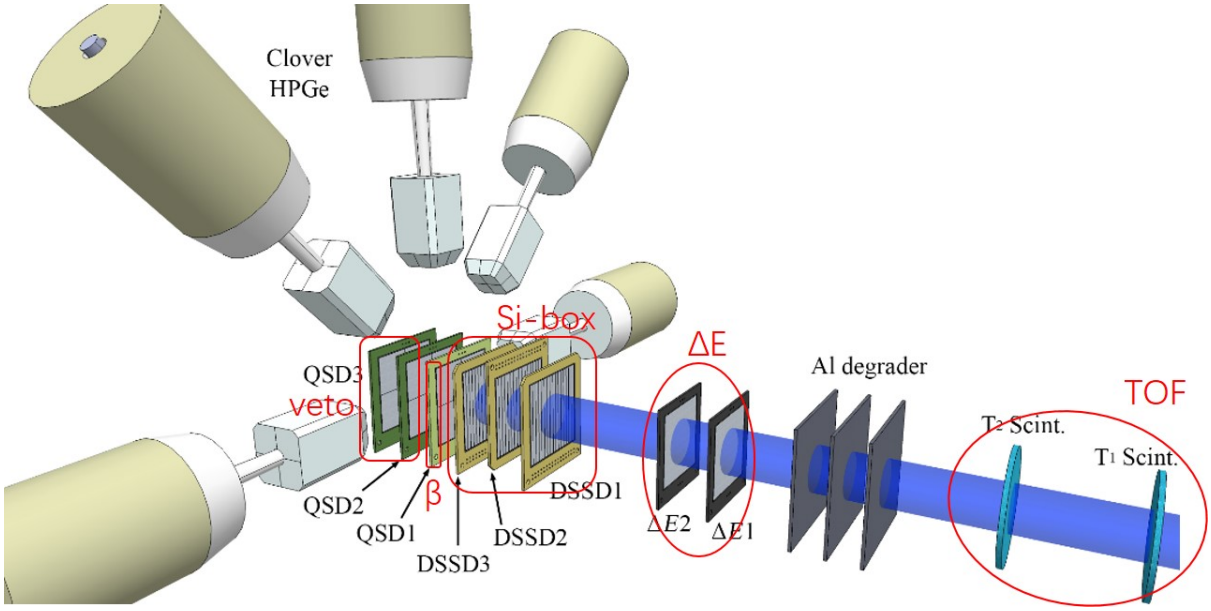
我们将为以下四种类型的实验提供固件和软件的通用组合。

6.2.1 在束 gamma 谱学

设计中…


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 衰变

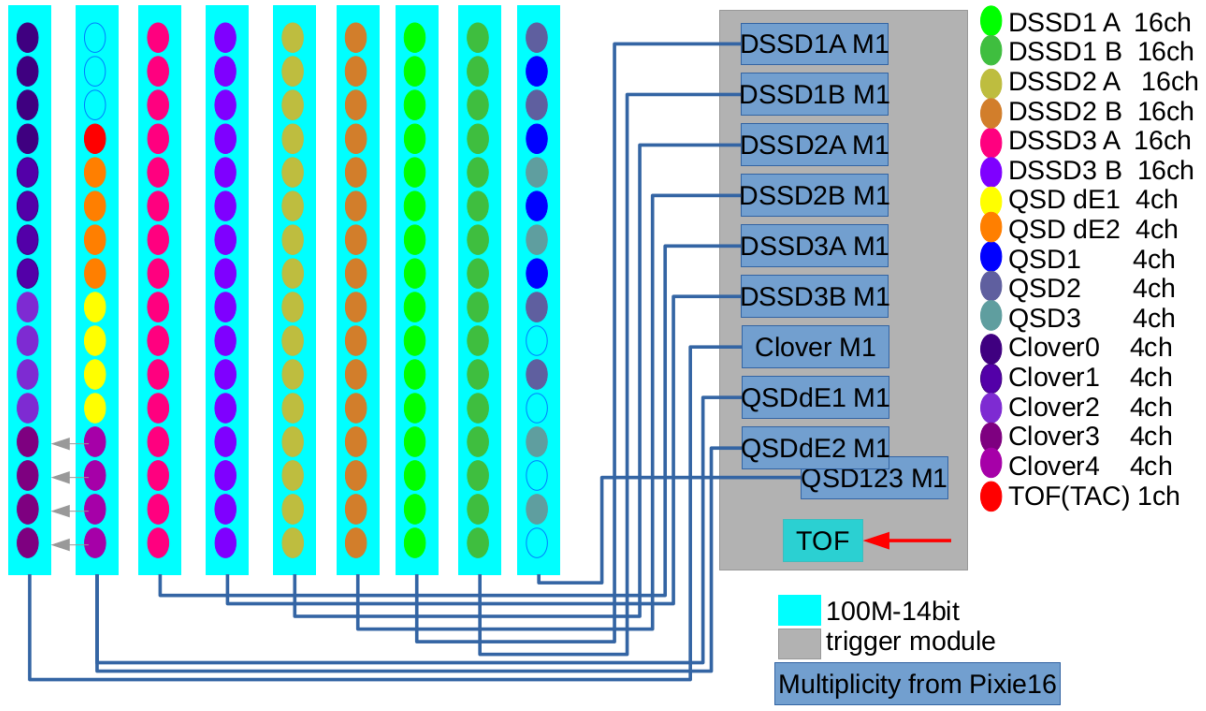


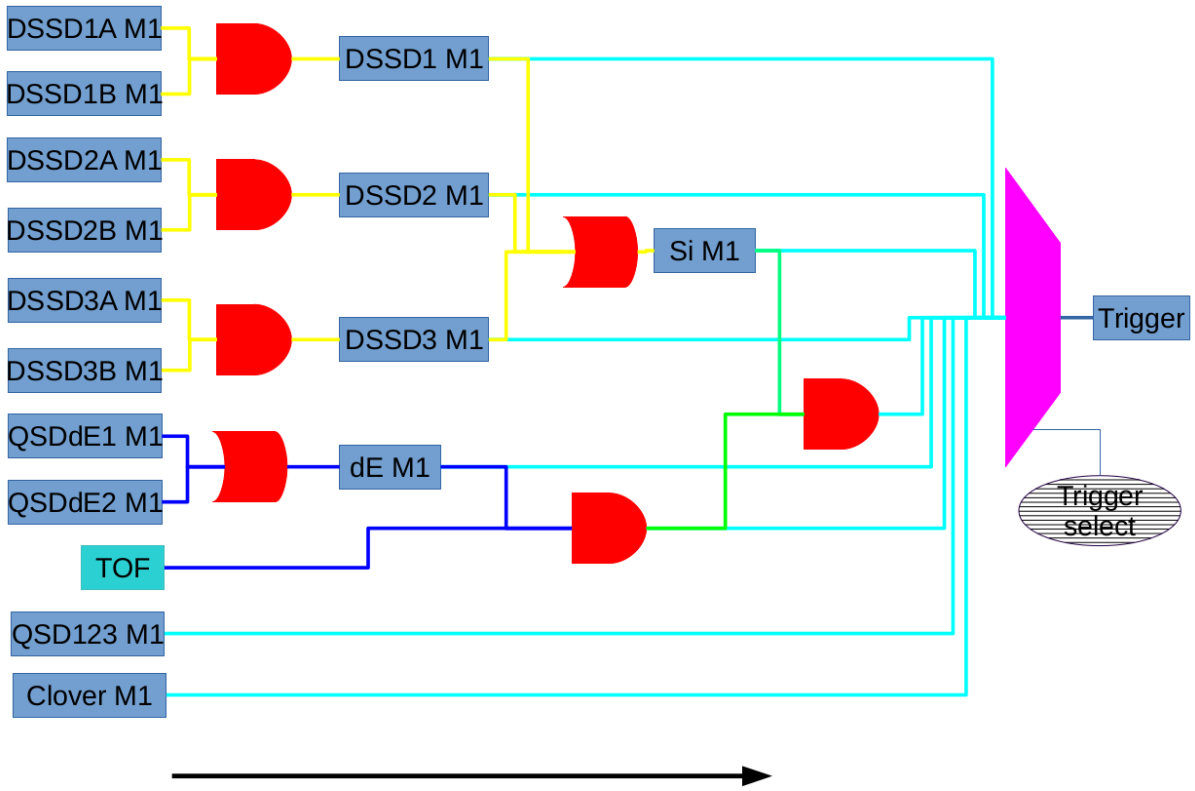
以下列出探测阵列中的硅探测器信息：

- **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

塑料闪烁体 T1，T2 信号经过 TAC 将时间差转为脉冲幅度信息，则可以使用 100MSPS 模块进行采集。

设计中…



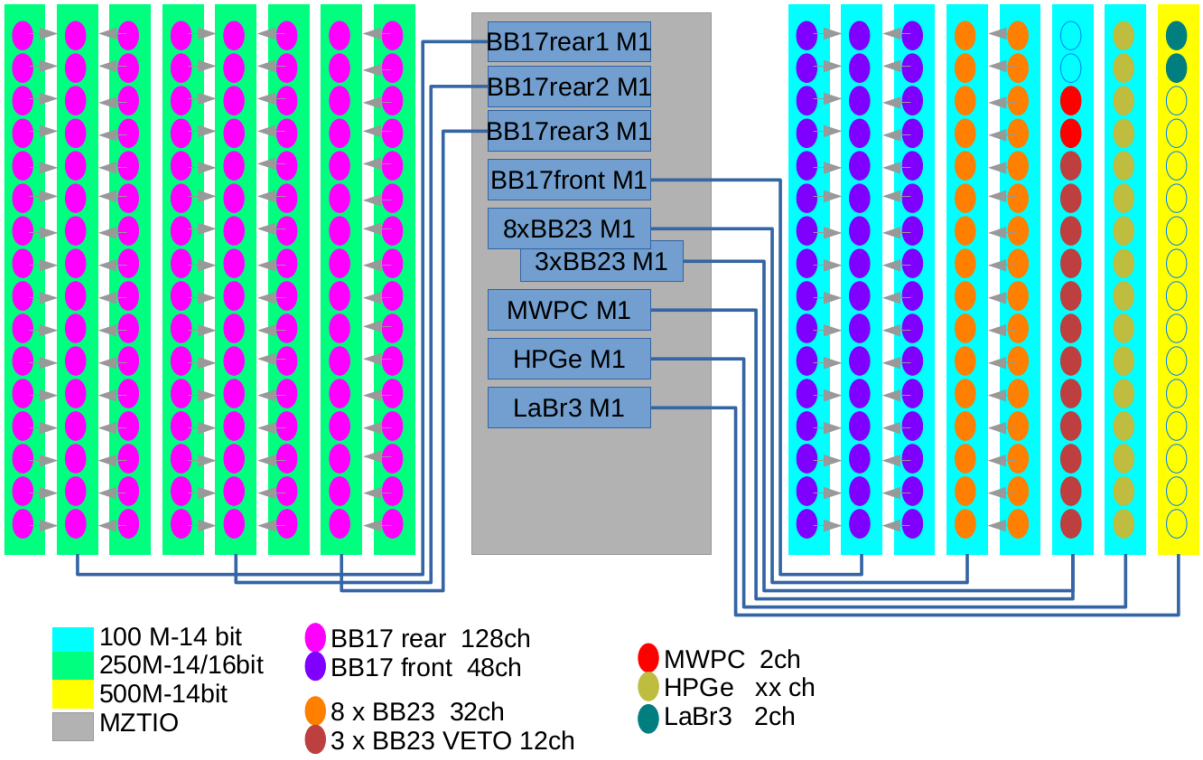


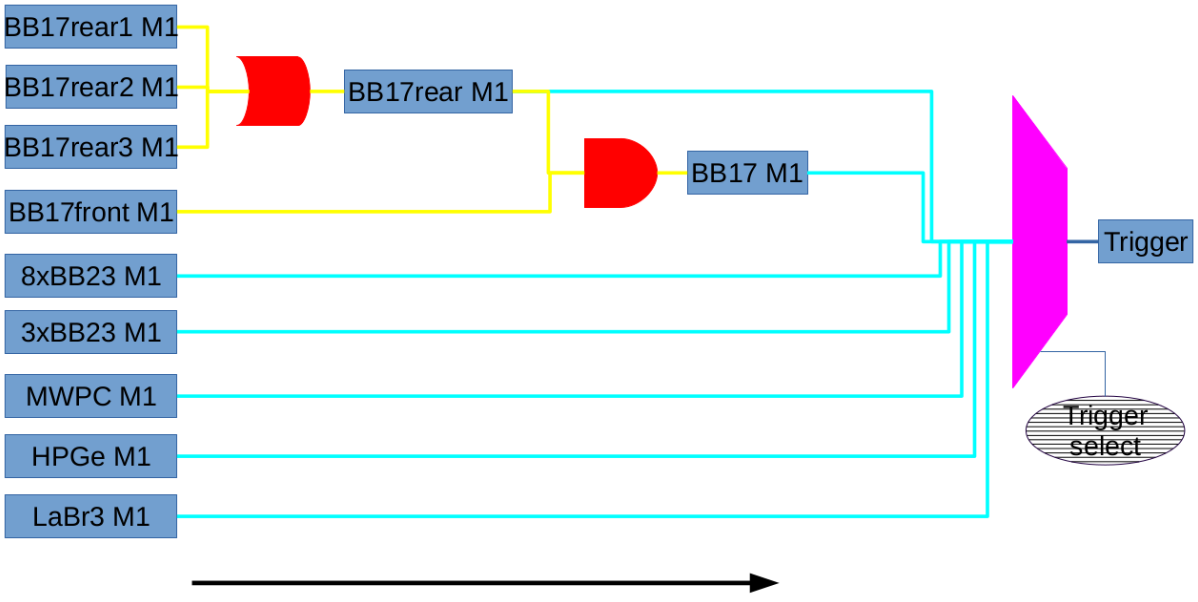
6.2.3 核反应

设计中…

6.2.4 超重核

设计中…


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

代码

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)

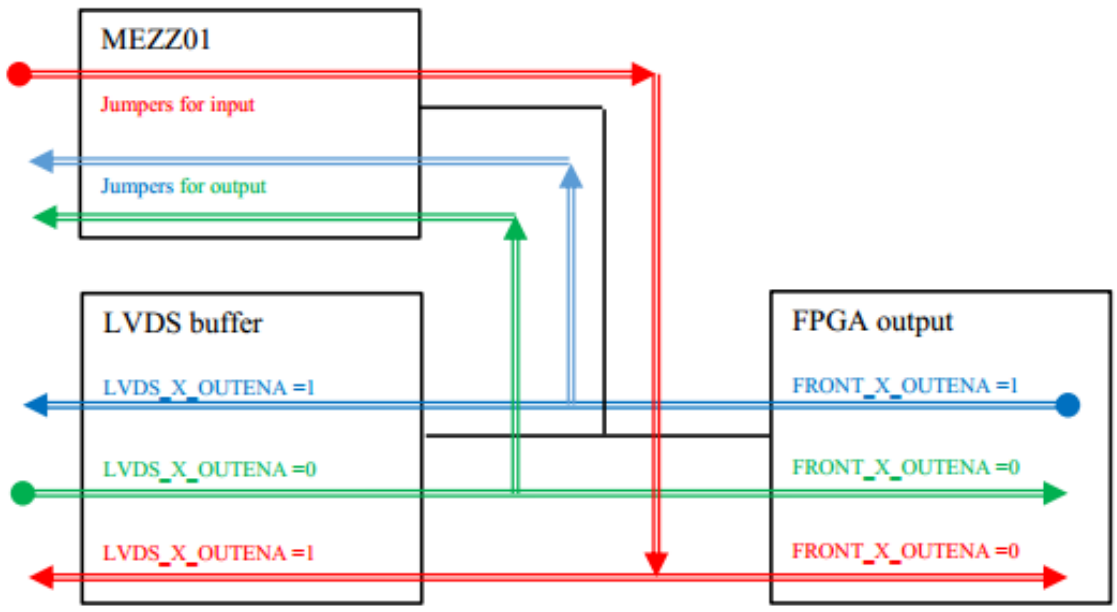
```
    dout,
    extend,
    clk
    );

    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

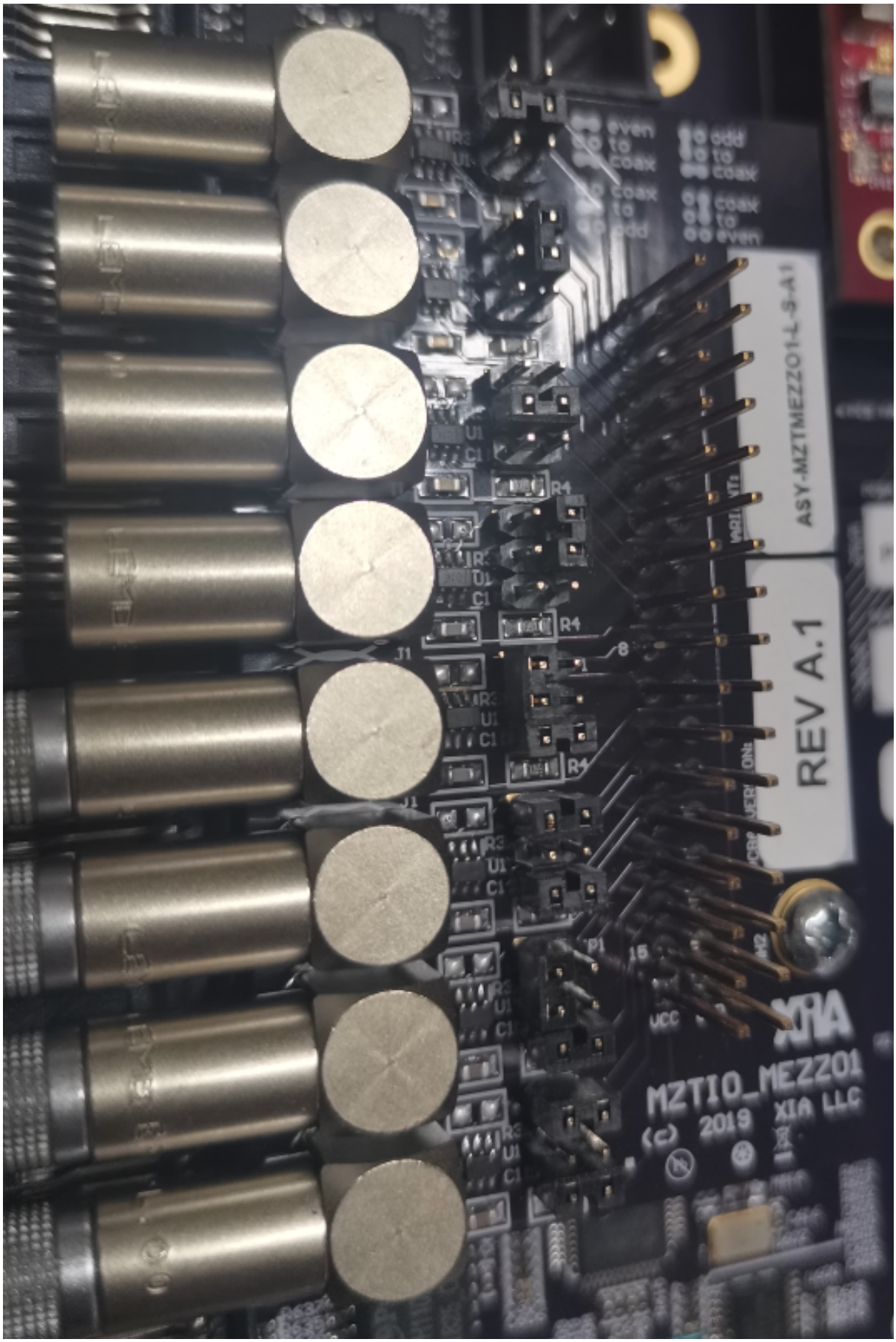


```
// The configuration of the FrontIO_A/B/C is completely flexible. For example, if
↪you connect the RJ-45 of a Pixie-16 to FrontI/O A 0-3 (the upper RJ-45 on the
↪trigger board), signals will connect
// FO5 - Front I/O A 3      FrontIO_Aena==0
// FO1 - Front I/O A 0      FrontIO_Aena==0
// FI5 - Front 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
XIA
even to coax
odd to coax
coax to odd
coax to even
J1
P1
U1
C1
R4

示波器监视模式：

- 0: A1_1
- 1: A1_2
- 2: A2_1
- 3: A2_2
- 4: A3_1
- 5: A3_2
- 6: A4_1
- 7: A4_2
- 8: B1_1
- 9: B1_2
- 10: B2_1
- 11: B2_2
- 12: B3_1
- 13: B3_2
- 14: B4_1
- 15: B4_2
- 16: C1_1
- 17: C1_2
- 18: C2_1
- 19: C2_2
- 20: C3_1
- 21: C3_2
- 22: C4_1
- 23: C4_2
- 24: DPM FULL 时处于低电平，统计处于高电平时间即为不丢失数据的时间
- 25: SYNCOUT，start=1，stop=0
- 26: Ext_ValidTrig_In 有信号时处于 High
- 27: Ext_FastTrig_In 有信号时处于 High
- 28: debug ch1
- 29: debug ch2
- 30: debug ch3
- 31: debug ch4
- 32:
- 33:
- 34:
- 35:

CHAPTER 8

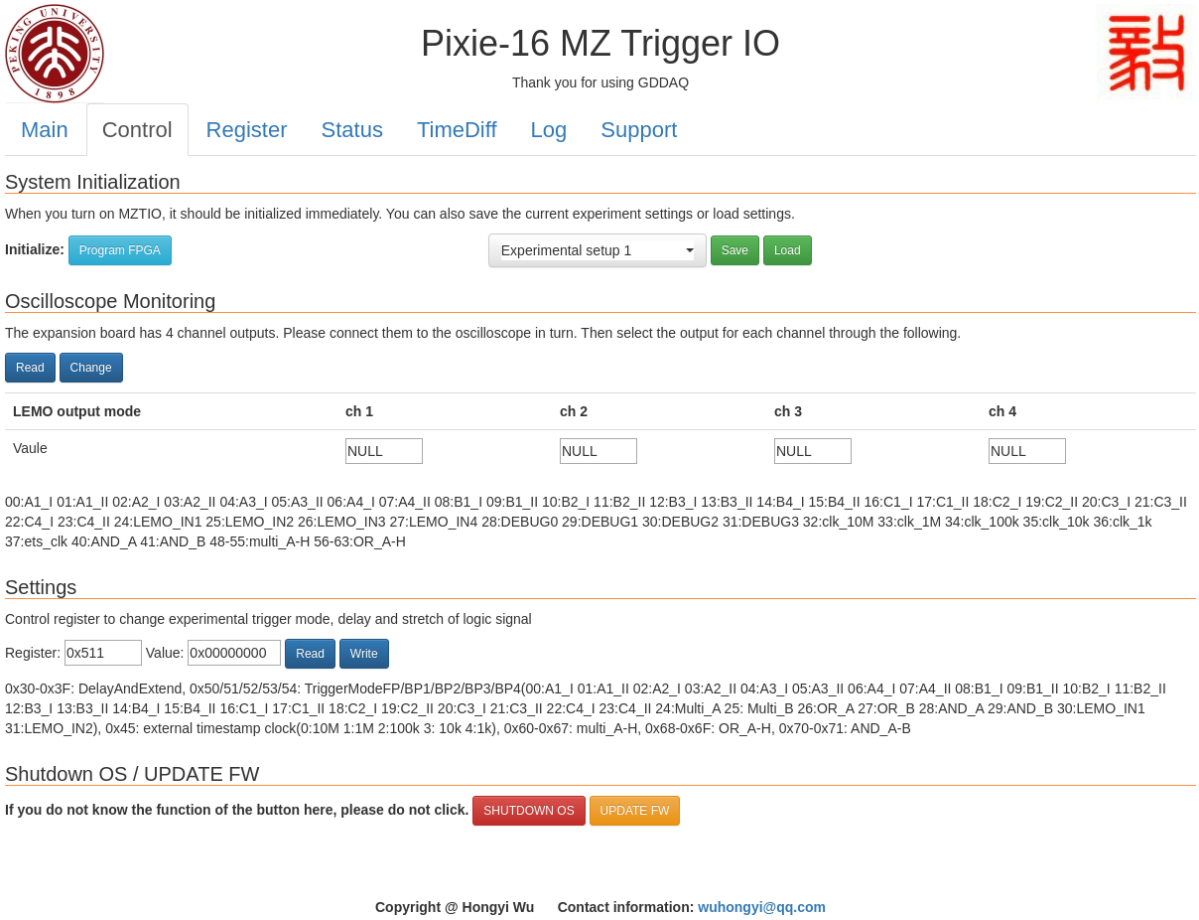
demo version 01

为了方便 GDDAQ 使用者熟悉 Pixie-16 模块的逻辑功能和 PKU 固件的特点，特别开发了本固件用于教学。用户可以在 https://github.com/wuhongyi/MZTIO/ 下载对应版本固件以及网页控制程序。

version/01 文件夹内包含固件 *xillydemo.bit* 和控制网页文件夹 *www*。**本固件及其配套的控制程序仅可用于获取的学习，实验版本请与吴鸿毅联系**。

MZTIO 前面板的最上面 12 个 RJ-45 连接器从上到下分别用以下符号表示：A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4。PKU 固件中 Pixie-16 模块上的 RJ-45 连接器输出 channel 0 和 channel 1 的多重性逻辑信号，分别用 _I 和 _II 表示。则 A1_I 表示 MZTIO 第一个 RJ-45 端口连接的 Pixie-16 模块中 channel 0 的多重性逻辑。

8.1 控制寄存器



按钮“Program FPGA”用于初始化系统配置，当操作系统上电之后第一时间点击该按钮来完成系统的初始化。

可以保存 5 个实验设置参数，分别为“Experimental setup 1-5”。通过修改寄存器进行实验逻辑配置之后，可以点击按钮“Save”保存，将会把当前 FPGA 寄存器参数保存到选定的实验配置中。按钮“Load”用于将选择的实验配置加载到 FPGA 中。

4 个 LEMO 输入通道分别命名为 LEMO1，LEMO2，LEMO3，LEMO4。其中 LEMO4 连接机箱背板的“run inhibit”信号，当输入信号为高电平时，获取系统停止运行。

示波器监视部分用于选择 4 个 LEMO 输出通道的输出信号，下表中列出了当前所有可供选择的选项。点击“Read”按钮即可读取当前的设置参数，按钮“Change”用于将当前输入框的参数写入 FPGA 中。

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
08	B1_I
09	B1_II

下页继续

表 1 – 续上页

vaule	signal
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

寄存器设置部分用于读取或者修改寄存器设置参数。读取寄存器时，需要输入要读取寄存器的地址，然后点击按钮“Read”；修改寄存器时，输入要修改寄存器的地址以及参数值，然后点击按钮“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

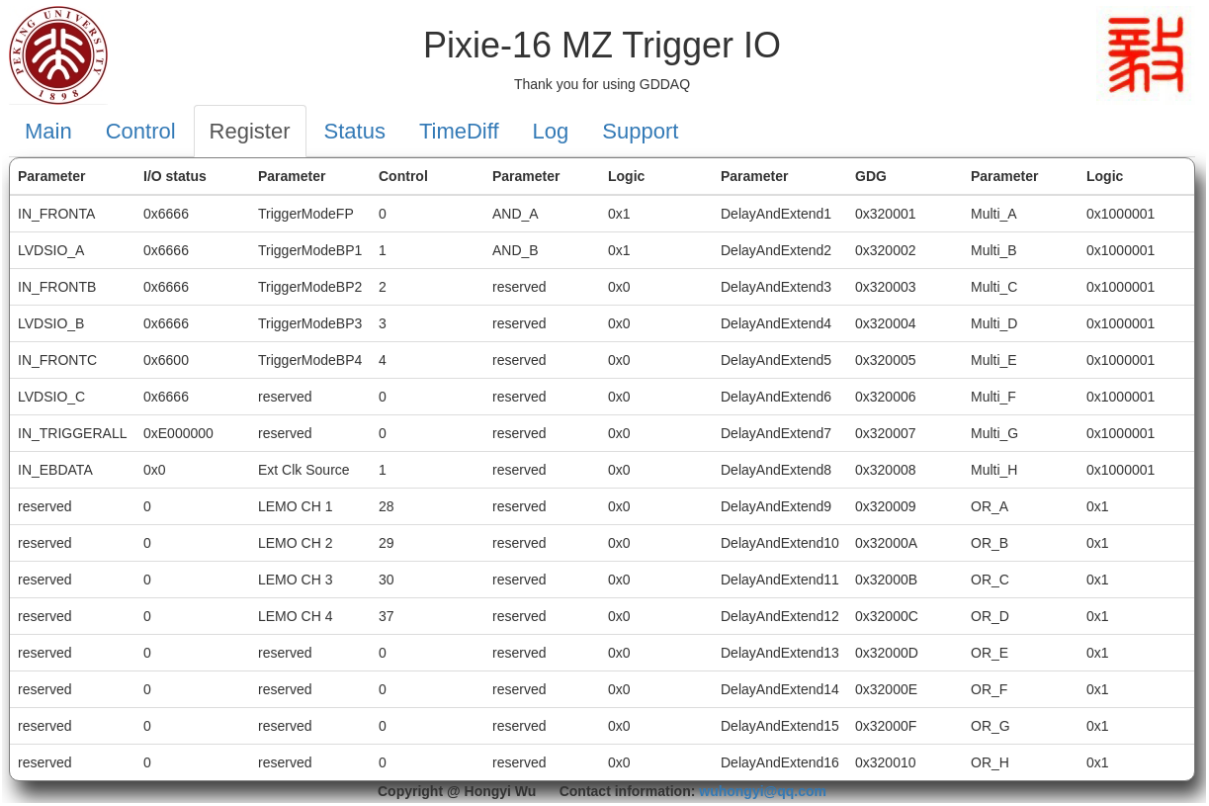
表 2 – 续上页

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)

网页也可用于 MZTIO 中 LINUX 操作系统的关闭，点击红色按钮“SHUTDOWN OS”将会弹出密码输入框，输入正确的密码后将立即关闭操作系统，此后将无法访问网页，需要重新上电才能开启操作系统。该按钮仅用于关闭机箱之前的关闭 MZTIO 操作系统。

黄色按钮“UPDATE FW”用于升级固件并重启操作系统。需要将要升级的固件放置在 */root* 目录下，然后点击按钮。如果固件升级成功，网页将会提示操作系统将在一分钟之后重启，如果升级失败，则提示找不到固件文件。

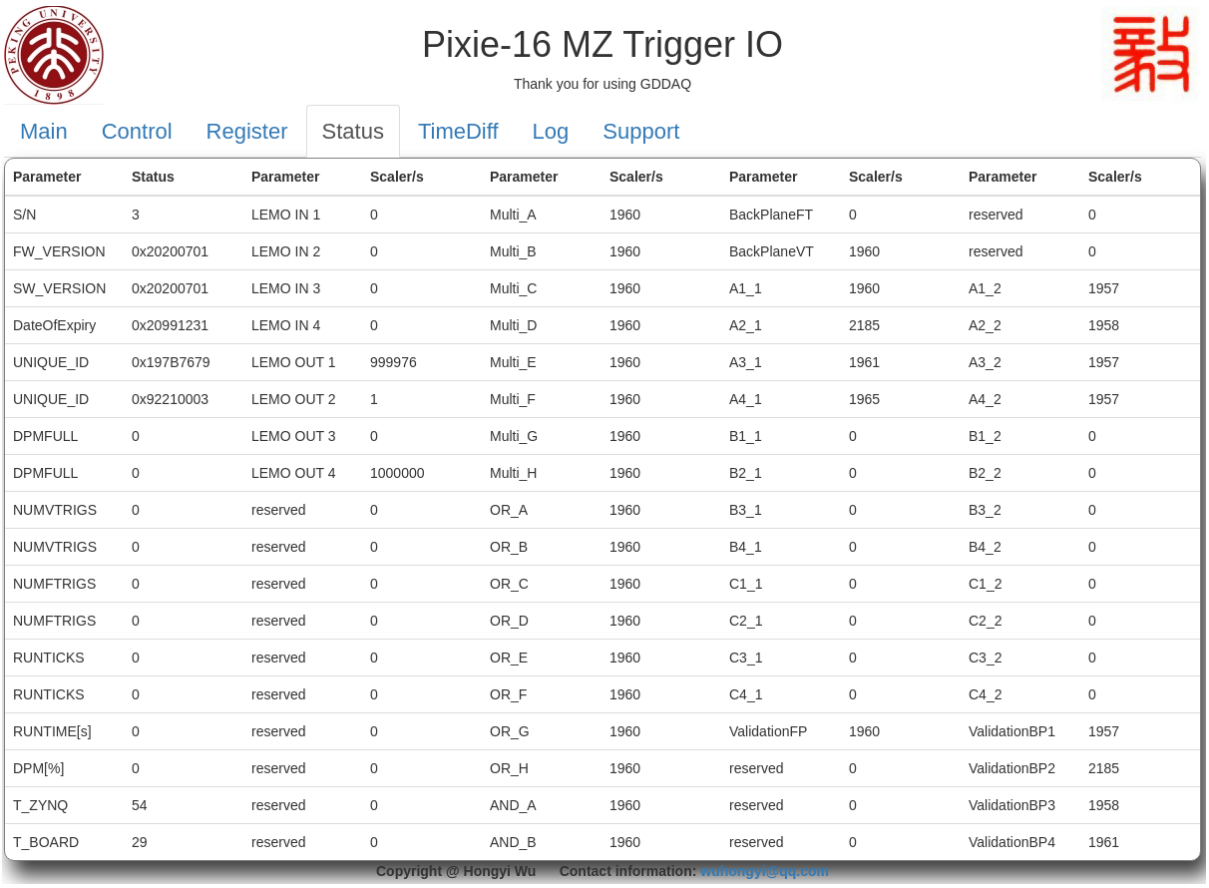
8.2 寄存器状态



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

该页面用于查看所设置的寄存器参数。

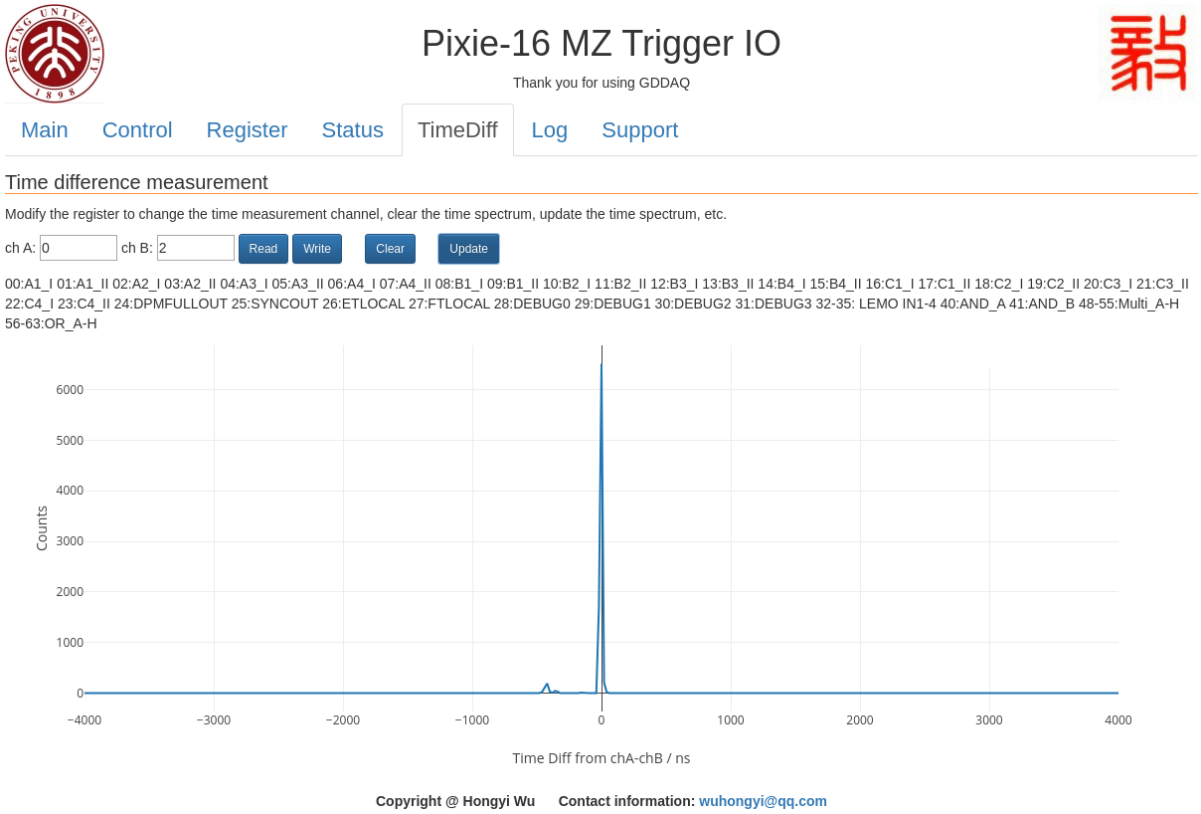
8.3 触发率监视



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

该页面用于实时的计数率监视。当前版本包含了 4 个 LEMO 输入通道的计数率，4 个 LEMO 输出通道的计数率，Multi_A-H 的计数率，OR_A-H 的计数率，12 个 RJ-45 连接器输入多重性信号的计数率等。

8.4 时间差谱测量



该页面实现了任意两个逻辑信号的时间差谱测量（chA-chB，时间差大于 0 表示 chA 信号晚于 chB 信号）。按钮“Read”用于读取信号源参数；按钮“Write”用于更改信号源；按钮“Clear”用于清除 FPGA 中的时间差谱，当更改信号源后必须清除 FPGA 中的时间差谱。按钮“Update”可用于从 FPGA 中读取当前的时间差谱并显示在网页上。

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
18	C2_I
19	C2_II

下页继续

表 3 – 续上页

vaule	signal
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