

---

# MZTIO

发布 0.0

Hongyi Wu(吴鸿毅)

2019 年 09 月 24 日



---

## Contents:

---

<b>1</b>	<b>README</b>	<b>3</b>
1.1	安全须知	3
1.2	逻辑编程	4
<b>2</b>	<b>网页控制界面</b>	<b>5</b>
2.1	寄存器	5
2.2	网页	6
2.3	示波器监视	8
2.4	FIFO IP 核的限制	8
<b>3</b>	<b>远程控制</b>	<b>11</b>
3.1	minicom	11
3.2	静态 IP 设置	12
<b>4</b>	<b>ubuntu</b>	<b>13</b>
4.1	basic configuration	13
4.2	恢复 SD 卡原始空间	14
4.3	升级启动文件	15
4.4	/dev/mmcblk0p1	15
<b>5</b>	<b>Vivado</b>	<b>17</b>
5.1	安装	17
5.2	编译	17
5.3	In system debug	17
<b>6</b>	<b>实验</b>	<b>19</b>
6.1	在线监视	19
6.2	实验模式	19
6.3	在束 gamma 谱学	19
6.4	beta 衰变	20
6.5	核反应	20
6.6	超重核	20
<b>7</b>	<b>Code</b>	<b>21</b>
7.1	PS code	21
7.2	PL code	21
7.3	xillydemo	23







---

如果您需要固件，请联系吴鸿毅 ([wuhongyi@qq.com](mailto: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  $\pm 3.3\text{V}$ 。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
- .....



### 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。

不安全 | 222.29.111.225/index.html



# Pixie-16 MZ Trigger IO

Thank you for using PKUXIADAQ



Main

Status

Log

Support

The Pixie-16 MZ-TriggerIO 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

**Do not visit the Status page while execute other tasks.**

**When you access the Status page, the page will automatically refresh every 5 seconds.**

Copyright © Hongyi Wu    Contact information: [wuhongyi@qq.com](mailto:wuhongyi@qq.com)


2.2.2 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。

← → 不安全 | 222.29.111.225/status.cgi



# Pixie-16 MZ Trigger IO

Thank you for using PKUXIADAQ



Main

Status

Log

Support

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
COINTEST	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	0x32EB0001	reserved	0x0	reserved	0x0	DS10	100	reserved	0

Copyright © Hongyi Wu    Contact information: [wuhongyi@qq.com](mailto:wuhongyi@qq.com)

2.2.3 log page

在开发中，此页面将保存状态参数并读取历史参数。

2.2.4 support page

该页面提供了一些基本说明，包括 XIA 说明，PKU 说明等。

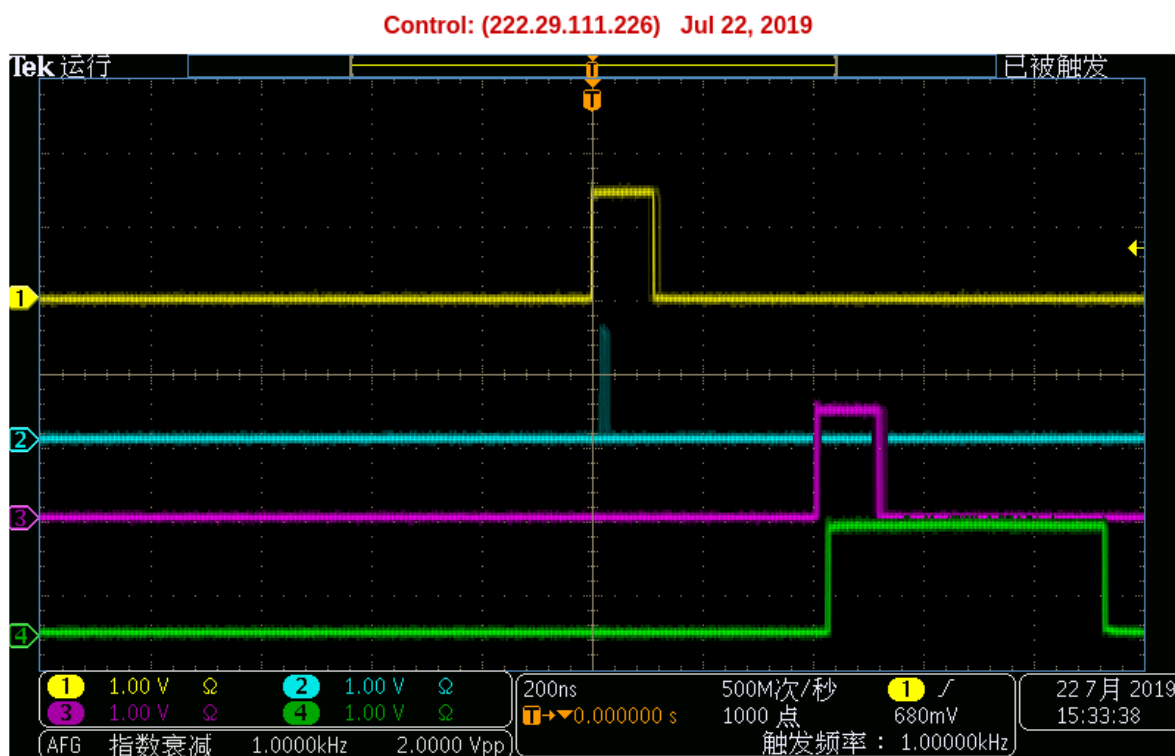
## 2.3 示波器监视

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。

Component Name

Basic

Native Ports

Status Flags

Data Counts

Summary

Optional Flags

☐ Almost Full Flag

☐ Almost Empty Flag

Handshaking Options

Write Port Handshaking

☐ Write Acknowledge

Active High

☐ Overflow

Active High

Read Port Handshaking

☐ Valid Flag

Active High

☐ Underflow Flag

Active High

Programmable Flags

Programmable Full Type

No Programmable Full Threshold

Full Threshold Assert Value

511

[6 - 511]

Full Threshold Negate Value

510

[5 - 510]

Programmable Empty Type

No Programmable Empty Threshold

Empty Threshold Assert Value

4

[4 - 510]

Empty Threshold Negate Value

5

[5 - 511]

由于 FIFO IP 核的限制，延迟设置为最少 4 个时钟。



### 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!



## 4.1 basic configuration

### 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。

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
```

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
```

```
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

## 4.2 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
```



### 5.1 安装

### 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 `.../verilog/vivado`
- 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???



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。

## 6.3 在束 gamma 谱学

设计中...

## 6.4 beta 衰变

设计中...

## 6.5 核反应

设计中...

## 6.6 超重核

设计中...



### 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

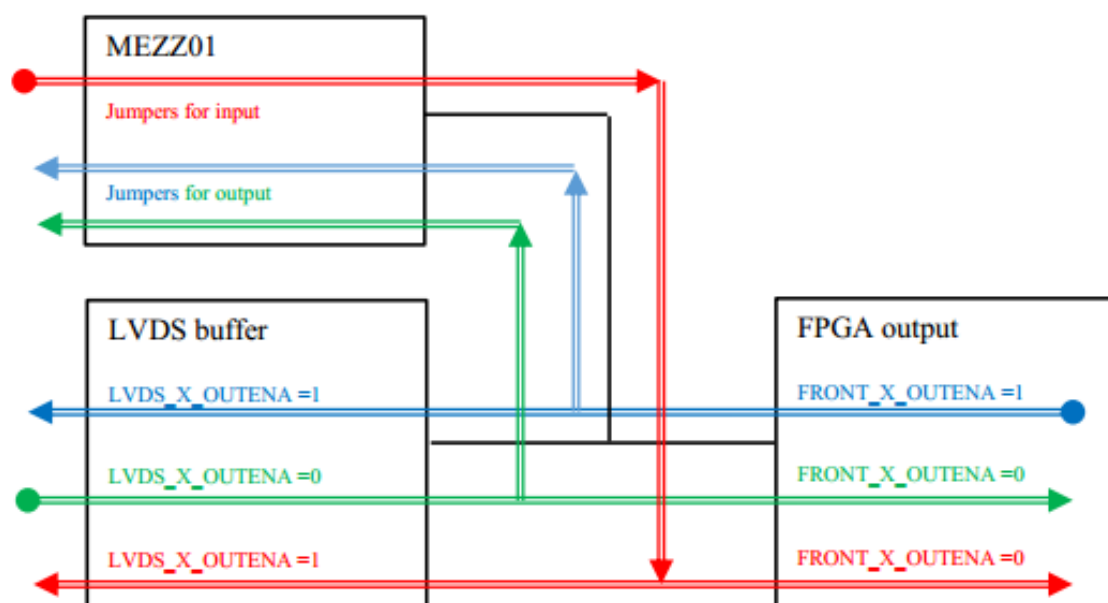
### FIFO

```

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



DB

```

// 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
// F05 - Front I/O A 3      FrontIO_Aena==0
// F01 - 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)
// F0 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出，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。