[Chinese Version](README.md)

##背景

在加载图片，执行 kmd 时，需要对代码输入输出功能和寄存器功能进行调试

##下载

Download from https://gitee.com/yds90/git_test.git
```bash
git clone https://gitee.com/yds90/git_test.git
```

输入输出内存功能使用

#fpga 上运行

```bash
Usage: ./nvdla_runtime [-options] --loadable <loadable_file>
where options include:
    -h                          print this help message
    -s                          launch test in server mode
    --image <file>              input jpg/pgm file
    --normalize <value>         normalize value for input image
    --mean <value>              comma separated mean value for input image
    --rawdump
```

```bash
insmod drvtest.ko
insmod opendla.ko
./nvdla_runtime   --loadable mnist_32x8x8.nvdla --image 9.jpg
```

>说明： --loadable 和--image 后面可替换成其它 loadable 文件和图片。

#echo 命令读取数据

```bash
cd /sys/class/mytest_class/mytest_device
```

>若输入"RM "或"WM "，则命令如下：

echo "RM start_addr,size" > my_device_test

或 echo "WM　start_addr,size" > my_device_test

#define path "/home/root/flw_kmd_test/weight_mnist_data.bin"
```bash

>说明： input、output、weight 数据写入文件的路径。

```

#运行截图

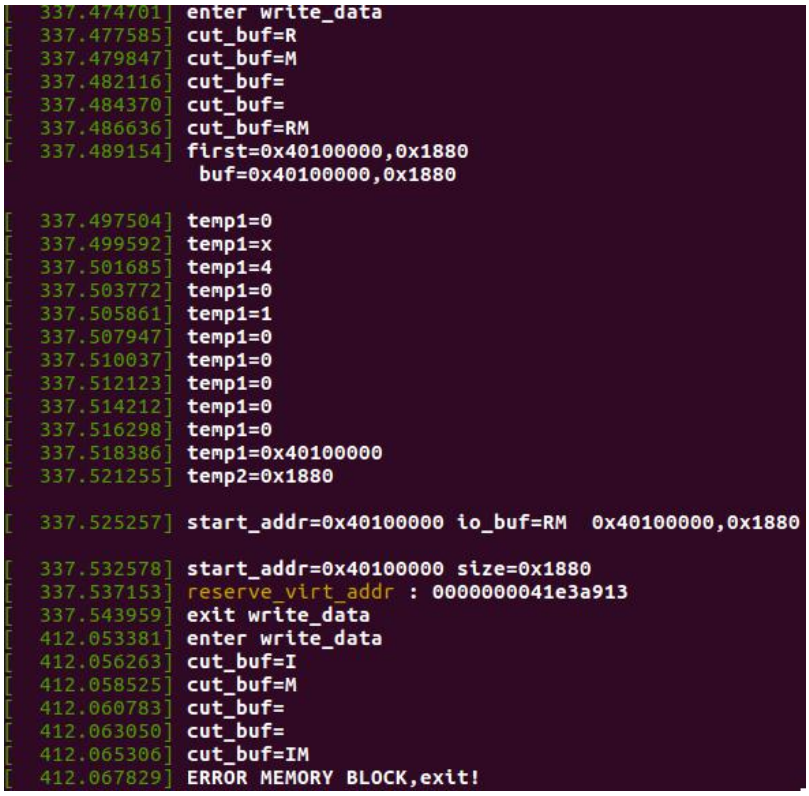
正确输入：echo "RM　0x40100000,0x1880" > my_device_test

正确运行如下，运行完成会生成二进制文件：weight_data.bin。

```
[  337.474701] enter write_data
[  337.477585] cut_buf=R
[  337.479847] cut_buf=M
[  337.482116] cut_buf=
[  337.484370] cut_buf=
[  337.486636] cut_buf=RM
[  337.489154] first=0x40100000,0x1880
                buf=0x40100000,0x1880

[  337.497504] temp1=0
[  337.499592] temp1=x
[  337.501685] temp1=4
[  337.503772] temp1=0
[  337.505861] temp1=1
[  337.507947] temp1=0
[  337.510037] temp1=0
[  337.512123] temp1=0
[  337.514212] temp1=0
[  337.516298] temp1=0
[  337.518386] temp1=0x40100000
[  337.521255] temp2=0x1880

[  337.525257] start_addr=0x40100000 io_buf=RM  0x40100000,0x1880

[  337.532578] start_addr=0x40100000 size=0x1880
[  337.537153] reserve_virt_addr : 0000000041e3a913
[  337.543959] exit write_data
```

错误输入：echo "IM　0x40100000,0x1880" > my_device_test

错误运行结果：



寄存器功能使用

```bash
cd /sys/class/mytest_class/mytest_device
```

>若输入"IO "，则命令如下：

```
echo "IO  start_addr,end_addr" > my_device_tes
```

>说明： 目前只能输入一个 start_addr 地址或者一段地址，可以打印出对应的寄存器数据

#运行截图

正确输入：echo "IO 0x3030,0x3098" > my_device_test

正确运行如下，打印一段寄存器数值。

```
[  255.612361] start_addr=0x3030 io_buf=IO  0x3030,0x3098

[  255.618977] start_addr : 0x3030
[  255.622116] READ reg_info : 0x0
[  255.625335] start_addr : 0x3034
[  255.628472] READ reg_info : 0x4000d000
[  255.632304] start_addr : 0x3038
[  255.635437] READ reg_info : 0x0
[  255.638657] start_addr : 0x303c
[  255.641791] READ reg_info : 0x4000d000
[  255.645619] start_addr : 0x3040
[  255.648747] READ reg_info : 0x8
[  255.651973] start_addr : 0x3044
[  255.655107] READ reg_info : 0x0
[  255.658329] start_addr : 0x3048
[  255.661456] READ reg_info : 0x8
[  255.664675] start_addr : 0x304c
[  255.667811] READ reg_info : 0x10001
[  255.671377] start_addr : 0x3050
[  255.674509] READ reg_info : 0x0
[  255.677730] start_addr : 0x3054
```

错误输入：echo "IO 0x40100000,0x1880" > my_device_test

运行得到错误结果：

```
[  220.685234] enter write_data
[  220.688119] cut_buf=I
[  220.690392] cut_buf=M
[  220.692646] cut_buf=
[  220.694909] cut_buf=0
[  220.697169] cut_buf=IM 0
[  220.699693] ERROR MEMORY BLOCK,exit!
[  346.358897] enter write_data
[  346.361786] cut_buf=I
[  346.364046] cut_buf=O
[  346.366316] cut_buf=
[  346.368578] cut_buf=
[  346.370841] cut_buf=IO
[  346.373362] first=0x40100000,0x1880
                buf=0x40100000,0x1880

[  346.381709] temp1=0
[  346.383795] temp1=x
[  346.385884] temp1=4
[  346.387971] temp1=0
[  346.390059] temp1=1
[  346.392149] temp1=0
[  346.394235] temp1=0
[  346.396322] temp1=0
[  346.398411] temp1=0
[  346.400497] temp1=0
[  346.402588] temp1=0x40100000
[  346.405454] temp2=0x1880

[  346.409465] start_addr=0x40100000 io_buf=IO  0x40100000,0x1880

[  346.416900] error input address,exit!
```