User Guide

OF CAFFE PORTING ONTO ACL

2017.05.30

****

目录

[1、Purpose 2](#_Toc484086325)

[2、Install Guide 2](#_Toc484086326)

[2.1、Directory Structure 2](#_Toc484086327)

[2.2、Compiled Environment Prepared 2](#_Toc484086328)

[2.3、Compile ACL 3](#_Toc484086329)

[2.4、Compile Caffe 3](#_Toc484086330)

[2.5、How to write the Makefile for compiling applications 3](#_Toc484086331)

[2.6、How to configure the libraries path to run applications 3](#_Toc484086332)

[3、Configure Guide 4](#_Toc484086333)

[3.1、Configure options in compile time 4](#_Toc484086334)

[3.2、Configure the bypass of ACL Layer 4](#_Toc484086335)

[3.3 configure the log informations 4](#_Toc484086336)

[4、Test and Performance Tuning Guide 5](#_Toc484086337)

[4.1、Compile Application and setup runtime environment 5](#_Toc484086338)

[4.2、To run the application with ACL and log performance information 5](#_Toc484086339)

[4.3、To run the application with original Caffe’s layers and log performance information 6](#_Toc484086340)

[4.4、Improve the performance by mixing ACL Layers and Caffe’s original Layers 7](#_Toc484086341)

# 1、Purpose

This document will help user utilize the code of CaffeOnACL(Caffe+ACL) to improve the performance of their applications based on the Caffe framework.

# 2、Install Guide

## 2.1、Directory Structure

Assume the directory structure of the code on firefly3399 is:

ACL：/home/firefly/ComputeLibrary (git clone <https://github.com/ARM-software/ComputeLibrary.git)>（arm\_compute v17.05）

Caffe：/home/firefly/caffeOnACL (git clone <https://github.com/OAID/caffeOnACL>.git)

Googletest: /home/firefly/googletest (git clone https://github.com/google/googletest.git)

## 2.2、Compiled Environment Prepared

sudo apt-get update -y

sudo apt-get upgrade -y

sudo apt-get install build-essential -y

sudo apt-get install -y pkg-config automake autoconf

sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libhdf5-serial-dev protobuf-compiler -y

sudo apt-get install libatlas-base-dev -y

sudo apt-get install --no-install-recommends libboost-all-dev -y

sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev -y

sudo apt-get install libopencv-dev -y

sudo apt-get install python-pip python-dev -y

sudo apt-get install -y python-numpy python-scipy

pip install --upgrade pip

sudo apt-get install scons –y

sudo apt-get install git –y

## 2.3、Compile ACL

cd /home/firefly/ComputeLibrary

# When in the cross compile environment, we maybe need the below command line to compile the libOpenCL.so

#aarch64-linux-gnu-gcc opencl-1.2-stubs/opencl\_stubs.c -Iinclude -shared -o build/libOpenCL.so

# pip install subprocess

scons Werror=1 -j8 debug=0 asserts=1 neon=1 opencl=1 embed\_kernels=1 os=linux arch=arm64-v8a

## 2.4、Compile Caffe

cd /home/firefly/caffeOnACL

cp Makefile.config.acl Makefile.config

make all

make distribute

## 2.5、How to write the Makefile for compiling applications

In the Makefile, the below content should be include(Read the Makefile in CaffeBenchmark as the example):

# include the configure file of caffeOnACL

include ../caffeOnACL/Makefile.config

#caffe’s libraries & include files

HOME=/home/firefly

CAFFE\_ROOT=$(HOME)/caffeOnACL

CAFFE\_INCS = -I$(CAFFE\_ROOT)/include -I$(CAFFE\_ROOT)/distribute/include/

CAFFE\_LIBS = -L$(CAFFE\_ROOT)/distribute/lib -lcaffe -lglog -lgflags -lprotobuf -lboost\_system -lboost\_filesystem

CAFFE\_RPATH =$(CAFFE\_ROOT)/distribute/lib

## 2.6、How to configure the libraries path to run applications

There’re 2 ways to configure the libraries of caffeOnACL:

* Copy “~/ComputeLibrary/build/arm\_compute/libarm\_compute.so” and “~/caffeOnACL/distribute/lib/libcaffe.so” to /use/lib
* export LD\_LIBRARY\_PATH=/home/firefly/ComputeLibrary/build/arm\_compute: /home/firefly/caffeOnACL /distribute/lib

# 3、Configure Guide

## 3.1、Configure options in compile time

* Enable ACL functions by “USE\_ACL :=1” in /home/firefly/caffeOnACL/Makefile.config, disable it with “USE\_ACL :=0”
* Enable profiling functions by “USE\_PROFILING := 1” in /home/firefly/caffeOnACL/Makefile.config, disable it with “USE\_PROFILING := 0”
* Experimental functions:
  + When USE\_PROFILING is true, we will enable “Layer’s performance statistic” which controlled by “-DLAYER\_PERF\_STAT” in /home/firefly/caffeOnACL/Makefile. You can remove “-DLAYER\_PERF\_STAT” to disable the feature.
  + Can add “-DUSE\_CONV\_CACHE” into /home/firefly/caffeOnACL/Makefile to enable the cache of convolution layer

## 3.2、Configure the bypass of ACL Layer

We can set “BYPASSACL” to bypass ACL layers, the control bit definitions are listed in the table below:

|  |  |
| --- | --- |
| BYPASS\_ACL\_ABSVAL | 0x00000001 |
| BYPASS\_ACL\_BNLL | 0x00000002 |
| BYPASS\_ACL\_CONV | 0x00000004 |
| BYPASS\_ACL\_FC | 0x00000008 |
| BYPASS\_ACL\_LRN | 0x00000010 |
| BYPASS\_ACL\_POOLING | 0x00000020 |
| BYPASS\_ACL\_RELU | 0x00000040 |
| BYPASS\_ACL\_SIGMOID | 0x00000080 |
| BYPASS\_ACL\_SOFTMAX | 0x00000100 |
| BYPASS\_ACL\_TANH | 0x00000200 |

For instance, we can use “export BYPASSACL=0x100” to bypass ACL Softmax layer; use “export BYPASSACL=0x124” to bypass ACL Softmax, Pooling and Convolution layers.

## 3.3 configure the log informations

We can set “LOGACL” to log the performance information of ACL and related caffe layers, the control bit definitions are listed in the table below:

|  |  |
| --- | --- |
| ENABLE\_LOG\_APP\_TIME | 0x00000001 |
| ENABLE\_LOG\_ALLOCATE | 0x00000002 |
| ENABLE\_LOG\_RUN | 0x00000004 |
| ENABLE\_LOG\_CONFIG | 0x00000008 |
| ENABLE\_LOG\_COPY | 0x00000010 |
| ENABLE\_LOG\_ABSVAL | 0x00000020 |
| ENABLE\_LOG\_BNLL | 0x00000040 |
| ENABLE\_LOG\_CONV | 0x00000080 |
| ENABLE\_LOG\_FC | 0x00000100 |
| ENABLE\_LOG\_LRN | 0x00000200 |
| ENABLE\_LOG\_POOLING | 0x00000400 |
| ENABLE\_LOG\_RELU | 0x00000800 |
| ENABLE\_LOG\_SIGMOID | 0x00001000 |
| ENABLE\_LOG\_SOFTMAX | 0x00002000 |
| ENABLE\_LOG\_TANH | 0x00004000 |

For instance, we can use “export LOGACL=0x100” to output the performance information of FC layer; use “export BYPASSACL=0x380” to output the performance information of LRN, FC and Convolution layers. If we copy the logs into Microsoft excel, we can sum the time with separated terms, the details of the column is –



# 4、Test and Performance Tuning Guide

## 4.1、Compile Application and setup runtime environment

We can read the “2、Install Guide” to learn how to compile the application with caffeOnACL and setup its runtime environment.

## 4.2、To run the application with ACL and log performance information

I assume your working directory is ~\test.

*#to use all ACL layers by set BYPASSACL to 0*

firefly@firefly:~/test$ export BYPASSACL=0

*#If we compile the caffeOnACL with “USE\_PROFILING := 1”, we can decide which information is logged into file by setting LOGACL. For instance, we log all layers’ information by setting LOGACL to 0x7fe1.*

firefly@firefly:~/test$ export LOGACL=0x7fe1

*#if we would like to check if “configure” take lots of time, we can set LOGACL to 0x08.*

firefly@firefly:~/test$ export LOGACL=0x08

*#if we would like to check if “memory copy” take lots of time, we can set LOGACL to 0x10.*

firefly@firefly:~/test$ export LOGACL=0x10

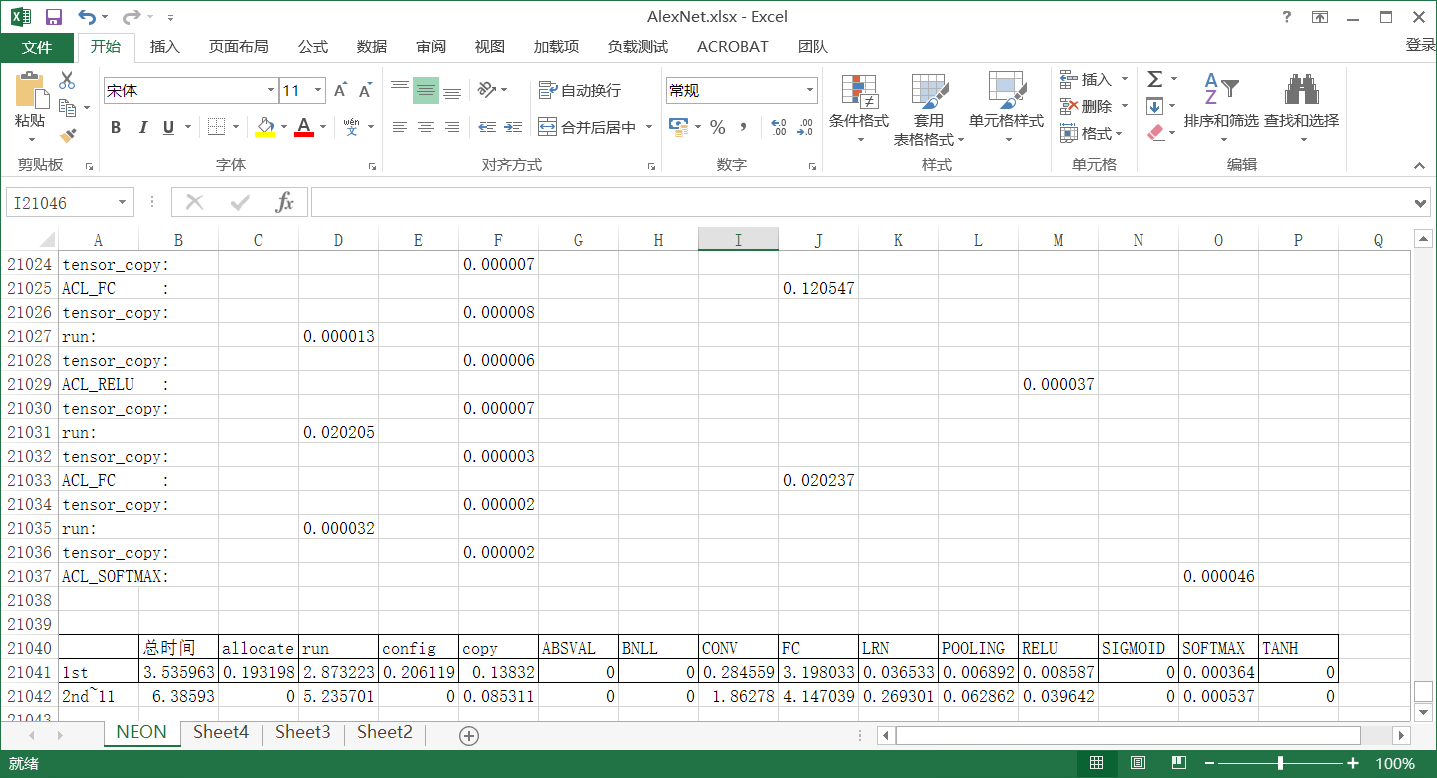
*#and then run your application and get the information of performance*

./your\_application parameters…

*#when got the log, we can copy it into Microsoft excel, and sum the columns (we use the AlexNet as the example – command line is :*

*taskset -a 10 ./distribute/bin/classification.bin ./models/bvlc\_alexnet/deploy.prototxt ./models/bvlc\_alexnet/bvlc\_alexnet.caffemodel data/ilsvrc12/imagenet\_mean.binaryproto data/ilsvrc12/synset\_words.txt examples/images/cat.jpg*

*)*



## 4.3、To run the application with original Caffe’s layers and log performance information

I assume your work directory is ~\test.

*#to use all ACL layers by set BYPASSACL to 0xffffffff*

firefly@firefly:~/test$ export BYPASSACL=0xffffffff

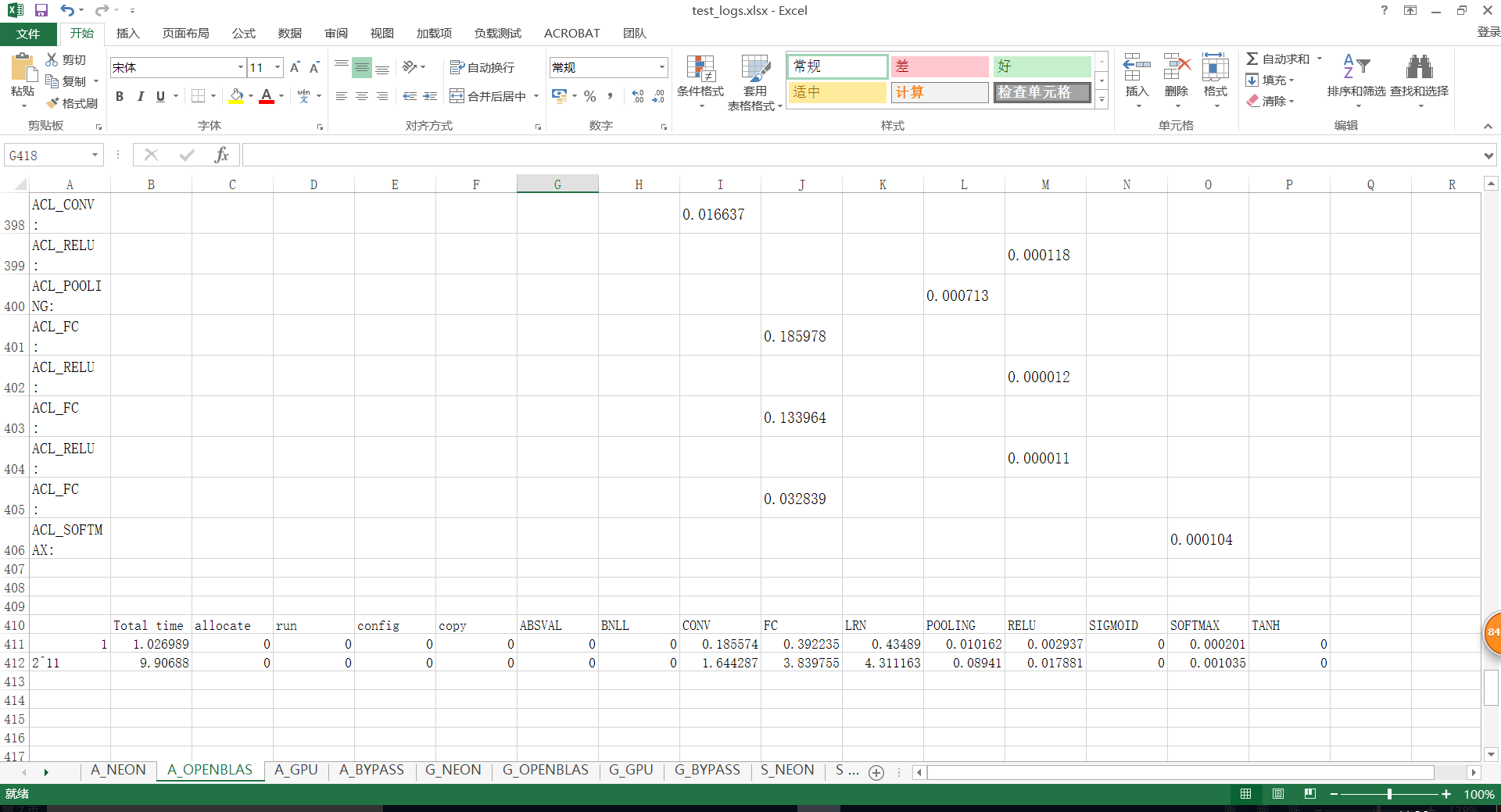
*#If we compile the caffeOnACL with “USE\_PROFILING := 1”, we can decide which information is logged into file by setting LOGACL. For instance, we log all layers’s information by setting LOGACL to 0x7fe1. (In this case, ENABLE\_LOG\_ALLOCATE、ENABLE\_LOG\_RUN、ENABLE\_LOG\_CONFIG and ENABLE\_LOG\_COPY are invalidate, these flags are all for ACL layers)*

firefly@firefly:~/test$ export LOGACL=0x7fe1

*#and then run your application and get the information of performance*

./your\_application parameters…

*#when got the log, we can copy it into Microsoft excel, and sum the columns*



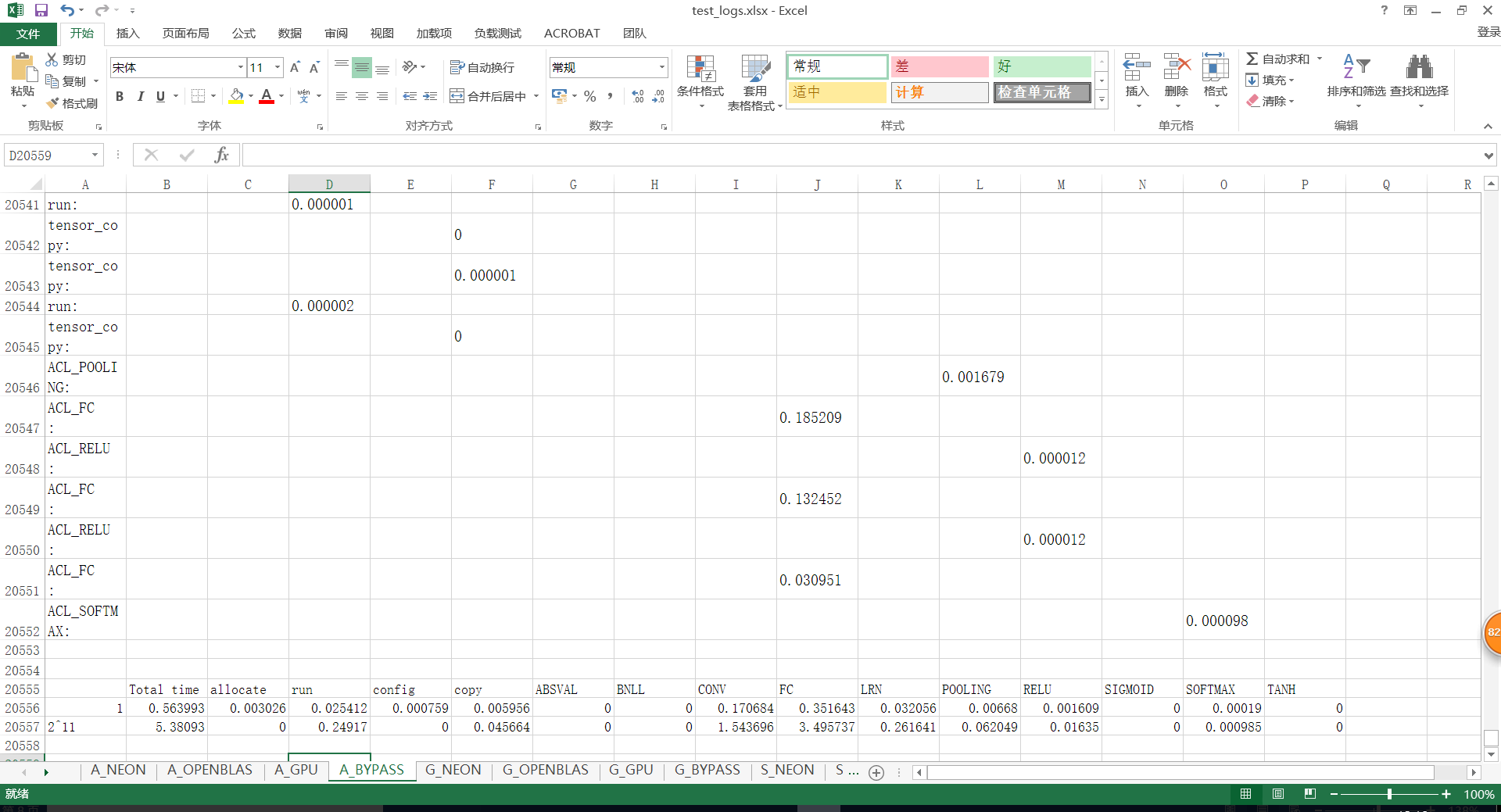
## 4.4、Improve the performance by mixing ACL Layers and Caffe’s original Layers

After retrieving the performance statistic data of Caffe’s layers and ACL’s layers in your application, we can compare their respective performances:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | total time | CONV | FC | LRN | Pooling | RELU | SOFTMAX |
| ACL\_NEON | 3.535963 | 0.284559 | 3.198033 | 0.036533 | 0.006892 | 0.008587 | 0.000364 |
| Caffe\_Org(OpenBLAS) | 1.026989 | 0.185574 | 0.392235 | 0.43489 | 0.010162 | 0.002937 | 0.000201 |

From the table above, we can observe that in the original caffe’s layer, CONV、FC、RELU and Softmax have faster running times than ACL’s layers. Therefore, we can set BYPASSACL to 0x14c to BYPASS the 4 ACL layers, and utilize the original caffe’s layers in the application. By choosing the layerset with the faster running time for each layer, we can optimize the total running time for this application.

The performance data is :



As you can see, we obtain optimal performance in combined mode (ACL: LRN、Pooling，Caffe’s original Layers：Conv、FC、RELU、Softmax) as in the table below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | total time | CONV | FC | LRN | Pooling | RELU | SOFTMAX |
| BYPASS  (CONV、FC、RELU and Softmax) | 0.563993 | 0.170684 | 0.351643 | 0.032056 | 0.00668 | 0.001609 | 0.00019 |