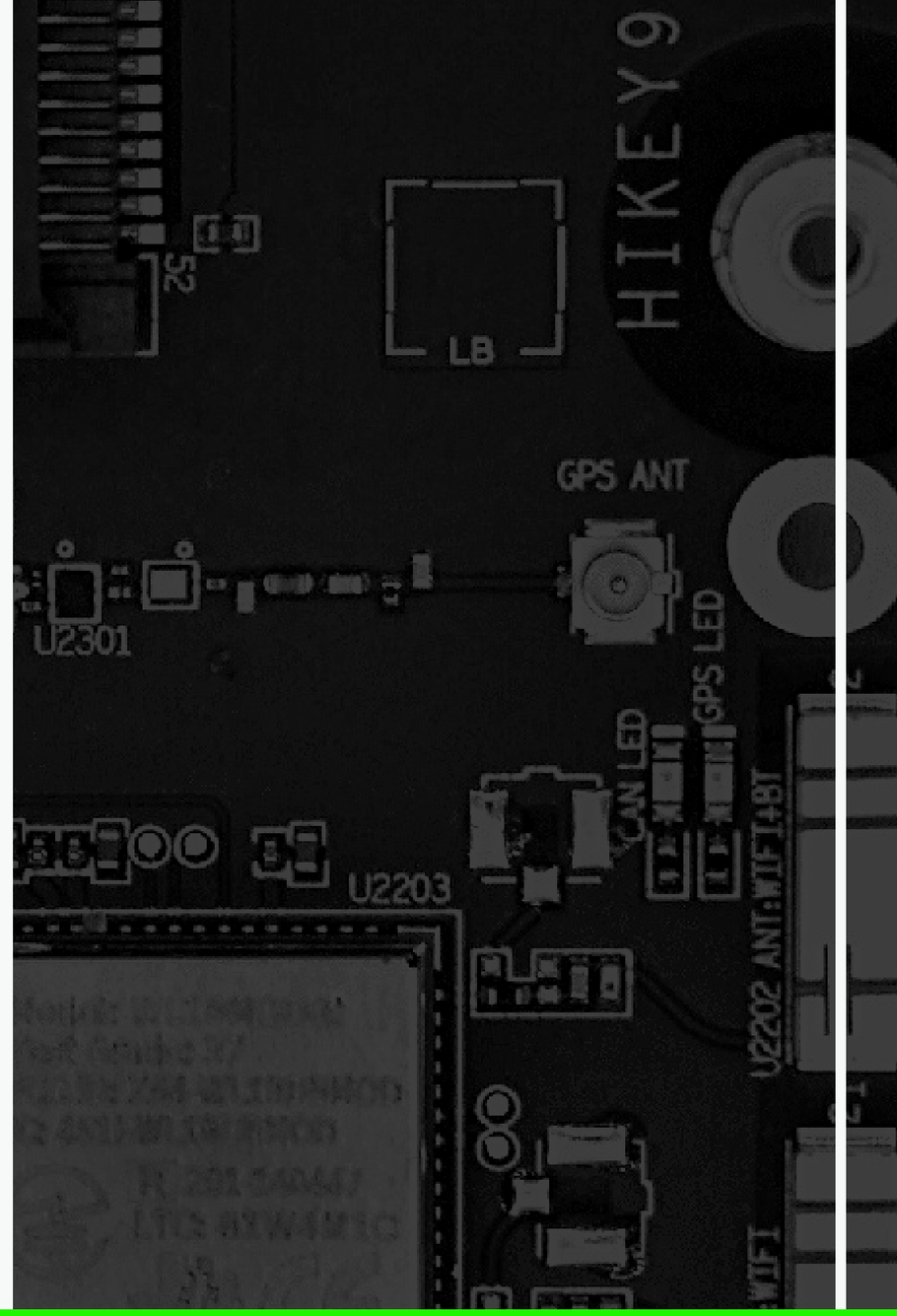
SHUNYA OS

RUNNING AIoT ON GPU -LESS LOW POWER EMBEDDED SYSTEMS

OUTLINE

AGENDA

- **AI BASICS**
- **CHALLENGES IN EMBEDDED**
- **DEPLOY AI ON EMBEDDED THE SMART WAY**



AI BASICS

CAN WE EVER GET **ENOUGH** PERFORMANCE ON EMBEDDED ?

AI BASICS

GENERAL PIPELINE

DATA INGESTION

DATA CLEANING

MODEL TRAINING

MODEL TESTING

MODEL DEPLOYMENT

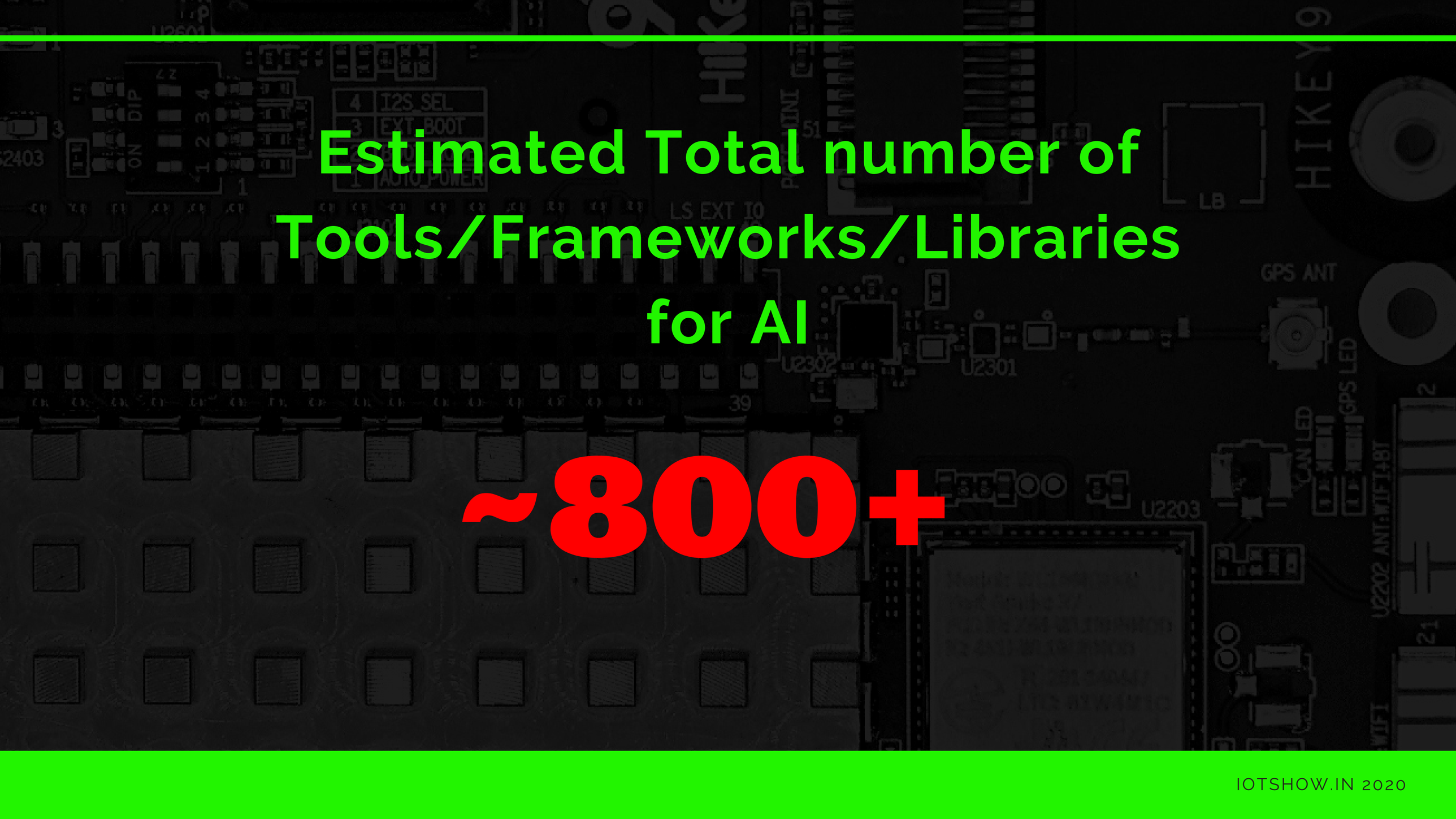
CHALLENGES IN EMBEDDED

Application Side

- Model size
- Choosing appropriate Model
- Choosing appropriate Frameworks
- Training Model

System Side

- Manual Installation of frameworks
- Configuring Heterogeneous computing
- System Optimization

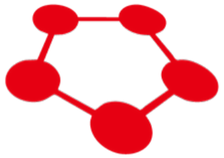

Estimated Total number of
Tools/Frameworks/Libraries
for AI
~800+

But ..

The most popular ones are

Caffe

~20+



Chainer

ONNX

Tencent 腾讯

PYTORCH

HOW DO WE INSTALL AI LIBRARIES ?

```
$ pip3 install tensorflow
```

HOW DO WE INSTALL AI LIBRARIES ?

$ pip3 install tensorflow

HOW DO WE INSTALL AI LIBRARIES ON EMBEDDED ?

Manual Install

SYSTEM SIDE CHALLENGES

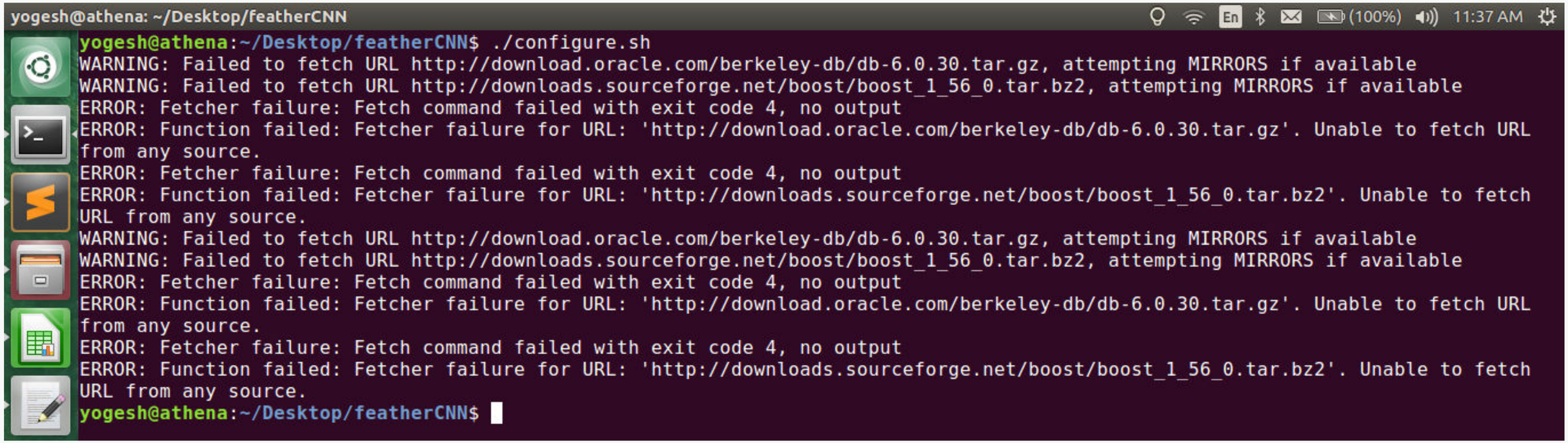
PORTING TOOLS/FRAMEWORKS/LIBRARIES FOR AI

FOR EACH TOOL/FRAMEWORK/LIBRARY

1. Fetch
2. Patch
3. Configure
4. Compile
5. Install

FETCH

Finding correct source for your packages is also a task



PATCH

Adding architecture support

Adding cross-compiler support

Adding/Removing features

CONFIGURE

Features supported by libraries/tools/frameworks need to be configured before compiling

```
bhushan@bhushan-Lenovo-ideapad-330-15IKB:~/Downloads/protobuf$ ./configure --prefix=$/home/bhushan/protobuf/
protobuf-arm --host=arm-linux CC=arm-linux-gnueabihf-gcc CXX=arm-linux-gnueabihf-g++ --with-protoc=$BASEDIR/
protobuf-host/bin/protoc
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... arm-unknown-linux-gnu
checking target system type... arm-unknown-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for arm-linux-strip... no
```

COMPILE

```
bhushan@bhushan-Lenovo-ideapad-330-15IKB:~/armnn/build$ cmake .. -DARMCOMPUTE_ROOT=/home/bhushan/ComputeLibrary
-DARMCOMPUTE_BUILD_DIR=/home/bhushan/ComputeLibrary/build -DBOOST_ROOT=/home/bhushan/boost_1_64_0/boost
-DCAFFE_GENERATED_SOURCES=/home/bhushan/caffe/build/src/caffe/proto -DBUILD_CAFFE_PARSER=1 -DARMCOMPUTENEON=1
-DARMCOMPUTECL=1
-- Boost version: 1.64.0
-- Found the following Boost libraries:
--   unit_test_framework
--   system
-- Tensorflow parser support is disabled
-- Tensorflow Lite parser support is disabled
Including backend common library into the build: /home/bhushan/armnn/src/backends/backendsCommon/common.cmake
Including backend common library into the build: /home/bhushan/armnn/src/backends/aclCommon/common.cmake
Including backend into the build: /home/bhushan/armnn/src/backends/reference/backend.cmake
Including backend into the build: /home/bhushan/armnn/src/backends/neon/backend.cmake
Including backend into the build: /home/bhushan/armnn/src/backends/cl/backend.cmake

Adding object library dependency to UnitTests: armnnClBackendUnitTests
-- Configuring done
CMake Error at cmake/Utils.cmake:41 (add_library):
  Cannot find source file:

    /home/bhushan/caffe/build/src/caffe/proto/caffe/proto/caffe.pb.cc

  Tried extensions .c .C .c++ .cc .cpp .cxx .m .M .mm .h .hh .h++ .hm .hpp
  .hxx .in .txx
Call Stack (most recent call first):
  CMakeLists.txt:76 (add_library_ex)


-- Generating done
-- Build files have been written to: /home/bhushan/armnn/build
```

```
bhushan@bhushan-Lenovo-ideapad-330-15IKB:~/armnn/build$ make
[  5%] Built target armnnUtils
[ 15%] Built target armnnClBackendWorkloads
[ 19%] Built target armnnBackendsCommon
[ 20%] Built target armnnAclCommon
[ 21%] Built target armnnRefBackend
[ 41%] Built target armnnRefBackendWorkloads
[ 43%] Built target armnnNeonBackend
[ 52%] Built target armnnNeonBackendWorkloads
[ 54%] Built target armnnClBackend
CMakeFiles/armnn.dir/build.make:1704: *** missing separator (did you mean TAB instead of 8 spaces?).
CMakeFiles/Makefile2:113: recipe for target 'CMakeFiles/armnn.dir/all' failed
make[1]: *** [CMakeFiles/armnn.dir/all] Error 2
Makefile:127: recipe for target 'all' failed
make: *** [all] Error 2
```

INSTALL

Libraries should be installed correctly!!

Also ran all the installations commands from *environment_setup.txt* and completed tensorflow installation.

In the last part of the *environment_setup.txt* file under validating everything, after I run

> *import tensorflow*

```
2019-01-22 23:07:41.156208: F tensorflow/core/platform/cpu_feature_guard.cc:37] The TensorFlow library
was compiled to use SSE4.2 instructions, but these aren't available on your machine.
Aborted (core dumped)
```

I am getting the above error

SYSTEM SIDE CHALLENGES

DEPENDENCIES

EACH TOOL/FRAMEWORK/LIBRARY

on an average has 4-5 dependencies

FOR EACH DEPENDENCY

REPEAT all the above **Process**

SYSTEM SIDE CHALLENGES

HETROGENEOUS COMPUTING

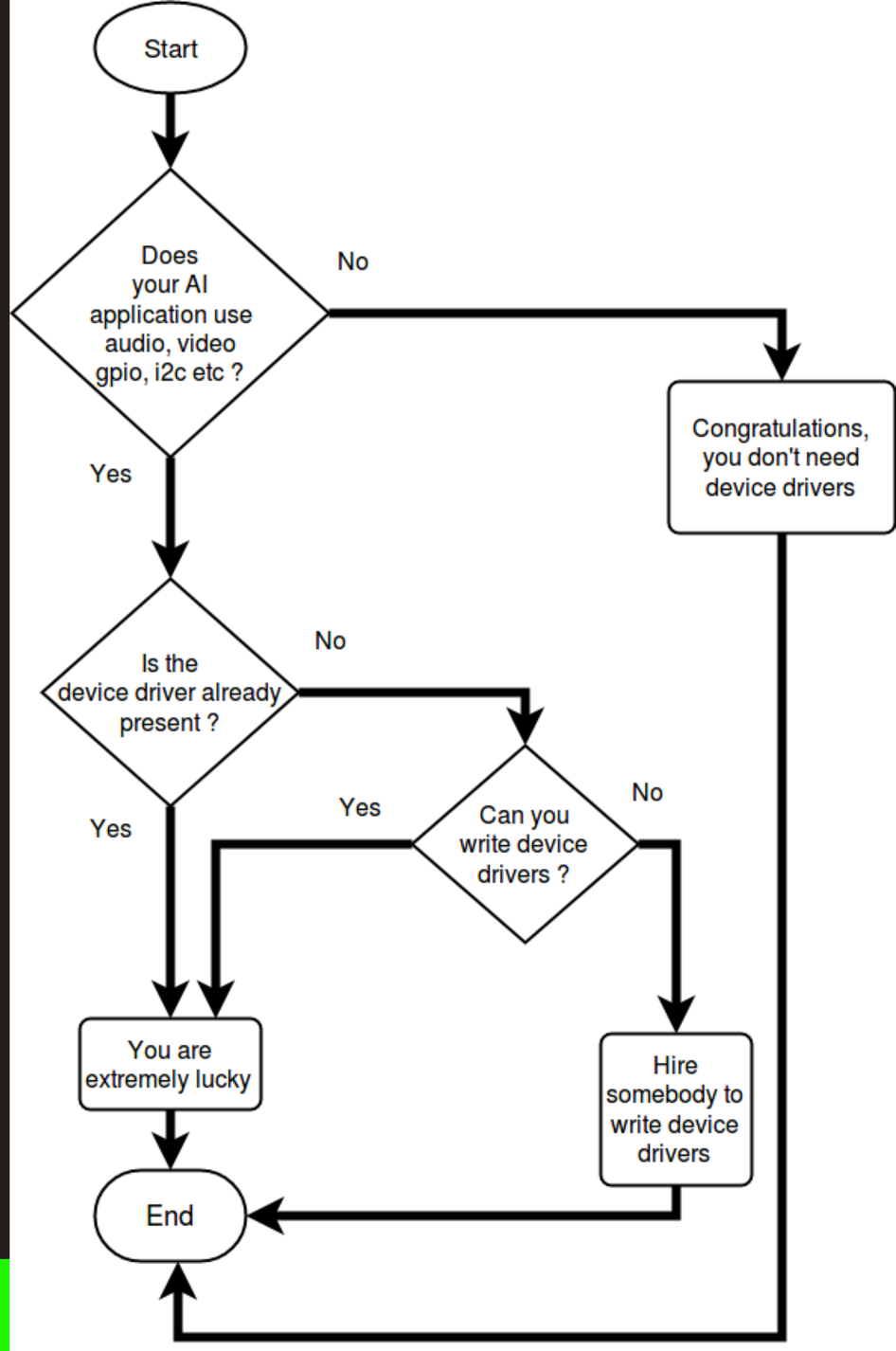
EMBEDDED SYSTEM

has various
computing hardware

For hetrogeneous
computing we need

SYSTEM SIDE CHALLENGES

DEVICE DRIVERS



All this effort just to make your AI application WORK

To optimize your Application
you need to put in more effort

18% OF DEVELOPERS

prefer base system

82% OF DEVELOPERS

prefer ready environment
for AI development

WE WANT YOU TO FOCUS ON YOUR APPLICATION

LEAVE ALL YOUR SYSTEM ENGINEERING ON

SHUNYA O/S

ALL YOU NEED TO LEARN IS

GET AI

One command integrates architecture tuned AI framework

within few minutes

PREPARE YOUR EMBEDDED SYSTEM TO RUN AI

Download

Link - team.iotiot.in/downloads

Burn

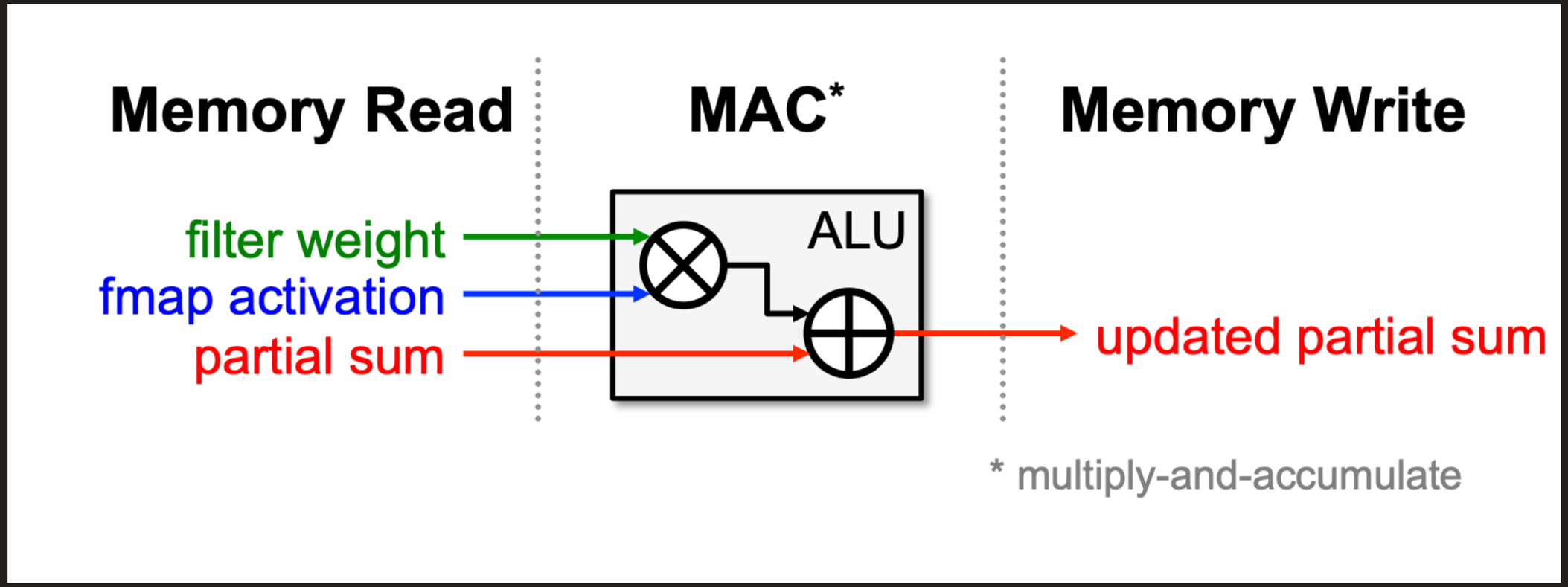
Power on

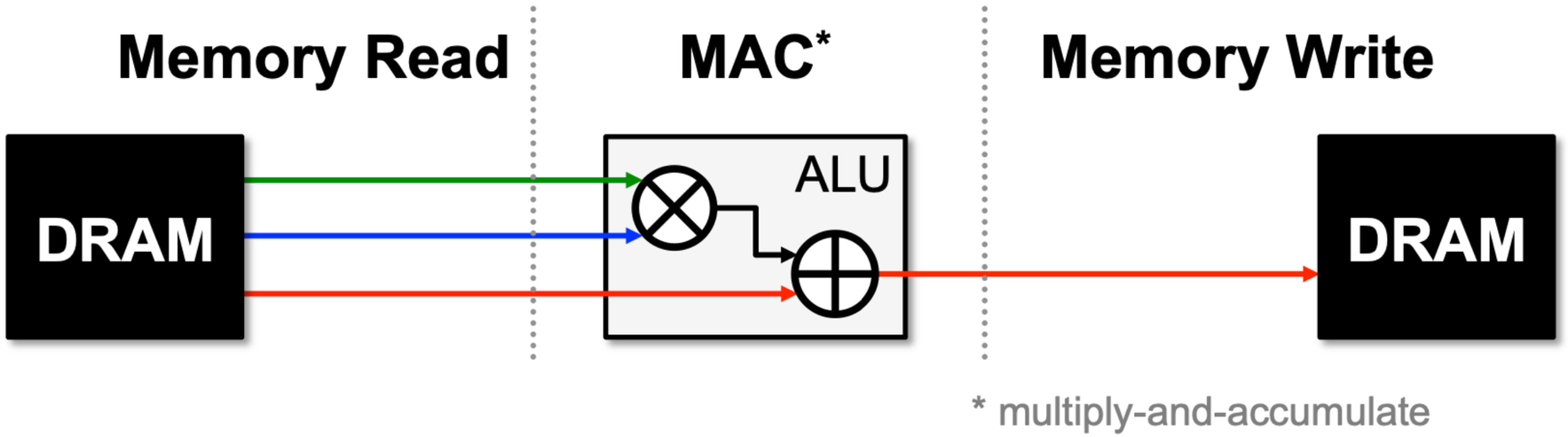
get-ai

EMBEDDED SYSTEM
READY TO RUN
AI APPLICATION

HOW TUNED IS THIS AI FOR EMBEDDED

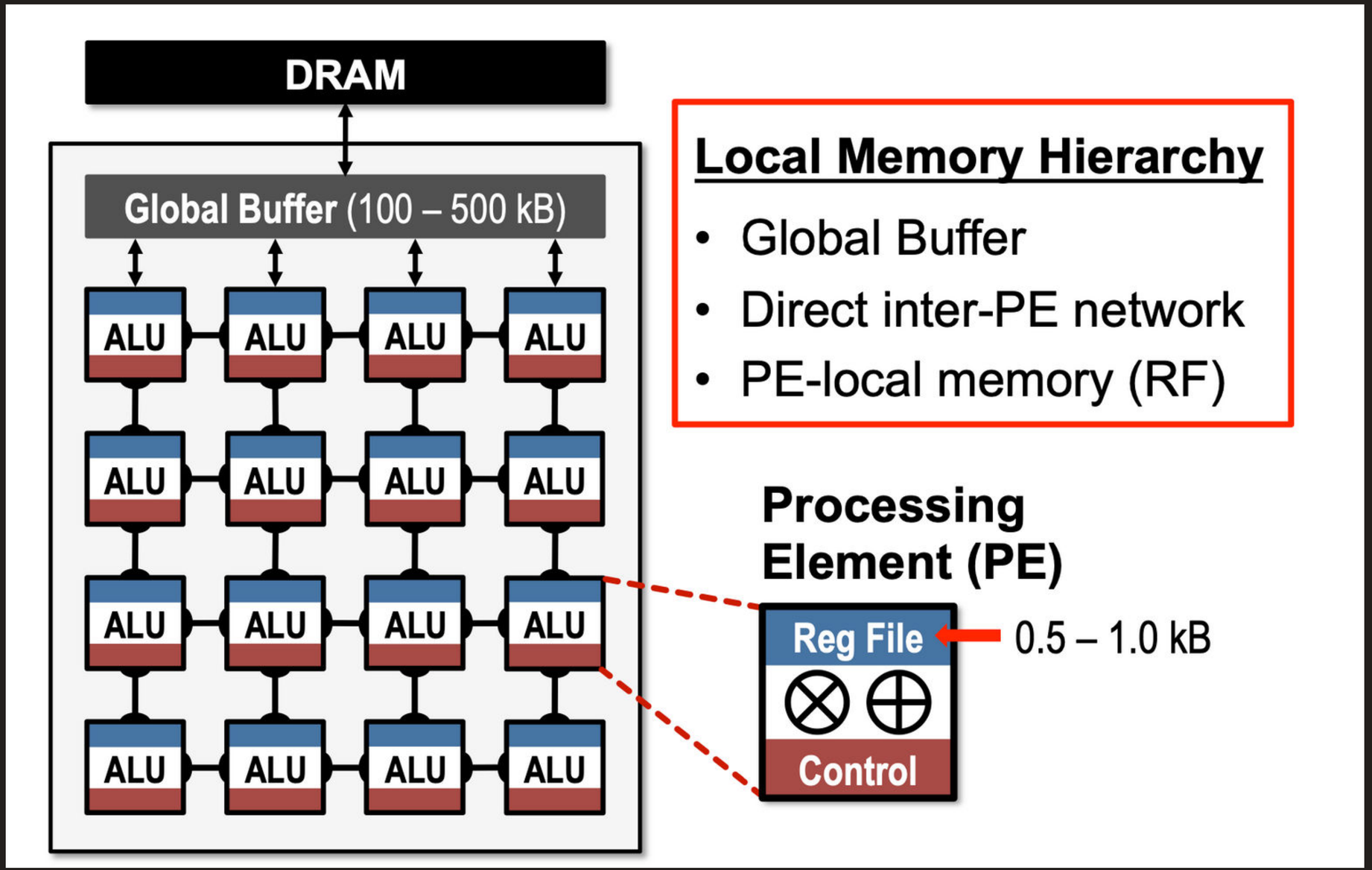
MEMORY AS BOTTLENECK

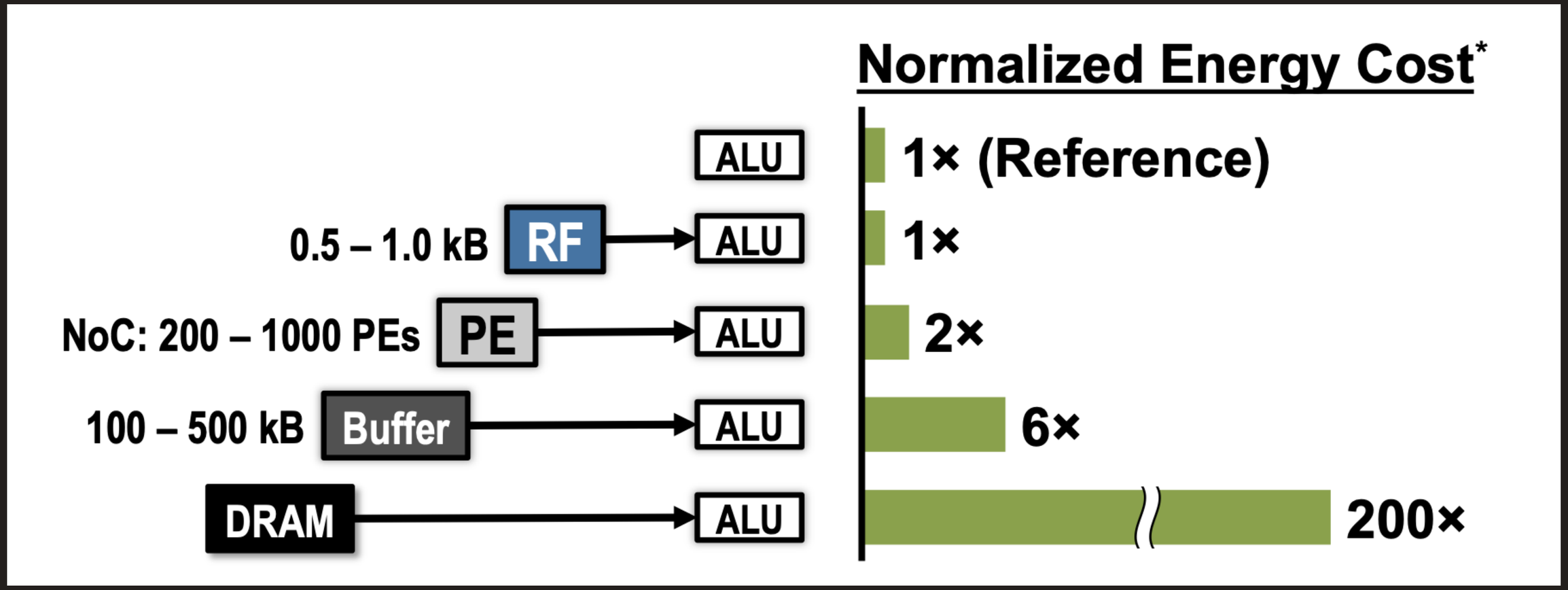




Worst Case: all memory R/W are **DRAM** accesses

- Example: AlexNet [NIPS 2012] has **724M** MACs
  → **2896M** DRAM accesses required


DRAM
Global Buffer (100 – 500 kB)
ALU
Local Memory Hierarchy
• Global Buffer
• Direct inter-PE network
• PE-local memory (RF)
Processing Element (PE)
Reg File
0.5 – 1.0 kB
Control


Normalized Energy Cost*
ALU
1× (Reference)
0.5 – 1.0 kB
RF
ALU
1×
NoC: 200 – 1000 PEs
PE
ALU
2×
100 – 500 kB
Buffer
ALU
6×
DRAM
ALU
200×

SHUNYA NEEDS 0.4 A
WHILE INTEL VPU NEEDS 4A

THANK YOU

www.shunyaos.org