Digital Signal Processing (DSP) and Data Science

RISC-V based Anasim Processor for FPGA (Soft Processor) and ASIC (Hard Processor) is optimized for Digital Signal Processing (DSP) and Artificial Intelligence (AI). For more details on RISC-V, please visit https://riscv.org/ .

Please visit http://www.differencebetween.net/technology/difference-between-asic-and-fpga/ to know more on FPGA and ASIC.

Our Proprietary Anasim Processor is improved upon RISC-V and extended with DSP (VLSI based) and AI capabilities which is suitable for FPGA and ASIC implementations. The Aansim Processor is aimed at AI and DSP related applications at cloud, in IoT and Edge Devices. We understand that to lower costs and volume production, customers have to go for ASIC ulitmately. Indian Central and State Governments may choose SCL (http://www.scl.gov.in/) as their semiconductor partner to fabricate AI Chips (ASIC) based on our Proprietary Anasim Processor IPCores; for privately owned organizations we have prototyping and production arrangements with international semiconductor fabrication facilities, Governments of other friendly countries may also choose SCL as their semiconductor partner with SCL's prior approval. As per our policy we charge for Proprietary IPCores licensing costs and new IPCore design, development & testing costs; hardware and fabrication charges are directly born by the customers resulting in significantly lower per chip costs.

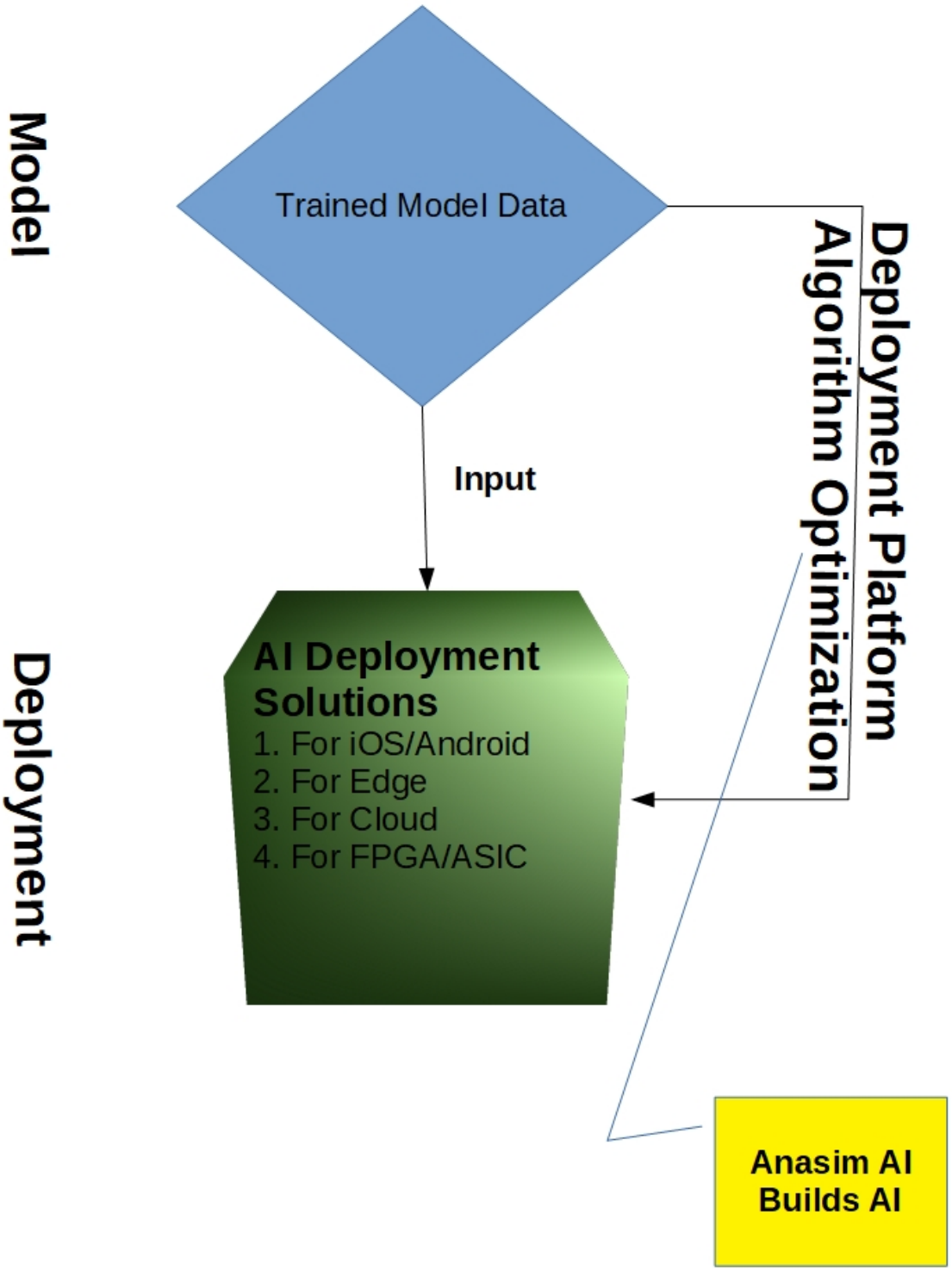
Convert your AI Research into AI Chip, download our Proprietary Free of Charge Anasim App (to be made available soon) from Apple iStore to automate your AI Chip Development Process. Apart from you need not to spend thousands of US Dollars on licenses of Semiconductor Design, Synthesis, Verification and Validation Tools and on hiring highly paid teams of highly skilled Engineers for months to run these Tools for you in order to design AI Chip for you, the Anasim App lets you:

1. Test Trained Data Model on your iPhone/iPad. You may go through the pre-requisites given below to obtain you Trained Data Model.

2. Obtain Anasim AI Soft Processor based on Your Trained Data Model for your FPGA Board. Both can be flashed with the FPGA flashing tools mentioned at the bottom of the page. Ship your FPGA board (single piece) for testing to us if it is not Xilinx Zynq 7000 Zybo Board. See Anasim AI Builds AI Diagram for details.

3. Connect, Test and Debug your FPGA Board through JTAG USB Cable.
4. Obtain technical support from us and resolve your technical issues.
5. Obtain commercial and non-commercial/academic licenses for our IPCores.
6. Know semiconductor fabrication facilities production schedule including for (Multi Project Wafer) MPW.
7. Obtain commercial quotes from fabrication facilities.
8. Make volume and prototype ASIC fabrication order.
9. Make payment for the licenses and the order.
10. Know order and delivery status.

Anasim AI Builds AI



Please go through AI-Development-and-Deployment-Process.pdf, AI-Solutions.pdf and other files in the DSP-and-Data-Science repository and other repositories for more details.

We hope that following material and the material given in this repository will also be helpful for data scientists in general apart from our Anasim App.

Pre-requisites:

Visitors may soon expect several regularly updated Caffe Model Zoo trained models optimized ready to run example HDL SoC codes for Xilinx Zynq 7000 Zybo Board under GPL License for non-commercial/academic use in this repository. You may also train your models by following the procedure given below:

Please visit http://caffe.berkeleyvision.org/ for more information on Caffe and visit http://docker.com/ for more information on docker container. Please visit https://aws.amazon.com/docker/ to run docker image at Amazon AWS and visit https://docs.microsoft.com/en-us/azure/docker/ to run docker image at Microsoft Azure cloud services. Our Caffe installations are tested with NVIDIA CUDA 10.0 and NVIDIA cuDNN 10.0 on Ubuntu 18.04 LTS, so please do not expect our free support on Caffe other than this.

Please visit https://github.com/BVLC/caffe to download latest version of Caffe to build from scratch or visit http://caffe.berkeleyvision.org/install_apt.html `for Caffe installation steps for Ubuntu.`

According to this Caffe Github Site, you may run Caffe docker image as follows:

Running an official image

You can run one of the version:

```
docker run -ti bvlc/caffe:cpu caffe --version
```

or for GPU support (You need a CUDA 10.0 capable driver and nvidia-docker):

```
nvidia-docker run -ti bvlc/caffe:gpu caffe --version
```

You might see an error about libdc1394, ignore it.

Docker run options

By default caffe runs as root, thus any output files, e.g. snapshots, will be owned by root. It also runs by default in a container-private folder.

You can change this using flags, like user (-u), current directory, and volumes (-w and -v). E.g. this behaves like the usual caffe executable:

```
docker run --rm -u $(id -u):$(id -g) -v $(pwd):$(pwd) -w $(pwd) bvlc/caffe:cpu caffe
train --solver=example_solver.prototxt
```

Containers can also be used interactively, specifying e.g. `bash` or `ipython` instead of `caffe`.

```
docker run -ti bvlc/caffe:cpu ipython
import caffe
...
```

The caffe build requirements are included in the container, so this can be used to build and run custom versions of caffe. Also, `caffe/python` is in PATH, so python utilities can be used directly, e.g. `draw_net.py`, `classify.py`, or `detect.py`.

Building images yourself

Examples:

```
docker build -t caffe:cpu cpu
docker build -t caffe:gpu gpu
```

You can also build Caffe and run the tests in the image:

```
docker run -ti caffe:cpu bash -c "cd /opt/caffe/build; make runtest"
```

Study Material:

http://caffe.berkeleyvision.org/

http://caffe.berkeleyvision.org/tutorial/

http://caffe.berkeleyvision.org/gathered/examples/imagenet.html

http://shengshuyang.github.io/A-step-by-step-guide-to-Caffe.html

http://vision.stanford.edu/teaching/cs231n/slides/2015/caffe_tutorial.pdf

https://www.panderson.me/images/Caffe.pdf

http://3dvision.princeton.edu/courses/COS598/2015sp/slides/Caffe/caffe_tutorial.pdf

Flashing Tools:

Free of Cost Quartus II Stand-Alone Programmer from Intel may be downloaded from following link to flash Altera FPGA based boards:

http://fpgasoftware.intel.com/16.1/?product=qprogrammer#tabs-4

Free of Cost Vivado Lab Solutions from Xilinx may be downloaded from following link to flash Xilinx 7 Family FPGA based boards:

https://www.xilinx.com/support/download.html