**Referring to the remarks:**

Meeting:

* Pick one board – run a simulation.
* Go from low to high level language.
* Compare inputs - outputs, test bench, take it as a spec check (the low level) it outputs the same outputs, just to be sure I’m running the same algorithm and nothing has changed.
* Compare the results as an indicator that the algorithms are the same, put some threshold above 5% for instance (accuracy).
* Iverilog, verilator – simulators with test bench.
* V Block diagram
* Go to more recent papers.
* Design time design complexity, resource area estimate, clock frequency, how many LUTs are taken.

Python translators:

<http://www.myhdl.org/>

<https://github.com/UCSBarchlab/PyRTL>

<https://web.csl.cornell.edu/pymtl2015/>

<http://www.pynq.io/>

<https://github.com/evlog/SysPy>

<https://www.eetimes.com/document.asp?doc_id=1329857>

C translators:

<http://www.cl.cam.ac.uk/~djg11/ctovpages/>

<https://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html>

(SVM)

Python Code:

<https://github.com/LasseRegin/SVM-w-SMO>

<https://github.com/soloice/SVM-python>

<https://github.com/hejujie/svm_smo>

C Code:

<https://github.com/livey/svm-embed>

<https://github.com/mbarga/mySVM>

Hardware code:

<https://github.com/venkatesh20/SVM>

<https://github.com/custom-computing-ic/SVM>

PPT:

<https://llvm.org/devmtg/2010-11/Rotem-CToVerilog.pdf>

Resources and References:

1. <https://pdfs.semanticscholar.org/9e5a/a50e064c30c23852dd6e66aba01bc66e1849.pdf>
2. [Hardware Implementations of SVM on FPGA: AState-of-the-Art Review of Current Practice](https://pdfs.semanticscholar.org/5470/547ba140cf8be54721ecac9d24a4f9f4b6b5.pdf)
3. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4629911>
4. [An Extended Architecture to Optimize Execution Time of 3D Image Processing Deflectometry Algorithm Using FPGA](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8120617)
5. [An FPGA-Based Hardware Implementation of Visual based Fall Detection](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6863065)
6. [Implementing Image Processing Algorithms in FPGA Hardware](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6716446)
7. [Implementing Image Applications on FPGAs'](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1047845)
8. [A Low-Cost FPGA-based SVM Classifier for Melanoma Detection](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7843526)
9. [SVM WITH OPENCL: HIGH PERFORMANCE IMPLEMENTATION OF SUPPORT VECTOR MACHINES ON HETEROGENEOUS SYSTEMS](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7351622)
10. [Implementing Legacy-C Algorithms in FPGA Co-Processors for Performance Accelerated Smart Payloads](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4526522)
11. \*\*[Evaluating Rapid Application Development with Python for Heterogeneous Processor-based FPGAs](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7966663)
12. [High Level Synthesis: Where Are We? A Case Study on Matrix Multiplication](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6732298)
13. [PyMTL: A Unified Framework for Vertically Integrated Computer Architecture Research](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7011395)
14. [SysPy: using Python for processor-centric SoC design](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5724624)
15. [MyHDL: a Python-Based Hardware Description Language](http://www.linuxjournal.com/article/7542)
16. [PYTHON AS A HARDWARE DESCRIPTION LANGUAGE: A CASE STUDY](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5782635)
17. [LegUp: High-Level Synthesis for FPGA-Based Processor/Accelerator Systems](http://delivery.acm.org/10.1145/1960000/1950423/p33-canis.pdf?ip=160.39.4.171&id=1950423&acc=ACTIVE%20SERVICE&key=7777116298C9657D%2ECCAFA7F43E96773E%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1521214808_58c24ee9bd58e23a3e93595e46059561)
18. [High-Level Synthesis for FPGAs: From Prototyping to Deployment](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5737854)
19. [High level synthesis methodology from C to FPGA used for a network protocol communication.](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1311103)
20. [FPGA-Based CNN Inference Accelerator Synthesized from Multi-Threaded C Software](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8226056)