FPGA Implementation of a Support Vector Machine (SVM)

May 31, 2005
Phil Tomson
ECE590

SVM Background

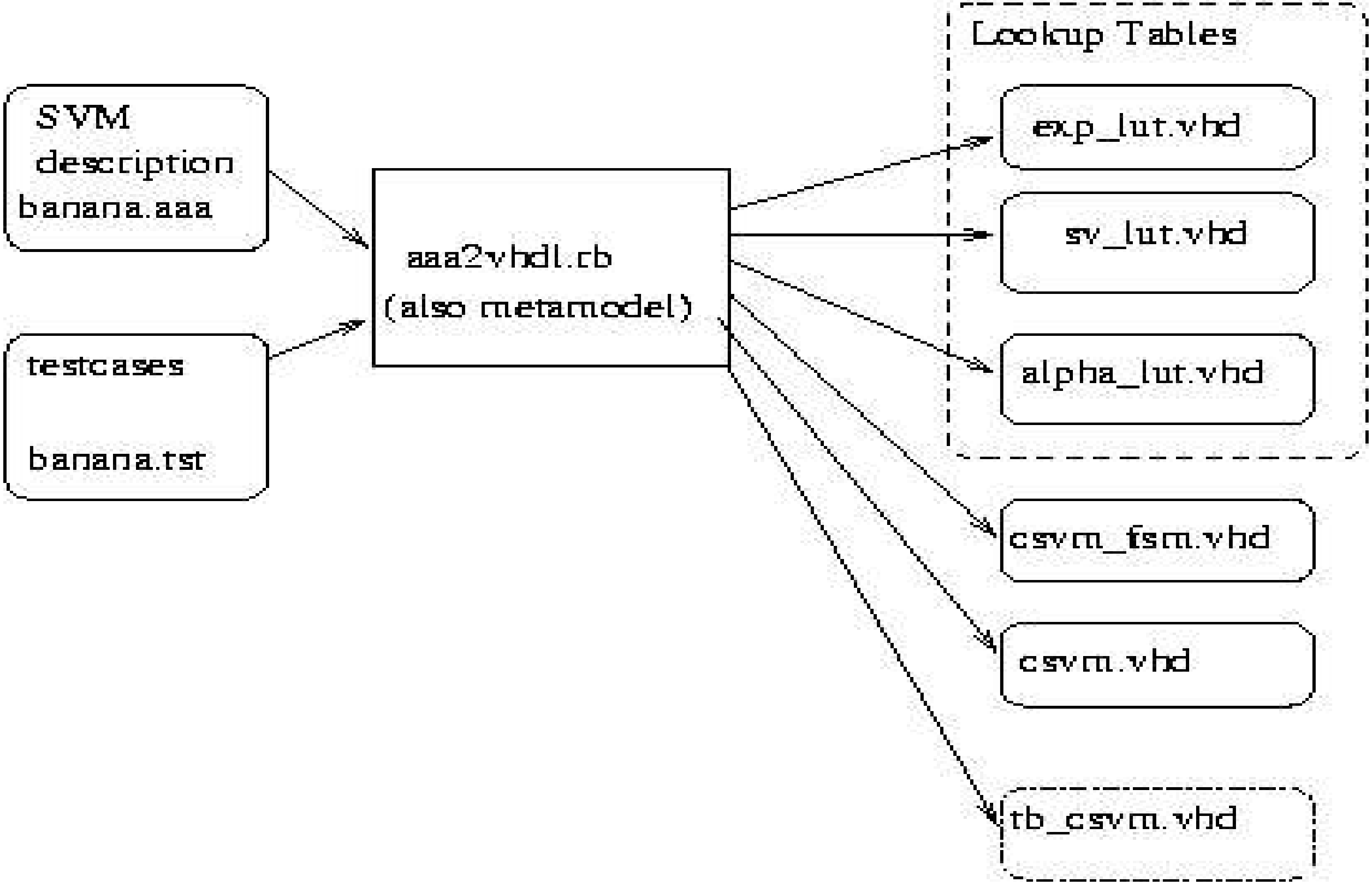
- Machine learning algorithm
- Classifiers
- Applications: OCR, speech recognition,pattern recognition, etc.
- SVM formula (Gaussian kernel):

$$bias + \sum_{i=0}^{NSV} alpha[i] * e^{-k*\gamma}$$

$$where: k = \langle \chi_i , \chi \rangle$$

aaa2vhdl

- Translate textual description of SVM (.aaa file) to VHDL.
- Block Diagram:

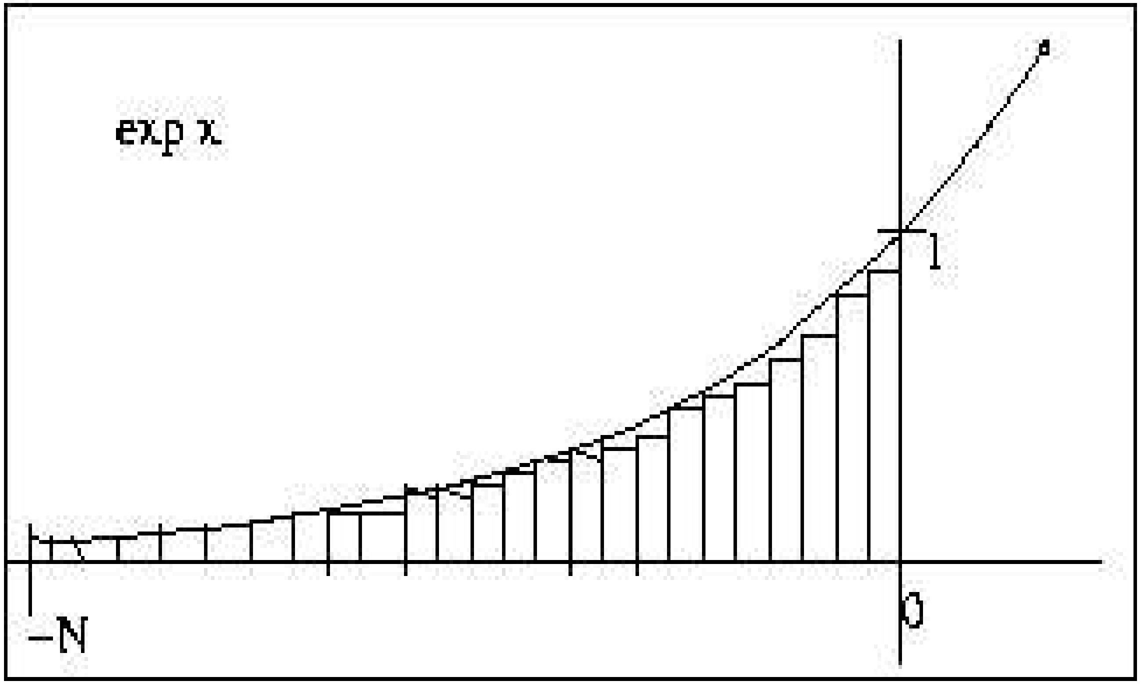


MetaModel levels

- Level 1: Implemented entirely using Ruby's built-in math functions (could be in C as well)
- Level 2: Implemented in Ruby using a FixedPt package which I developed. All math is done via fixed point ops and exp() is implemented as lookup table (array) indexed by an integer. (could be considered a micro-architectural model except for the FSM)
- Hybrid: Same as level 2, but exp() lookup table was implemented in VHDL. The Ruby meta model communicates with the VHDL simulator via pipes (Ruby model actually requests data from exp lookup table implemented in VHDL).

Hardware implementation issues

- Exp()
  - Lookup table issues:
    - Resolution (128 entries, 16 bits fractional)
    - Range (-12.25 to 0)
    - Experimentally determined



Validation

- 4900 testcases for banana SVM
  - 15 failures on the 'golden' level 1 meta model.
  - 13 to 15 errors in the level 2 meta-model
  - Generate testbench with the 4900 testcases to test the VHDL implementation (simulation could take a long time)

Current Status/TODO

- VHDL for Lookup tables (exp, alpha, sv) are generated.
- State machine VHDL code generated (csvm_fsm.vhd)
- Toplevel VHDL file (csvm.vhd) is generated, but still in debugging. (many problems fixed, but issues remain)
- Still need to generate the testbench.
- TODO: run final design through ISE to get area and timing info

Current Status/TODO cont.

- TODO(time permitting): Generate LUTs as case statements (not using RAM modules) and compare area/speed. Speed should be much improved without RAMs (RAMs require waiting for an extra clock period for data, case statement wouldn't need this extra 'wait' state), but might require too much area (1 128X18bit LUT, 1 244X18bit LUT, 1 488X18bit LUT). This will require that the FSM controller be parametizable to remove extra wait states which the RAM requires.