Centre de Physique des Particules de Marseille

LSTM on FPGA

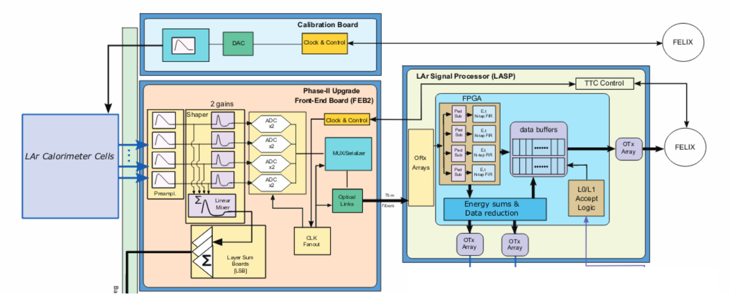
IN2P3/IRFU Machine Learning workshop

16 March 2021

Etienne FORTIN

Georges Aad, Thomas Calvet, Nemer Chiedde, Lauri Laatu, Emmanuel Monnier

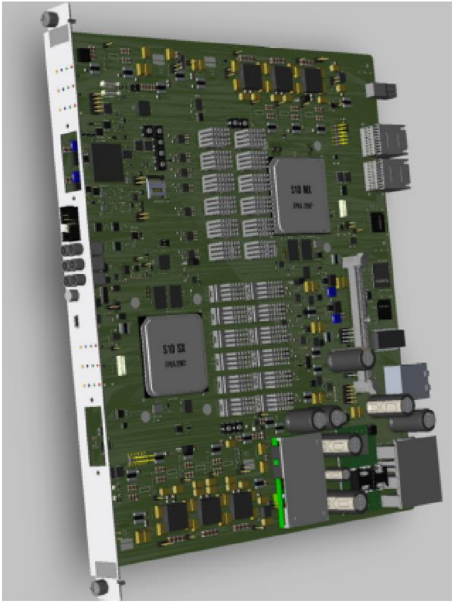
LAr electronics :Phase 2 upgrade



- Phase 2 upgrade of the LAr calorimeter for HL-LHC
  - Replacement of the full readout electronics chain
  - Improve real time energy reconstruction at high pile-up
- New back end board (LASP)
  - Receive digitized sampled data at 40MHz
  - Compute the deposited energies at 40MHz

LAr back-end electronics: LASP board

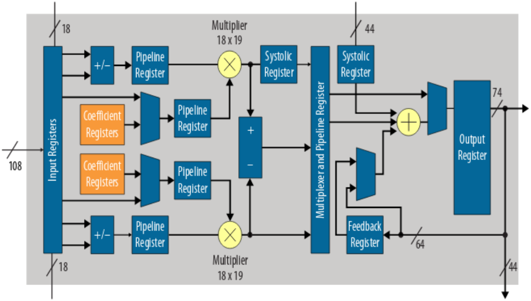
- ATCA board with 2 Stratix-10 FPGA
- Each FPGA processes:
  - Up to 500 channels
  - 1 Tb/s
- Full system needs 200 boards
- First prototype board available in September



FPGA

A Field Programmable Gate Array is a chip which contains an array of :

- ALUT/ALM, logic blocks
- RAM, memory blocks
- DSP, optimized multiplication blocks
- FIFO, buffering blocks

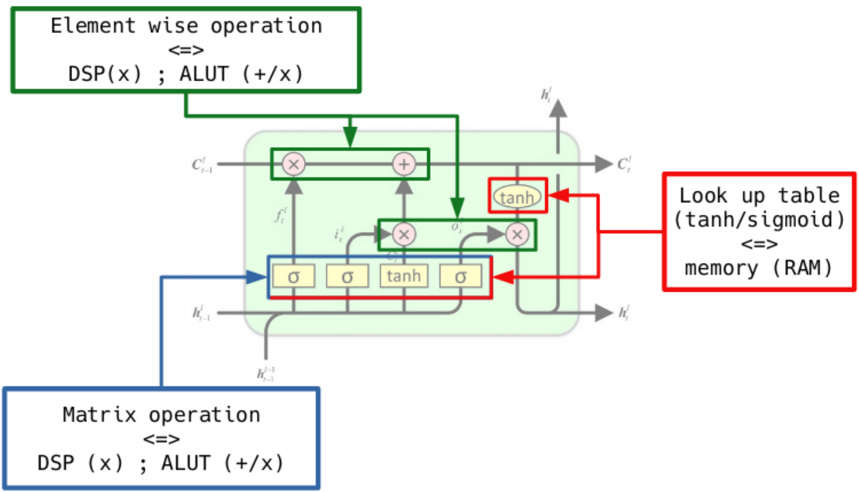


Stratix 10 DSP block: 18x19 multiplier

FPGA firmware/gateware design key parameters:

- Resource usage
- II: Initiation Interval: number of clock cycle between two inputs
- Latency: number of clock cycle for output to be calculated
- Frequency: frequency of the clock

LSTM Implementation



Element wise operation <=> DSP(x) ; ALUT (+/x)

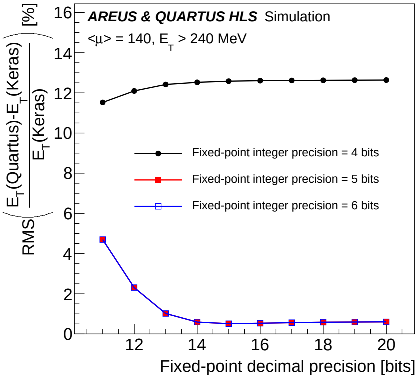
Look up table (tanh/sigmoid) <=> memory (RAM)

Matrix operation <=> DSP (x) ; ALUT (+/x)

First prototype fully implemented in Quartus High Level Synthesis (HLS)

Optimization 1: data fixed point representation

Data represented in fixed point with fixed number of bits



Sliding window LSTM resources

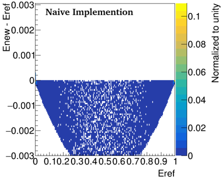
Data format	ALUT	FIFO	DSP
6-12	7.8%	6 %	11.3%
6-13	7.5%	6.3 %	12.8%
6-14	7.5%	6.6 %	27%
5-12	6.6%	4.7 %	11.3%
5-13	7.5%	6.1 %	12.8%
5-14	7.2%	6.4 %	14%

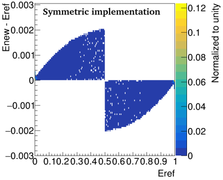
Choose 5-13: best compromise accuracy/resources

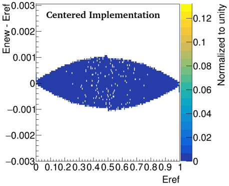
Optimization 2: Look Up Table

Implemented two LUT optimization:

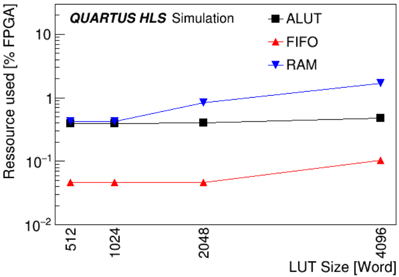
- Using symmetry properties of the function: f(-x) = -f(x)
- Using bin centers to compute the LUT output







The precision of the LUT is driven by the number of words in it. Doubling the number of words double the precision. But it also increase the resources usage



Summary of resource optimization

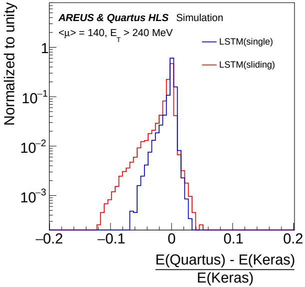
- Optimization 3: Adaptive fixed point representation
  - LUT outputs strictly between -1 and 1: no need for integer bits
- Other optimization ongoing

FPGA performance results for Sliding window LSTM

	Default	Optimised LUT	Optimised + Adaptative LUT
Fixed point	5-13	5-13	5-13
LUT words	4096	2048	2048
Resolution	-0.006 +/- 0.016	-0.004 +/- 0.015	-0.004 +/- 0.015
ALUT	160000(9%)	152809 (8.1%)	140089 (7.5%)
FIFO	258576(7%)	249993 (6.69%)	228365 (6.1%)
RAM	966(8%)	486 (4%)	486 (4%)
DSP	845(15%)	844 (15%)	739 (12.8%)

Results

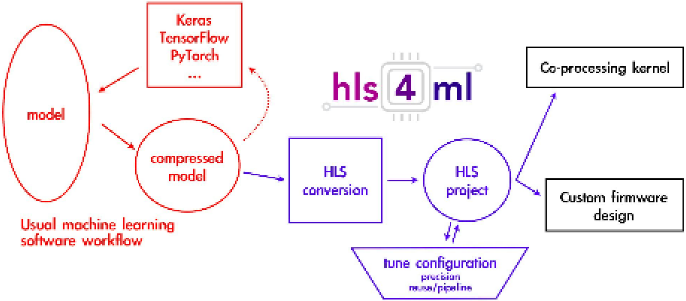
- Simulation with Quartus 20.4 and Modelsim 10.7c
- Good compatibility Firmware and Software
- Moderate FPGA resource usage and High speed processing
- Implementation ongoing in HLS4ML



Network	Frequency	Latency	II	Resource Usage			
	F_{max} [MHz]	clk_{core} cycles		#DSPs		#ALMs	
LSTM (single)	560	220	220	176	3.1%	18079	1.9%
LSTM (sliding)	517	363	1	738	12.8%	69892	7.5%

HLS4ML: concept

- Open source software designed to facilitate the implementation of machine learning into FPGA
  - Convert trained keras model into HLS and FPGA firmware

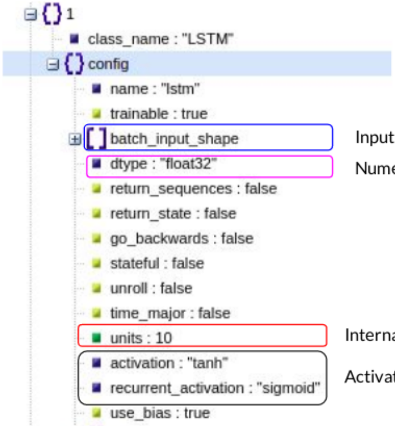


https://fastmachinelearning.org/hls4ml/

HLS4ML: LSTM implementation

- Integration of our LSTM HLS into HLS4ML
  - Configurable from Keras model

Keras JSON:



```
{} 1
   class_name : "LSTM"
   {} config
      name : "lstm"
      trainable : true
      [] batch_input_shape
      dtype : "float32"
      return_sequences : false
      return_state : false
      go_backwards : false
      stateful : false
      unroll : false
      time_major : false
      units : 10
      activation : "tanh"
      recurrent_activation : "sigmoid"
      use_bias : true
```

Input dimension

Numeric precision

Internal dimension

Activations functions

- Activation function implemented
- Other parameters ongoing

Conclusion

- LSTM Network firmware model developed in Quartus HLS
- Optimized network implementation to reduce resources usage while keeping good performance
  - Fixed-point precision
  - Look Up Tables for activation functions
- Network output validated with firmware simulation and compared to keras results
- Challenging to use LSTM for the full LAr data processing at 40 MHz
  - Needed for L1 trigger application
  - Investigation ongoing for simpler types of RNN (vanilla RNN)
  - LSTM can still be used for a subset of the data to improve overall resolution
- LSTM implementation in HLS4ML for Intel FPGAs is advanced
  - Suitable for high rate processing of time series data
- Test on real hardware started with Stratix 10 Devkit
  - Plan to test this year on LASP prototype