FPGA Implementation of Boolean Neural Networks using UML

Roman Kohut, Bernd Steinbach, Dominik Fröhlich

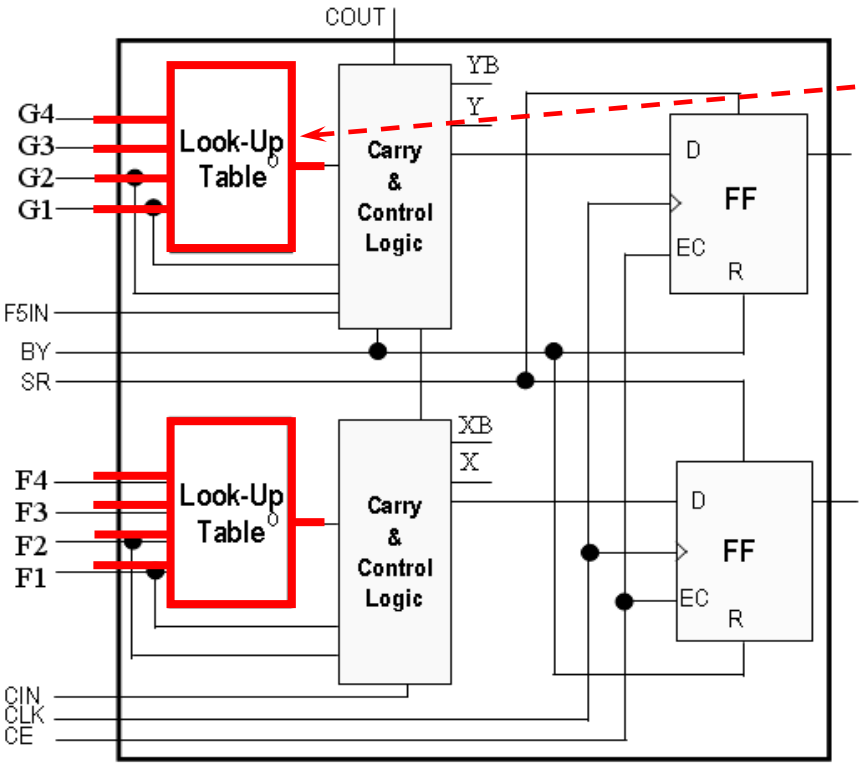
Freiberg University of Mining and Technology
Institute of Computer Science
Freiberg (Sachs), Germany

Outline

- Introduction
- Boolean Neural Networks
- UML-Models
- Experiment results
- Conclusion

Introduction

- FPGAs



$$y = f(\boldsymbol{x}),$$
$$\boldsymbol{x} = \{x_1, x_2, \ldots, x_{Nx}\}, \qquad N_x \leq 4$$

5 { *slice* → $N_x \leq 5$

6 { *CLB* → $N_x \leq 6$

Slice structure

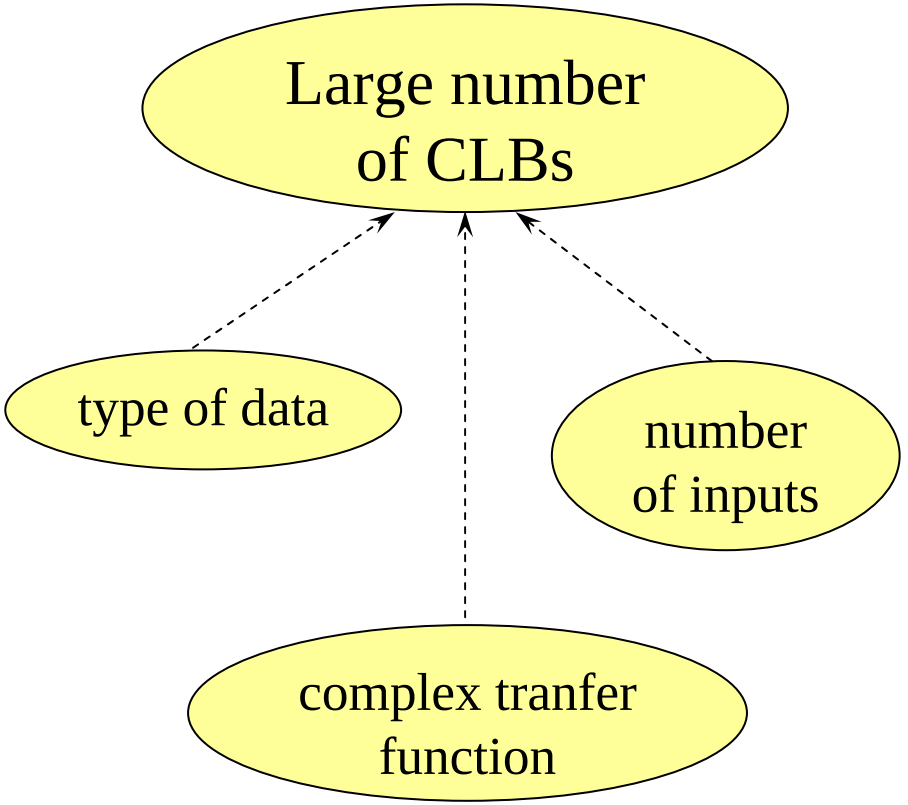
Introduction

- The Problem

• Connectivity problems,
• Structured problems
  • limited number of logic gates
  • and interconnections,
• On-Chip learning problems (sequential computations),
• …

Large number of CLBs (10^{th}-100^{th}) are required for **one single neuron**

GANGLION - (640-784) CLBs,
Gschwind NN - 22 CLBs,
Xilinx-NN - 51 CLBs,
Hopfield NN - 26 CLBs.



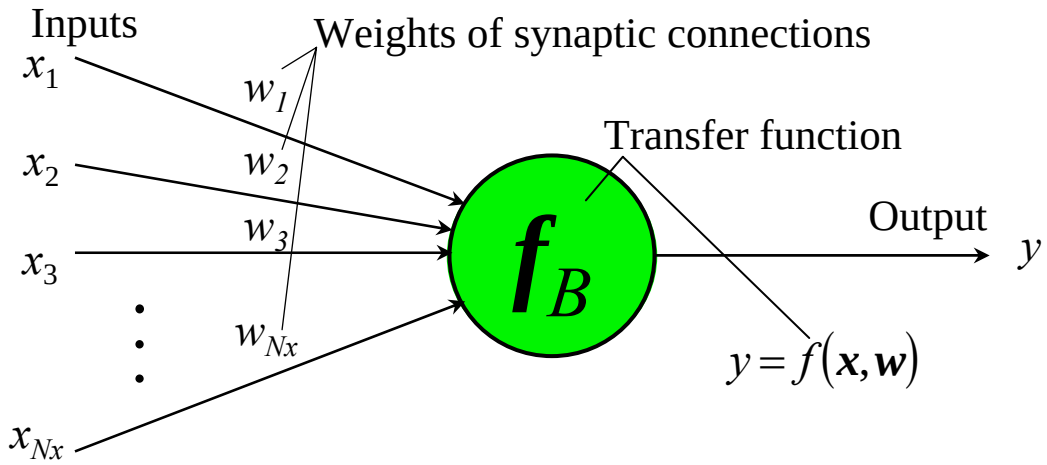
Kohut, R., Steinbach, B., Fröhlich, D. 7th International Workshop on Boolean Problems, Freiberg, Sept. 21-22, 2006

Boolean Neural Networks

- Boolean Neuron

$$y = f(\boldsymbol{x}, \boldsymbol{w})$$

$$y_B = f_B(\boldsymbol{x}_B, \boldsymbol{w}_B)$$



General structure of Boolean neuron

$$\boldsymbol{x}_B = \{x_1, x_2, ..., x_{N_x}\} \qquad x_i \in \{0,1\}$$

$$\boldsymbol{w}_B = \{w_1, w_2, ..., w_{N_x}\} \qquad w_i \in \{0,1\}$$

f_B - Boolean transfer function

y_B - output signal

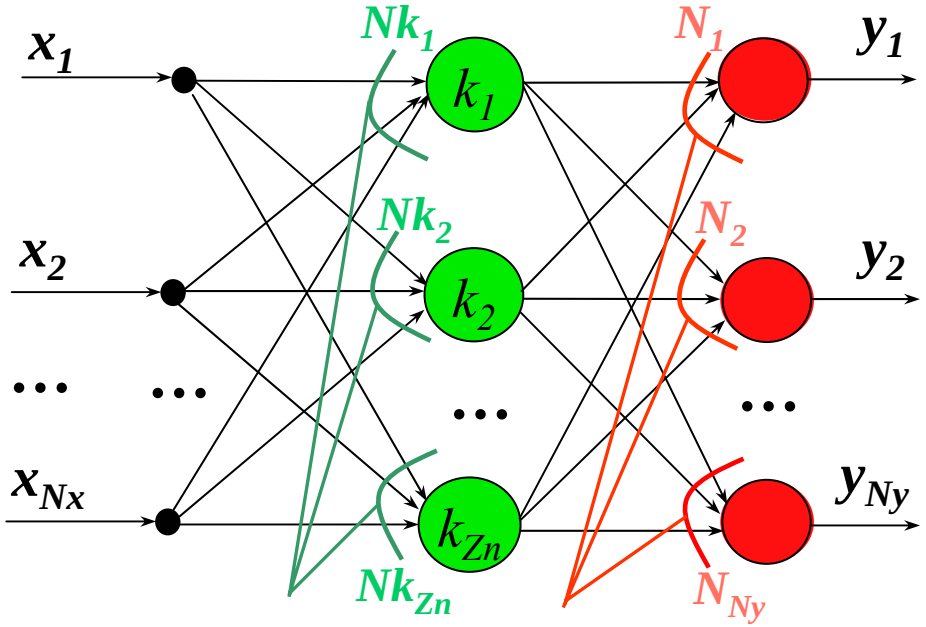
$f_B, y_B \in \{0,1\}$

Advantages of the BN:

- speeding up of calculation significantly,
- reduction of necessary memory size,
- possibility to map the BN into one single CLB of FPGAs.

Boolean Neural Network

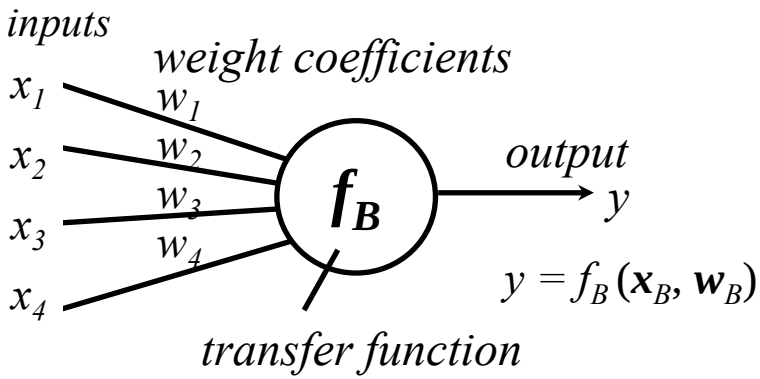
- Structure

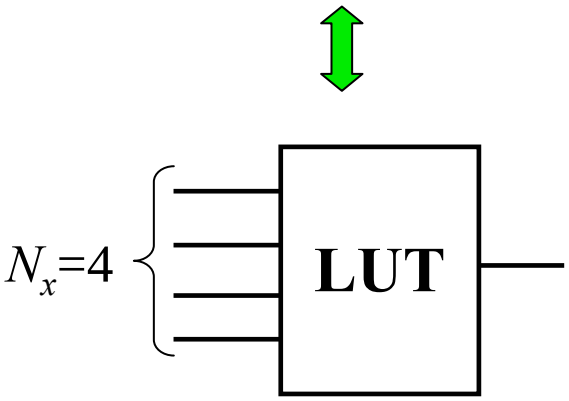


LUT: Nk_{Zn}, $N_{Ny} \leq 4$
Slice: Nk_{Zn}, $N_{Ny} \leq 5$
CLB: Nk_{Zn}, $N_{Ny} \leq 6$

Training algorithm

- LUT of CLB

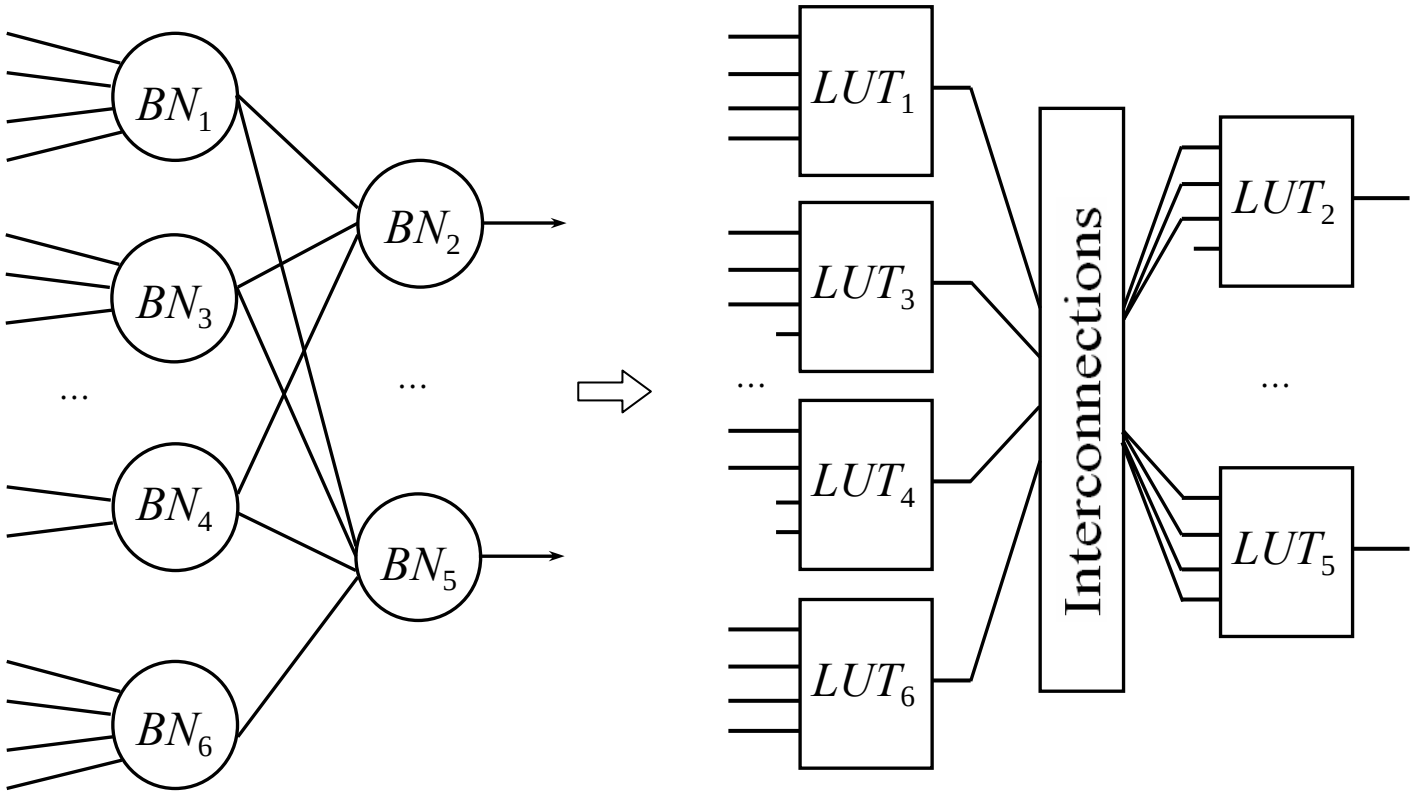




Kohut, R., Steinbach, B., Fröhlich, D. 7th International Workshop on Boolean Problems, Freiberg, Sept. 21-22, 2006

Boolean Neural Networks

- Mapping of BNN to FPGA



Kohut, R., Steinbach, B., Fröhlich, D. 7th International Workshop on Boolean Problems, Freiberg, Sept. 21-22, 2006

UML Models

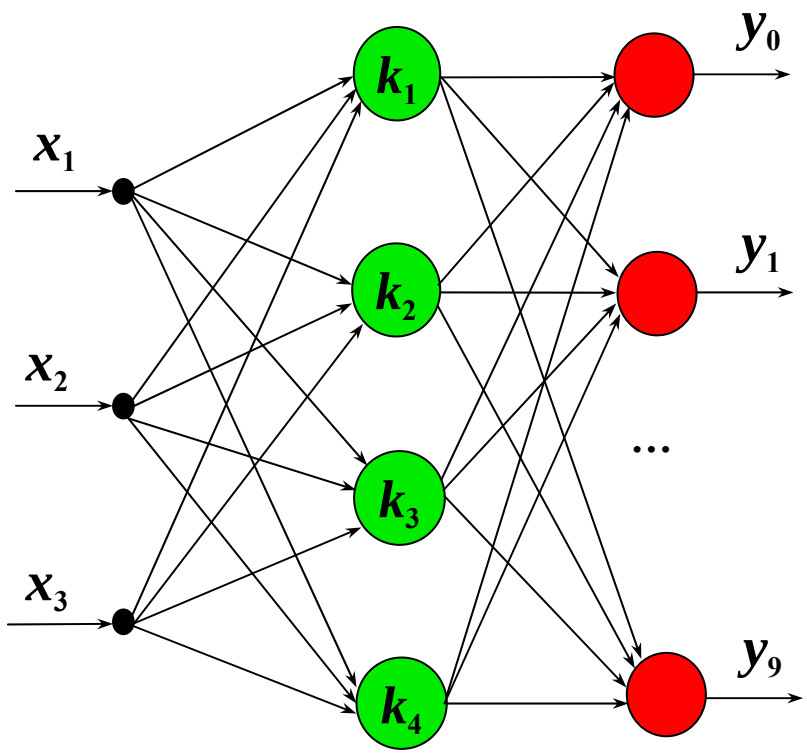
- Example

x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

k_1	k_2	k_3	k_4
0	0	1	0
1	0	1	0
0	0	0	1
1	0	0	0
0	1	0	0
1	0	0	1
1	0	0	0
0	0	0	1

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
k_1	1	1	0	1	0	1	1	0	0	0
k_2	0	1	0	0	1	0	1	1	1	0
k_3	0	0	1	1	0	0	1	1	1	0
k_4	0	0	1	0	1	1	0	0	1	1

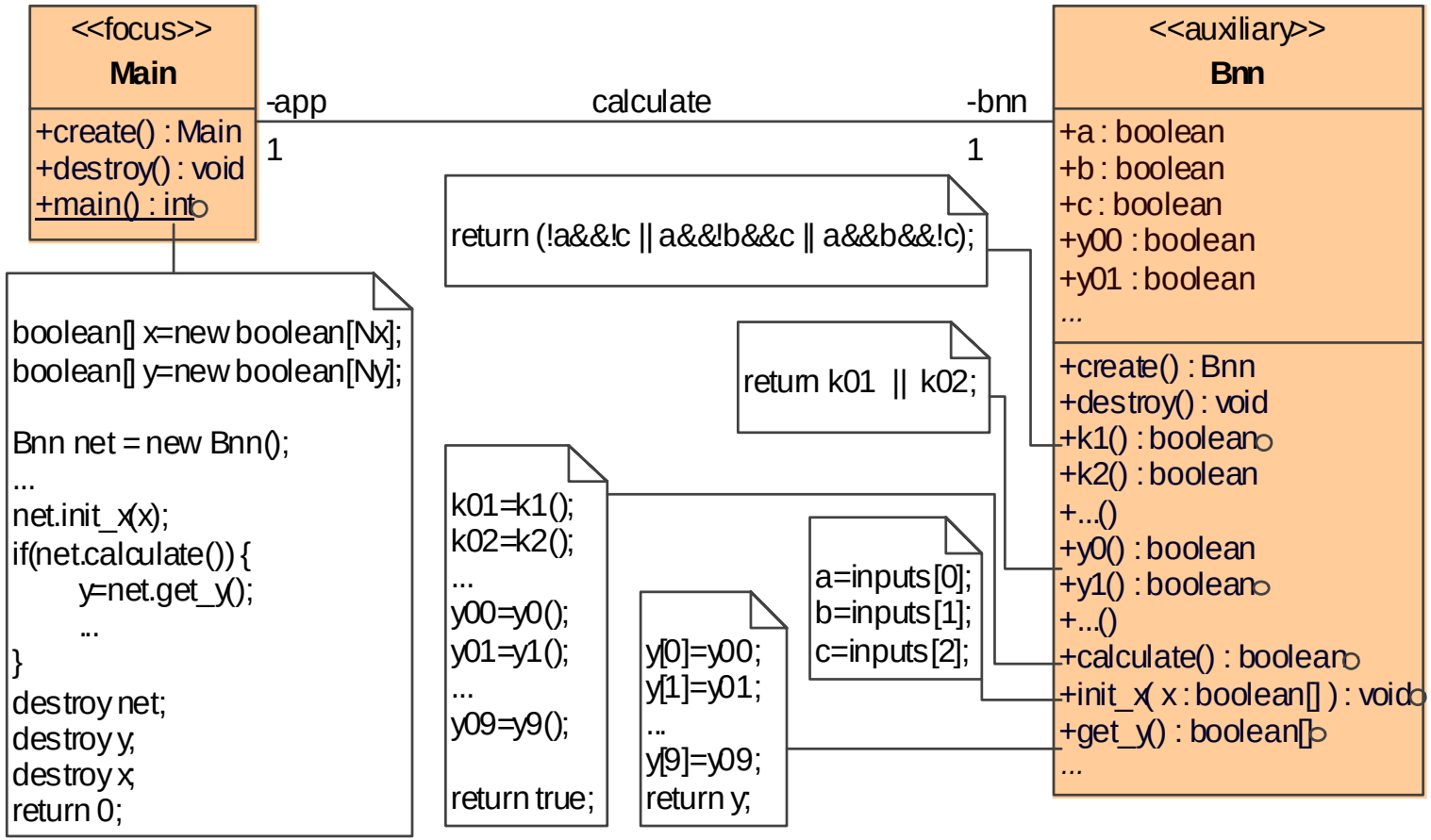
- Structure of BNN



Kohut, R., Steinbach, B., Fröhlich, D. 7th International Workshop on Boolean Problems, Freiberg, Sept. 21-22, 2006

UML Models

- Design Model

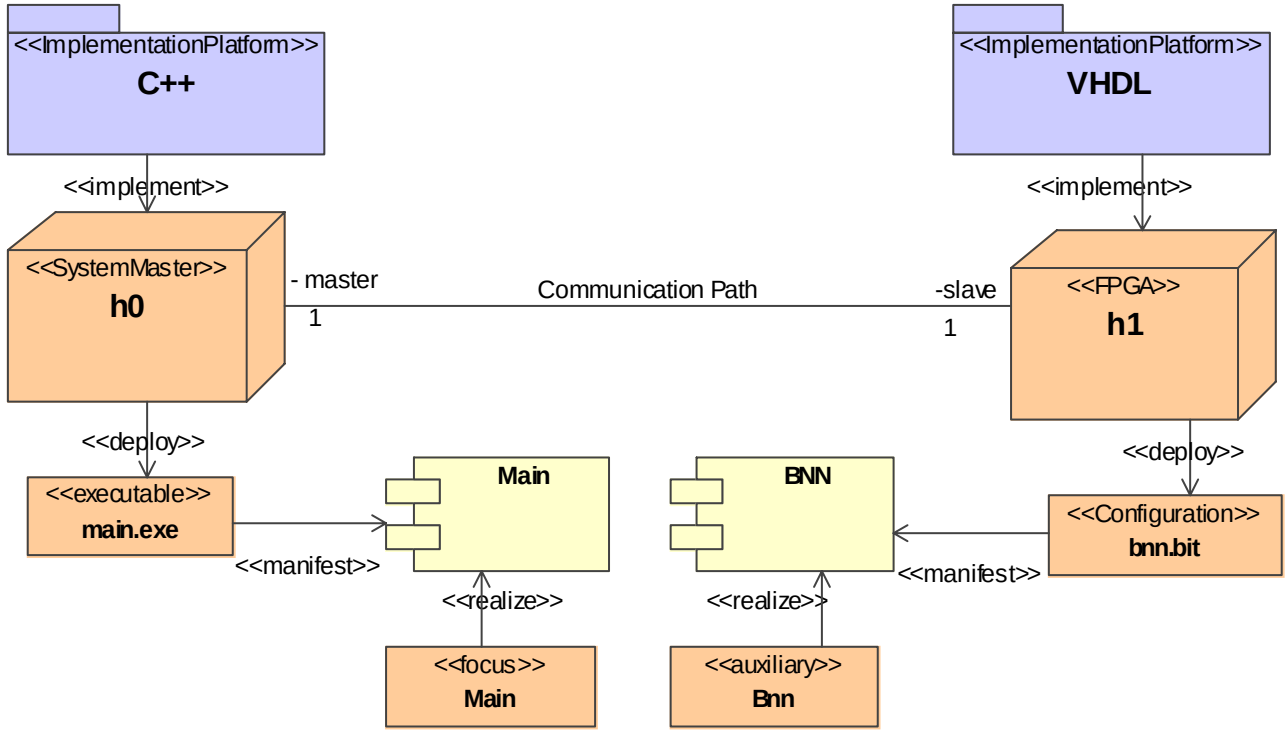


Kohut, R., Steinbach, B., Fröhlich, D. 7th International Workshop on Boolean Problems, Freiberg, Sept. 21-22, 2006

UML Models

- Deployment Model

- Implementations platforms:
  - C++
  - VHDL
- Hardware platform:
  - Pentium IV processor
    - 2.4 GHz
  - Xilinx Virtex-II FPGA
    - 3 million gates
    - 100 MHz
- Communication Path:
  - PCI-Bus
    - 33 MHz



Experiment results

- Device Utilization Summary

compilation/synthesis time:

- 3 - 5 minutes

Logic Utilization	Used for **Bnn**
# Slices:	64 (49)
# Flip Flops:	92 (79)
# LUTs:	91 (56)
# IOBs:	102

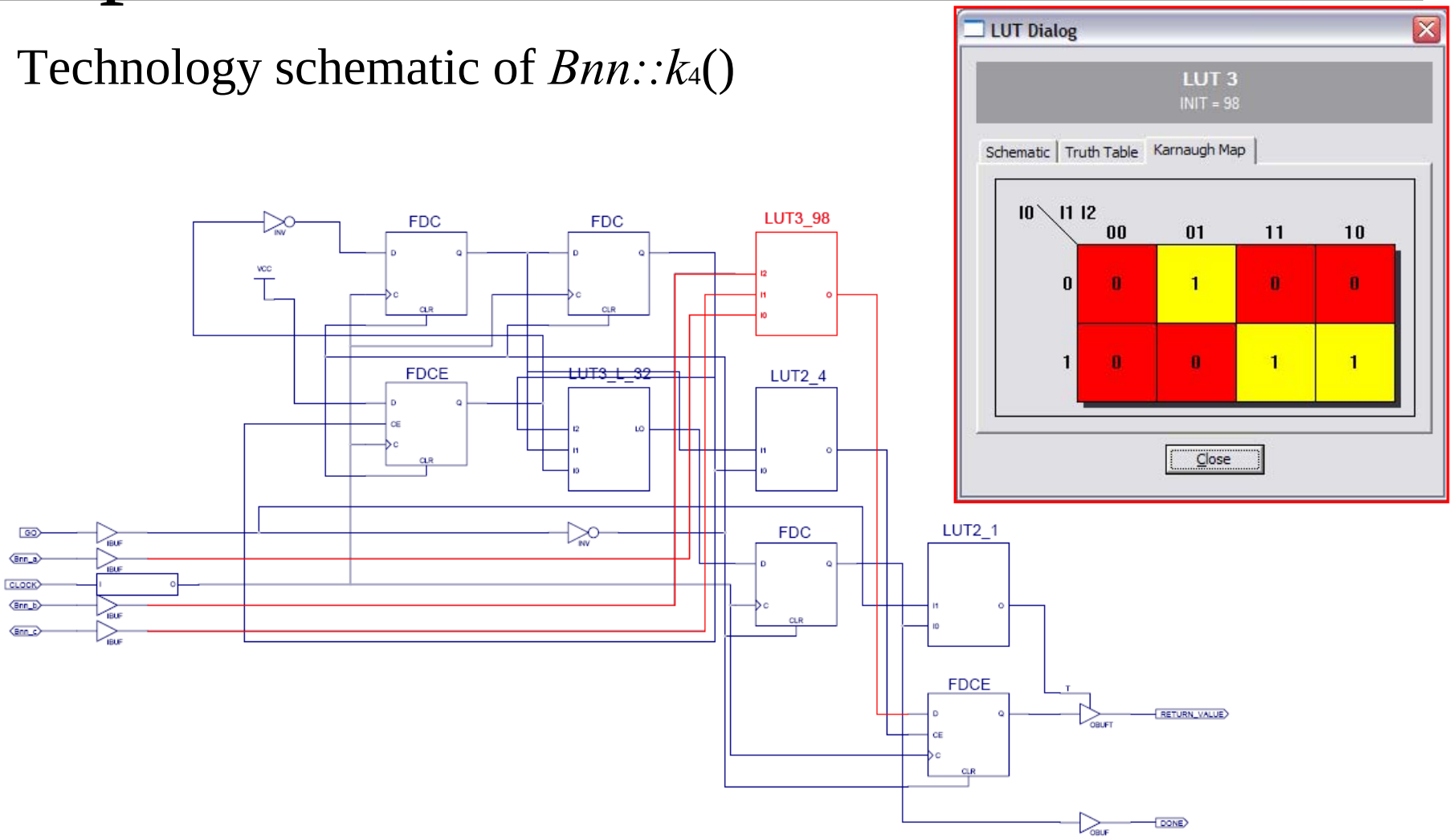
Bnn::calculate:

- 21 (14) LUTs,
- execution time: 0.200 µs

Method	# Slices	#Flip Flops	#4-input LUTs
Bnn::calculate	18	27	21
Bnn::create	1	1	0
Bnn::destroy	1	1	0
Bnn::k1	4	5	7
Bnn::k2	3	4	5
Bnn::k3	2	4	3
Bnn::k4	3	5	4
Bnn::y0	2	3	2
Bnn::y1	2	3	3
Bnn::y2	2	3	3
Bnn::y3	2	3	3
Bnn::y4	2	3	3
Bnn::y5	2	3	3
Bnn::y6	2	4	3
Bnn::y7	2	3	3
Bnn::y8	2	4	3
Bnn::y9	2	3	2

Experiment results

- Technology schematic of *Bnn::k*$_4$()



Conclusion

- Results

(1) UML based hardware/software co-design of Boolean neural networks,

(2) decreasing of the required number of configurable logic blocks (CLB) for the realizing of Boolean neuron,

(3) Boolean neuron can be mapped directly to lookup table (LUT) and configurable logic block (CLB) of FPGAs,

(4) efficient FPGA implementations of BNNs in terms of performance and gate count.

Conclusion

- Future work

  - optimal presentation of Boolean functions by BNNs,
  - automated hardware/software synthesis with MOCCA and UML,
  - optimization of FPGA implementation of Boolean neural networks,
  - design and develop of mapping methodology for Boolean neural networks with on-chip learning.