Multi-FPGA board design using CyberWorkBench, a high-level synthesis tool

Hiroaki Suzuki[†] , Wataru Takahashi[‡] ,
Kazutoshi Wakabayashi[§] , Hideharu Amano[†]

[†]Keio Univ.
[‡]NEC Corporation.
[§]The University of Tokyo. JAPAN

Introduction

Multi-FPGAs have recently attracted much attention for MEC (Multi-access Edge Computing) servers.

Programming Environment for Multi-FPGA Systems…

(1) HLS modules designed by users can be simulated under considering the parallel execution of boards before execution on the real machine.

(2) interface modules for connecting each board are inserted automatically.

(3) modules are automatically divided into the board considering the flow of pipelining.

Simulation of Multi-FPGA Boards

Previous simulation methods
→ The simulation was performed by setting output values for each module mounted on each board individually.

Problems

- Data transfer timing
- I/O delays
- Whether it works properly as a parallel program

Due to these problems
It is necessary to actually run the software on the actual device and debugging.

Building a Multi-FPGA Board Design Environment

We aim to build a multi-FPGA board design environment for multi-FPGA system FiC (Flow-in-Cloud).

Final Goals

- Automatic interface generation
- Automatic partitioning of modules with respect to board assignment

As the first step

In this presentation, we use CWB (CyberWorkBench), a high-level synthesis tool, to LeNet implementation and simulation for multi-FPGA boards and then running it on a FiC multi-board.

Related Research

- A CWB-Based Method for Finding Split Points of Large-Scale Algorithms for Multiple FPGAs[1]

  → Increasing the amount of resources used versus increasing the speed by parallelizing the program is reported.

- How to use MPI for programming in a heterogeneous environment with FPGAs connected to Xeon[2]

  → The parallelization of processing between FPGAs connected to the host is different from FiC, which is a direct connection system between FPGAs.

[1] K. Daiki, T. Miyajima, and H. Amano, "A circuit division method for high-level synthe-sis on multi-fpga systems," 2013 27th International Conference on Advanced InformationNetworking and Applications Workshops, pp.156-161, 2013.

[2] P. Chow, M. Saldana, A. Patel, and C. Madill, "Programming the nallatech xeon ˜+ multi-fpga heterogeneous platform," 2009 IEEE Hot Chips 21 Symposium (HCS), pp.1-16, 2009.

The proposed design flow

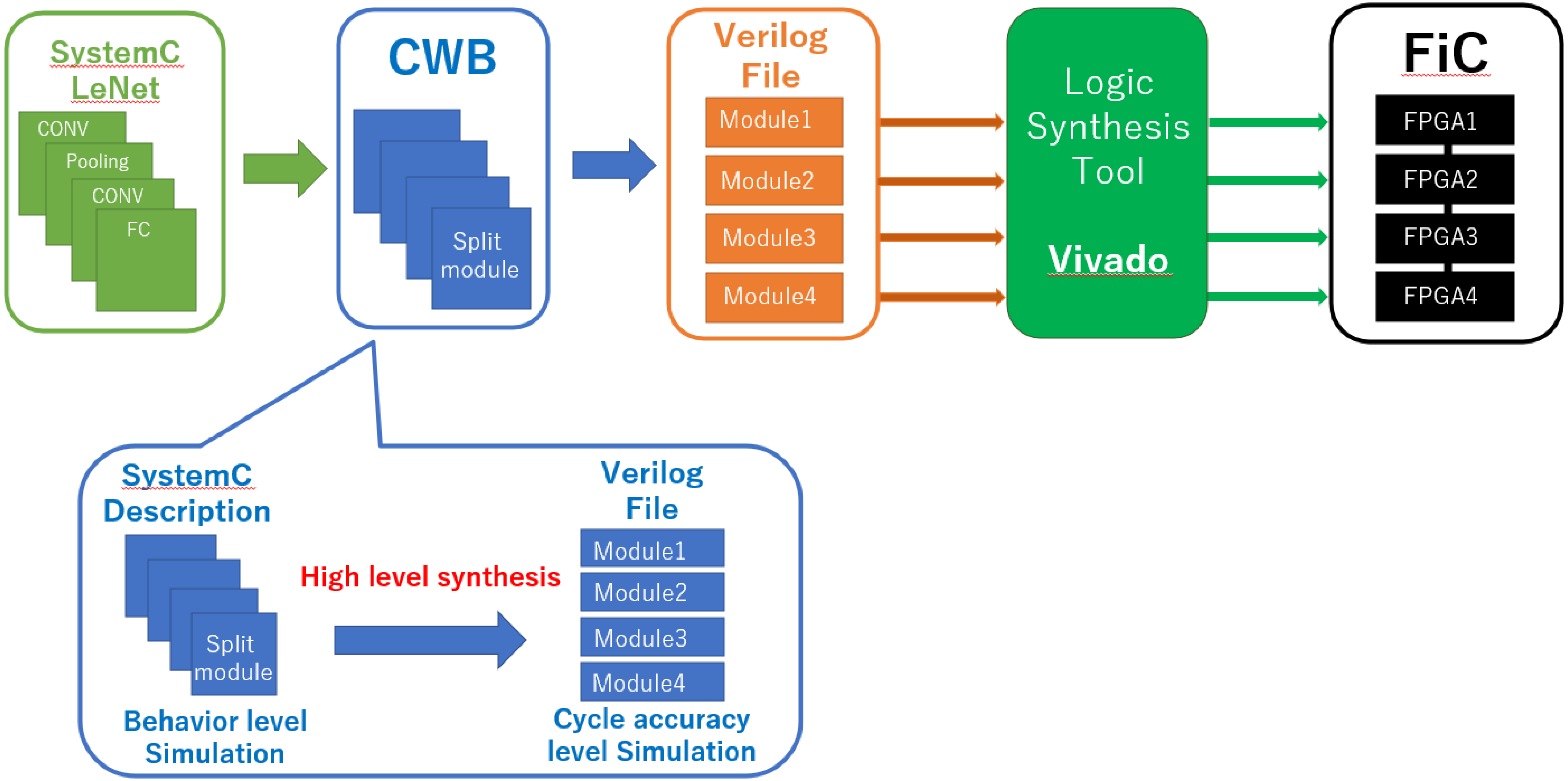


Figure 1: The design flow for a multi-FPGA system

FiC(Flow-in-Cloud)

The FPGA boards of FiC called FiC-SW are connected to each other with high-speed serial links.

Since FPGA boards are directly interconnected, we do not have to increase the number of host servers for expanding computing resource.

The direct connected multi-FPGA systems are more scalable than other FPGAs in the cloud computing environment.

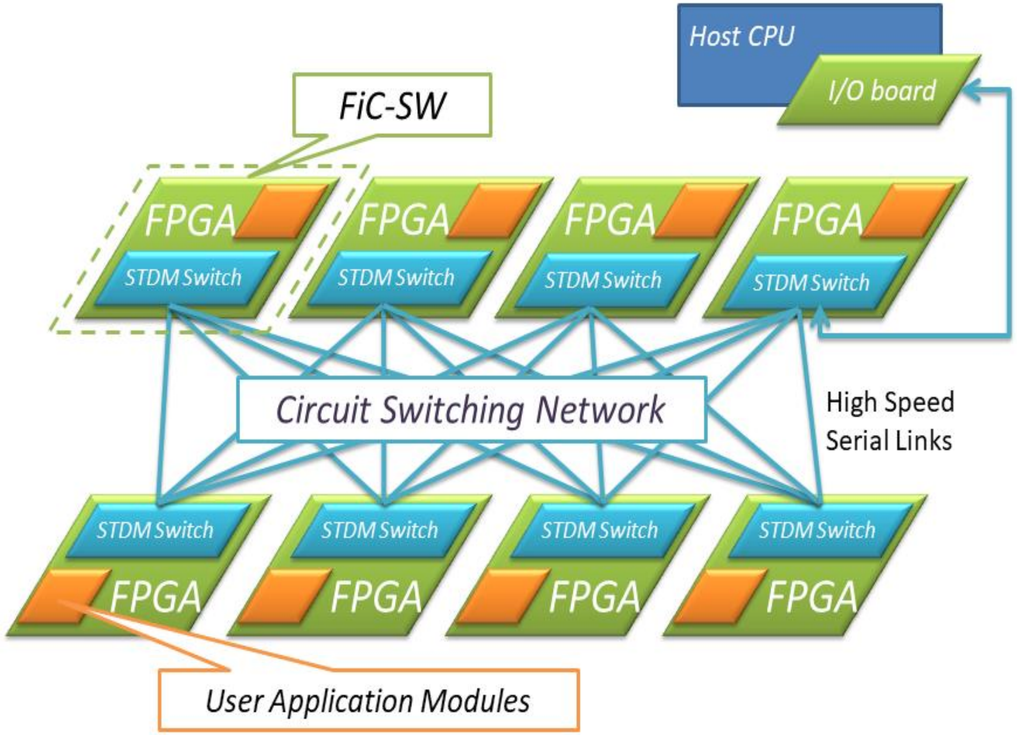


Figure 2: Flow-in-Cloud

FiC SW Board

FiC Board Configuration

- Kintex Ultrascale XCKU095/115
- 32 GTH serial interfaces
- DDR4 SDRAM 16GB×2
- Raspberry Pi3

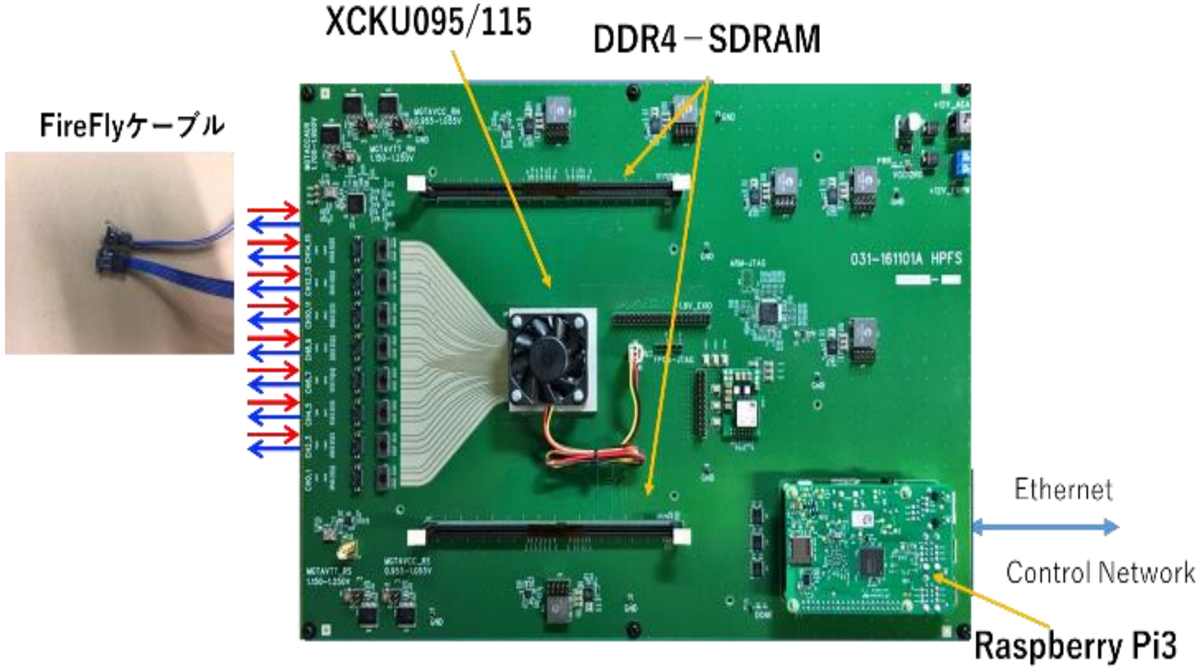


Figure 3: FiC SW Board

Core part of the program

Parallelism patterns used in FiC.

a) Stream processing, in which data flows in one direction.

b) Data is sent from the host to be processed in parallel, and then the results are collected.

c) a pattern that performs iterative computation while broadcasting.

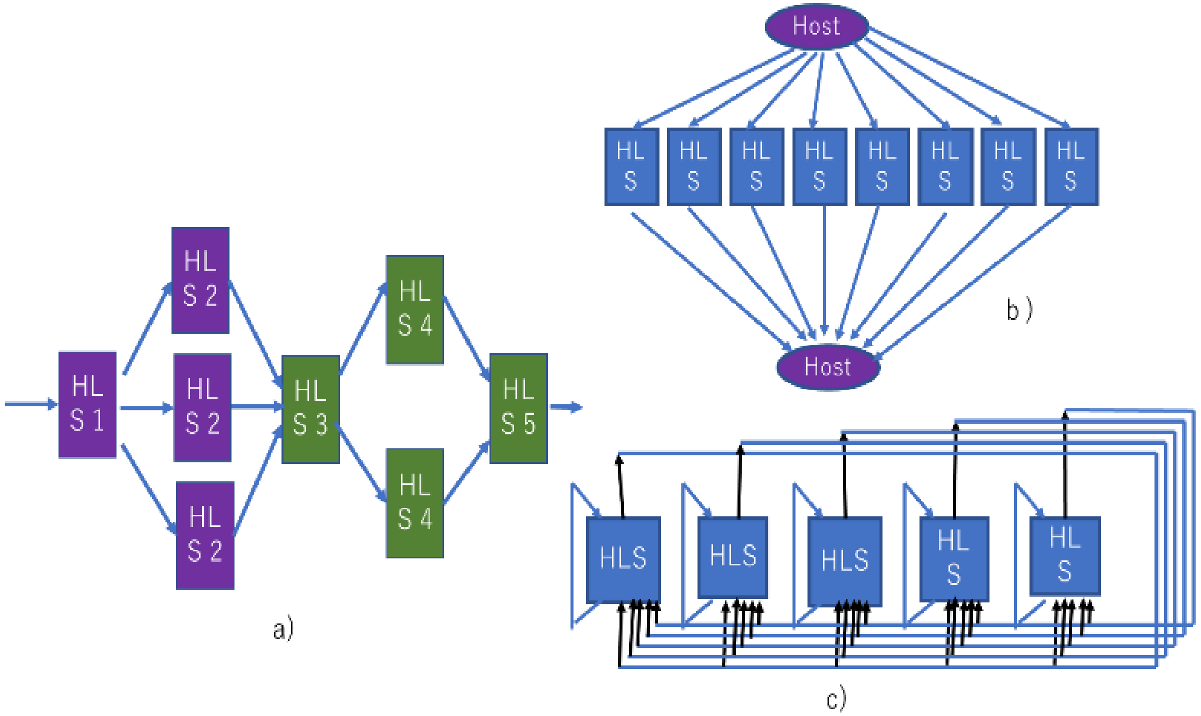


Figure 4: Parallel Processing Patterns in FiC

FPGA Design Tools

Xilinx's Vivado-HLS is currently the most widely used.

- It has a specialized simulation environment for designing within a single FPGA.

  → Vivado-HLS also supports SystemC, but we cannot expect the vendor to automatically insert interfaces or automatically partition modules, which is the ultimate goal.

Intel's OpenCL and Xilinx's SDAccel are also single-chip design environments, with no scalability to multiple FPGAs.

NEC's CyberWorkBench

A base for building parallel programming environments for multi-FPGA systems.

CWB(CyberWorkBench)

CWB

→ C language-based high-level synthesis tool for ASIC and FPGA design developed by NEC Corporation.

CWB consists of a variety of tools, not just a motion synthesis tool

→ Not only synthesis, but also verification such as debugging can be performed.　This automates most of the work processes required for LSI design.

Reduce the amount of description to about 1/7 and speed up simulation by several hundred times.[3]

[3] http://www.nec.co.jp/press/ja/1108/2501.html, 2011

SystemC

SystemC is a hardware modeling language based on C++, and a class library implemented in C++.

SystemC builds hardware systems and runs them together with software such as C and C++.

→ It makes it possible to perform hardware and software co-design and verification at high speed.

Conventional RTL description

Interfaces and Internal structure with cycle accuracy Detailed structure is required.

SystemC

Because the design can be done in TLM with high level of abstraction Reduction in the amount of description code, and faster simulation compared to RTL.

LeNet

As the first step in building a simulation environment for FiC,
we implemented LeNet [4] as an example application implementation.

Structure of LeNet

・CONV layer

・pooling layer

・FC layer

Separately trained for CONV and FC layers
weighted data.

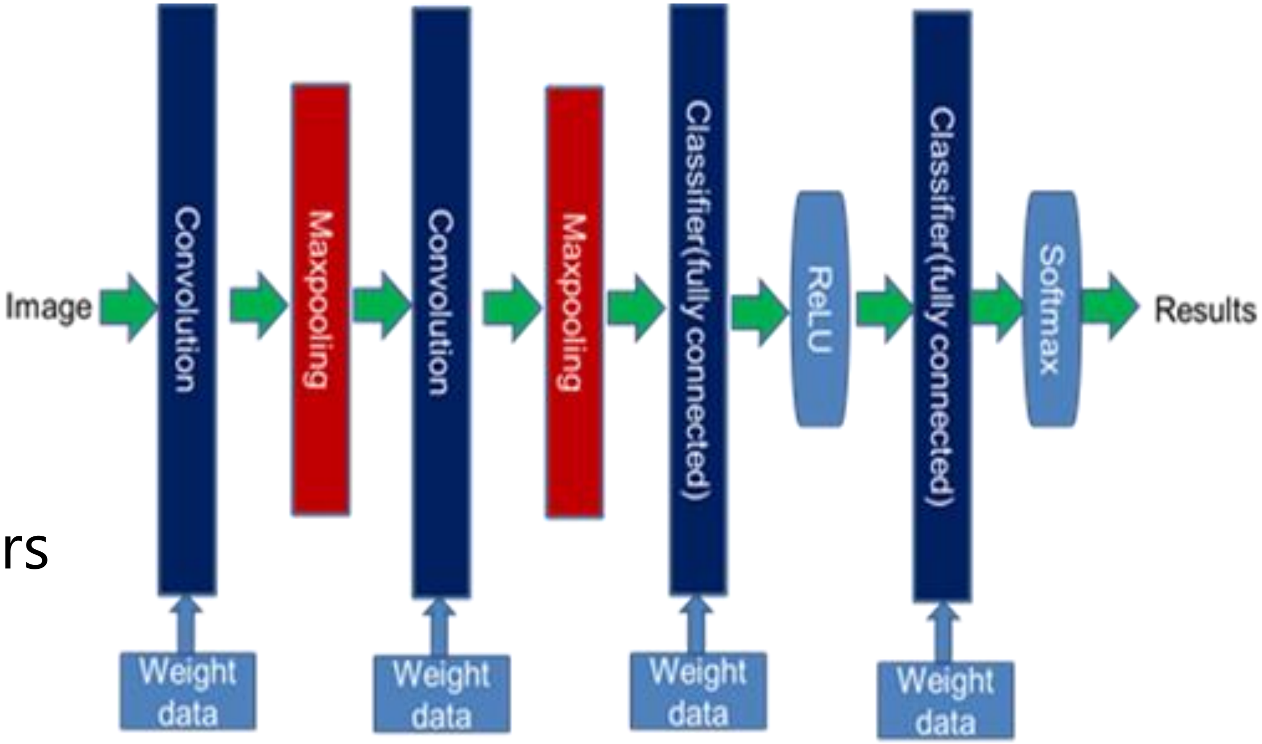


Figure 5: LeNet layer structure

[4]Y.LeCun, P.Haffner, L.Buttou, Y.Benio,
http://yann.lecun.com/exdb/publis/pdf/lecun-99.pdf

Implementation of LeNet by SystemC on CWB

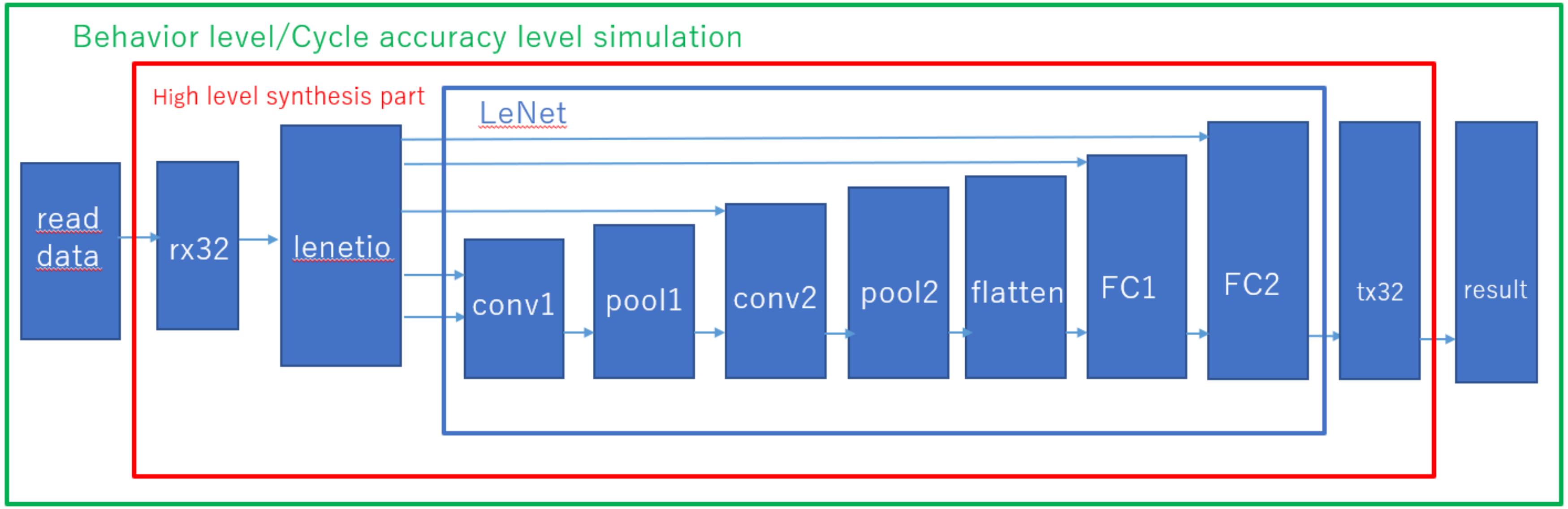

Behavior level/Cycle accuracy level simulation
High level synthesis part
LeNet
read data
rx32
lenetio
conv1
pool1
conv2
pool2
flatten
FC1
FC2
tx32
result

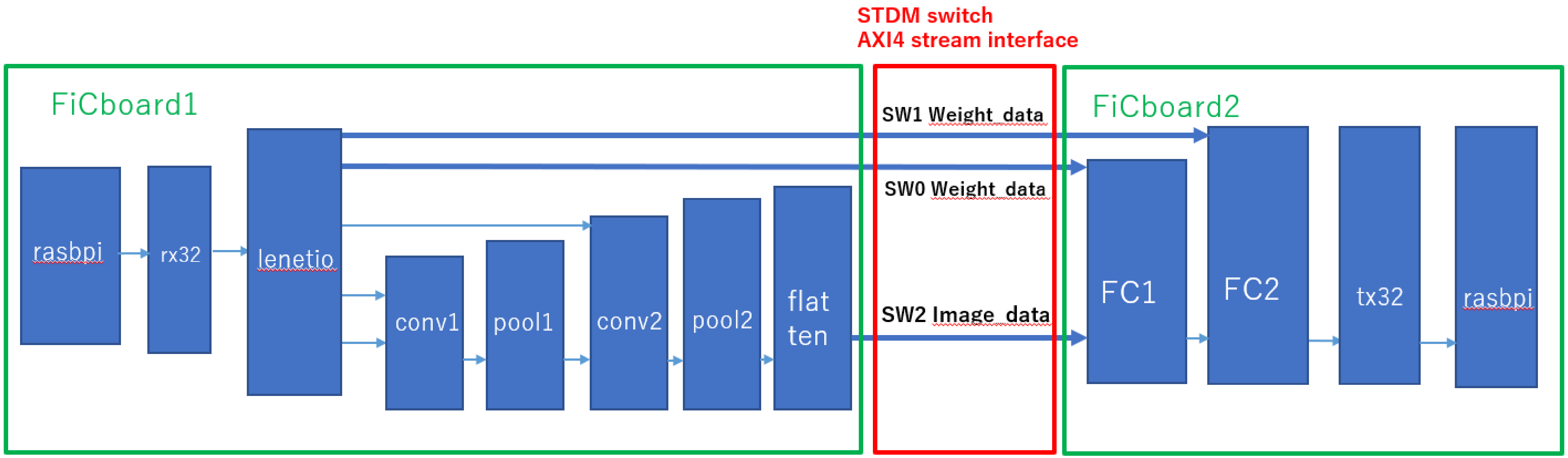
Figure 6: LeNet configuration diagram by SystemC assuming operation on FiC

Modularize each layer of LeNet

Each module can have a process, and The process is triggered by a clock or signal change, and The processes are executed in parallel.

All modules are connected by FIFO.

Configuration diagram for multi-board operation



- Figure 7: Configuration diagram for mounting on a multi-board

FiC board 1 and FiC board 2 are connected by AXI4 stream interface, and currently four switches (SW0~SW3) are available for FiC.

Send weight data to SW0 and SW1, and image data to SW2.

Evaluation

Evaluation Environment

・High level synthesis ：CyberWorkBench 8.3
Vivado-HLS 2019.1.3

・ Logic synthesis
Place and Route ：Vivado 2019.1.3

・CPU ：AMD Ryzen Threadripper 3960X

・device ：Kintex Ultrascale XCKU095

Evaluation

Table 1: Execution time with conventional tool and CWB (seconds)

	Vivado HLS	CWB multiboard
Execution time	0.2492	0.2354

Table 2: Execution time for each simulation (seconds)

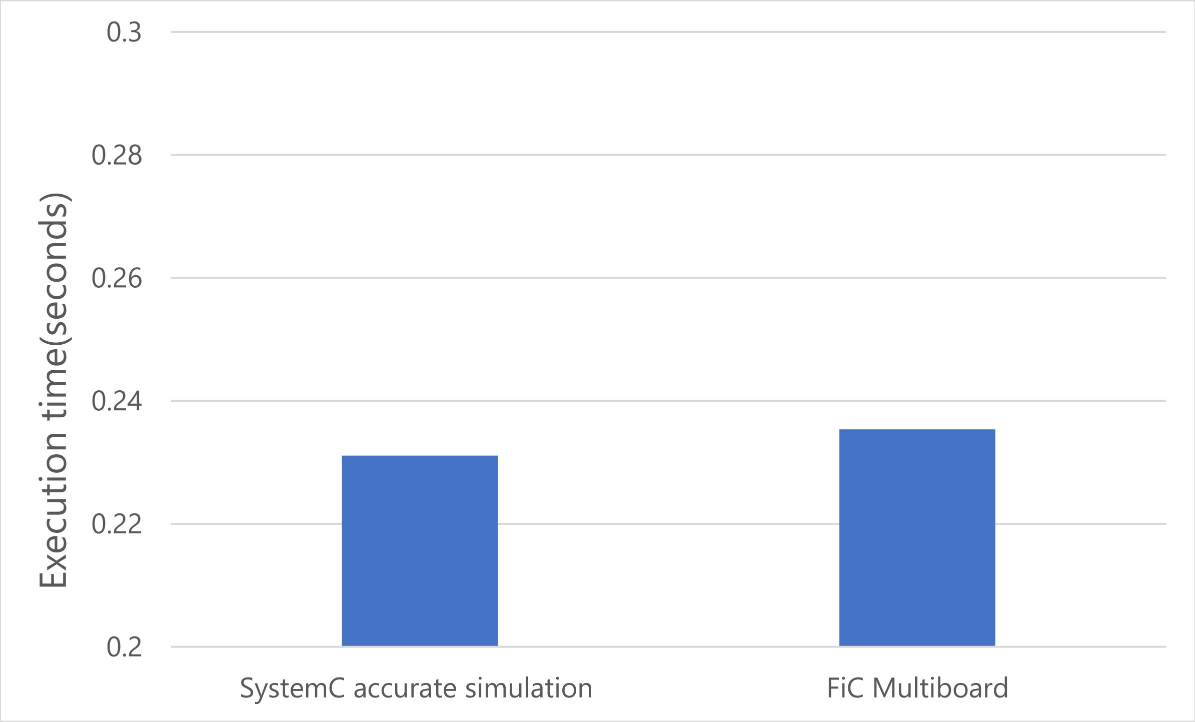
C++	SystemC Behavioral	SystemC Cycle accurate Generated by CWB
0.013	2.9	105.34

Evaluation

Table 3: Execution time of each LeNet module

Module	Execution time[ns]
rx32	10
lenetio	10
conv1	28869700
pool1	691240
conv2	160058250
pool2	192030
flatten	32010
fc1	40021080
fc2	512460
tx32	10
total	230376800

Table 4: Comparison of estimated time and execution time



Conclution

A multi-FPGA programming environment based on NEC's integrated design tool CWB is introduced for a multi-FPGA system FiC.

As an example, the description of a simple CNN LeNet is shown, and implementation on a real system using the tool is evaluated.

The estimated cycles is only 2.2% difference of the real boards execution result.

Our future work is to finish the design flow automatically as possible.