

ROS-COMPLIANT FPGA COMPONENT TECHNOLOGY – INSTALLATION OF FPGA INTO ROS

Takeshi Ohkawa^{*}, Yutaro Ishida^{**},
Yuhei Sugata^{*}, Hakaru Tamukoh^{**}

^{*}Utsunomiya University,

^{**}Kyushu Institute of Technology

This research and development work (done by Utsunomiya Univ.)
was supported by the MIC/SCOPE#152103014.



宇都宮大学
UTSUNOMIYA UNIVERSITY



Kyutech
Kyushu Institute of Technology

Problem of computer vision on robots

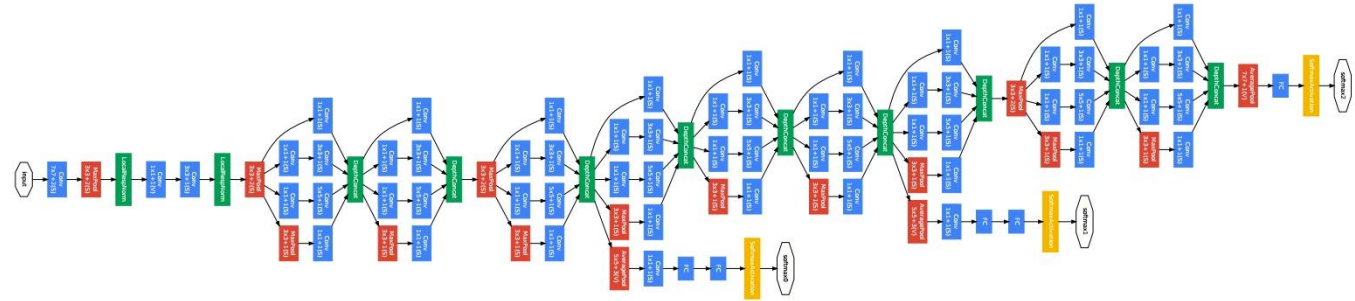
RoboCup 2016
Domestic Pre-Competition



Over 1 min. to recognize just six objects

FPGA accelerates DNN

- “1 min. is too long for the task!”
- GoogLeNet[3] (22 layers)
- **4 seconds** for an inference
 - CPU: Core i5-5200U 2.2GHz



arXiv:1409.4842v1 [cs.CV] 17 Sep 2014 「Going deeper with convolutions」

- Need of accelerator
 - GPU – Power consumption, Heat!
 - **FPGA** – optimized circuit for the app = **high performance/power efficiency**

[3] “Model Zoo”, http://dl.caffe.berkeleyvision.org/bvlc_googlenet.caffemodel

Evaluation of CPU, GPU, FPGA*

- Model of Deep Neural Networks: VGG16
- Dataset: Cifar 10

Device	NVIDIA Jetson TX1		Xilinx ZCU102 (FPGA)
	Quad-core ARM Cortex-A57	256-core Maxwell GPU	Zynq UltraScale+ MPSoC
Clock Freq.	1.9 GHz	998MHz	100MHz
Efficiency [FPS/W]	0.032	0.376	1.43
Accuracy [%]	92.35		90.30

3.8 times efficient



* H. Yonekawa and H. Nakahara, "On-Chip Memory Based Binarized Convolutional Deep Neural Network Applying Batch Normalization Free Technique on an FPGA," Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017 IEEE International, 2017.

Recent trends on FPGA

- Application
 - Image Recognition (local feature, DNN)
 - Security (encryption)
 - Big Data (compression, mining)
- Device
 - Single Chip - ARM + FPGA
 - Mar. 2011, Xilinx Zynq
 - Oct. 2011, ALTERA SoC FPGA
 - 2016- (announced) Intel Broadwell Xeon + FPGA (ALTERA Arria 10)
- Tool
 - High-Level Synthesis (C to HDL)
- Service
 - Apr. 2017, Amazon EC2 F1 instance
 - FPGA: Xilinx Virtex UltraScale+ VU9P (2,586,150 Logic Cells)



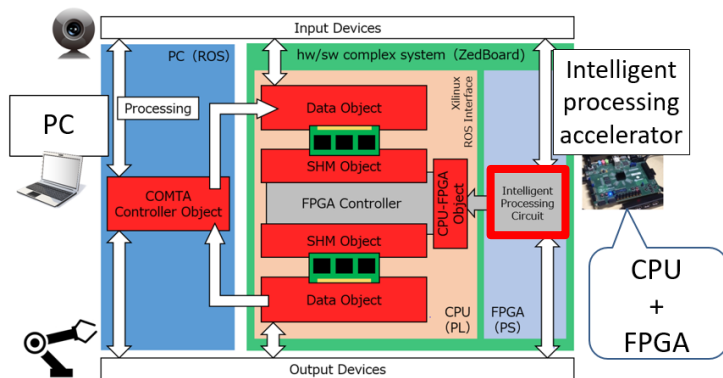
Intel's \$16.7B Altera Buyout
(June 2, 2015)

Problem for introducing FPGA

- “Difficult design of FPGA”
- Why FPGA is difficult to develop?
 - Language: **HDL** (Hardware Description Language) or **HLS** (High-level Synthesis)
 - Need of performance tuning techniques (processing and memory access: **FPGA expertise**)
 - Long compile time
- **Component technology is necessary!**
 - In order to use/reuse **high-performance FPGA design** easily from software.
 - Let's make ROS-node as FPGA-component!

Our proposal: FPGA component technologies for ROS system (3 types)

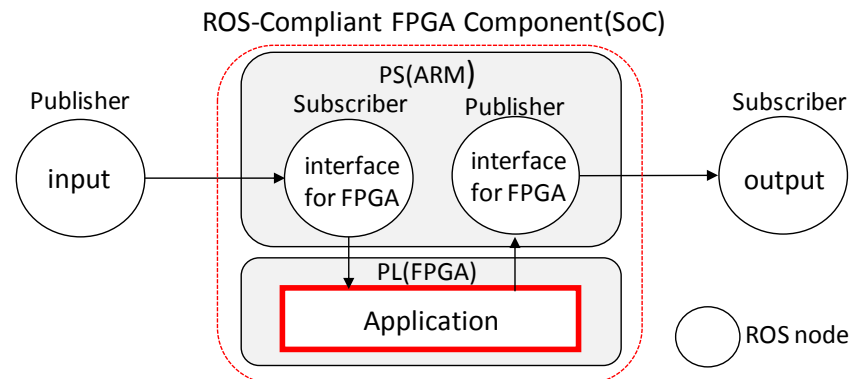
- (1) COMTA (PC and Programmable SoC*)
 - PC: ROS on Linux
 - ARM: bridge (TCP/IP)
 - **FPGA: Application logic**



Proposed by Kyutech
Mar. 2015



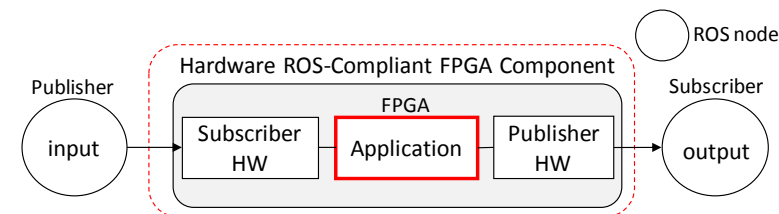
- (2) ROS-compliant FPGA component on Programmable SoC*
 - ARM: ROS on Linux
 - **FPGA: Application logic**



Proposed by Utsunomiya Univ.
Mar. 2015



- (3) Fully-hardwired ROS-compliant FPGA component
 - Talks ROS protocol by using TCP/IP stack
 - Direct connection to **FPGA Application logic**

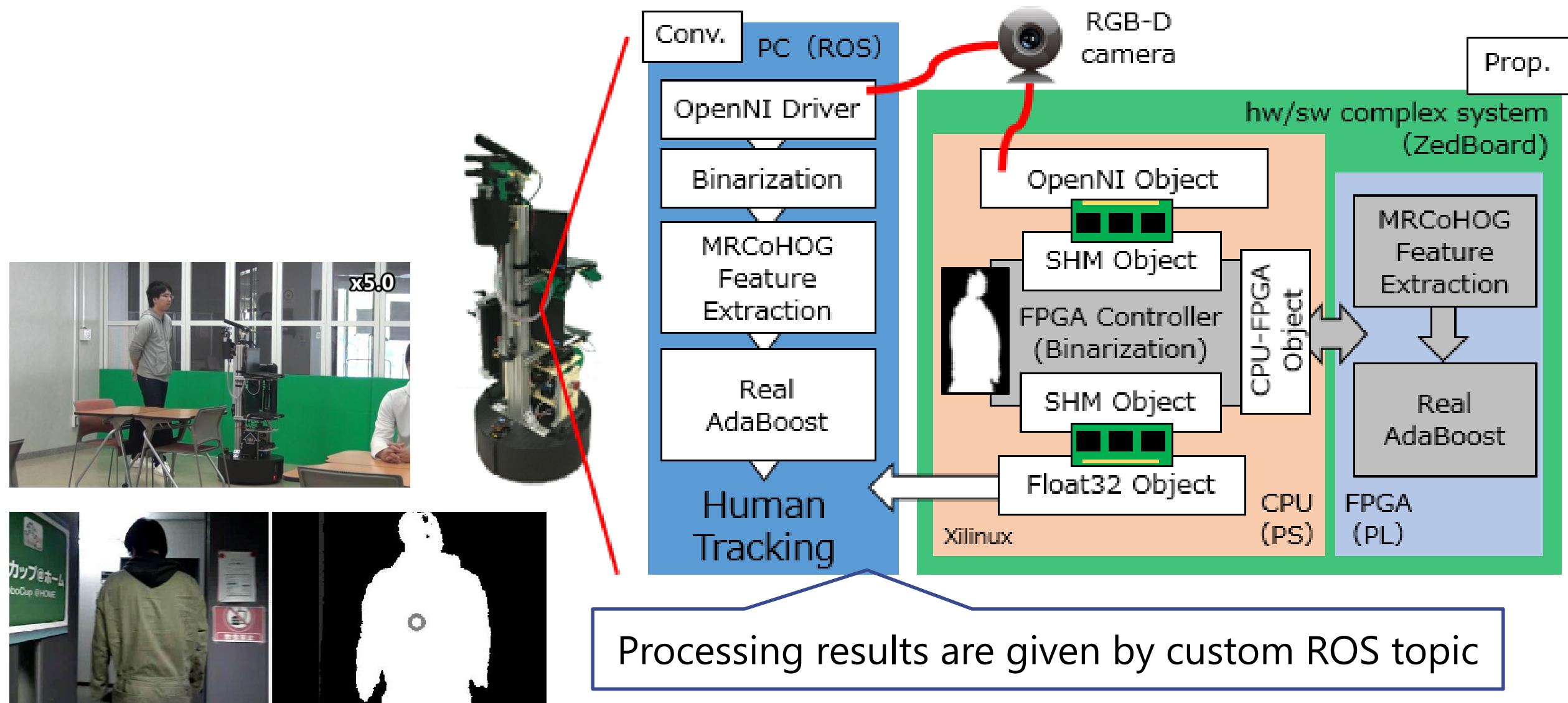


Proposed by Utsunomiya Univ.
Mar. 2017



*Programmable SoC: System-on-chip, ARM+FPGA, i.e. Xilinx Zynq-7000, Intel (ALTERA) SoC FPGA

(1) COMTA Experiment: Following human by using image processing



Processing results are given by custom ROS topic

(1)COMTA Experiment: Results

Device	Conv.	Prop.
	Core i5-5200U 2.2GHz	ARM Dual Core 667MHz + FPGA
Frame Rate [FPS]	29.06	17.25
Power [W]	26	4.7
Efficiency [FPS/W]	1.12	3.69

3.3 times efficient

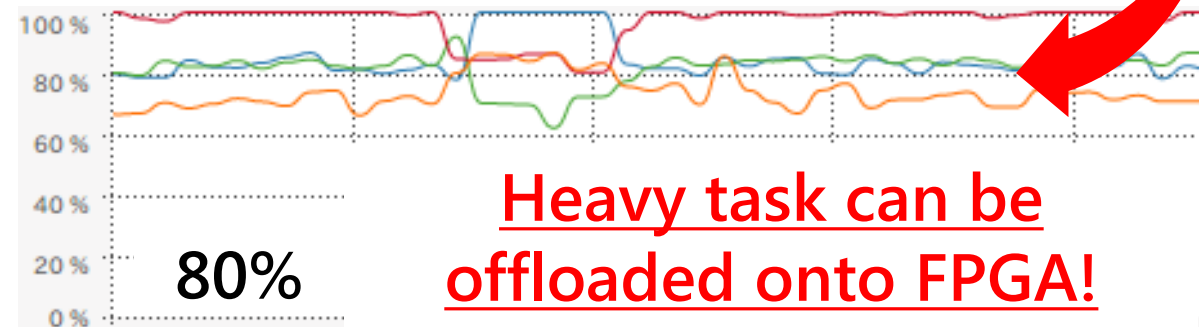
Computing load of Conv. method

Status: Idling



50 points increase

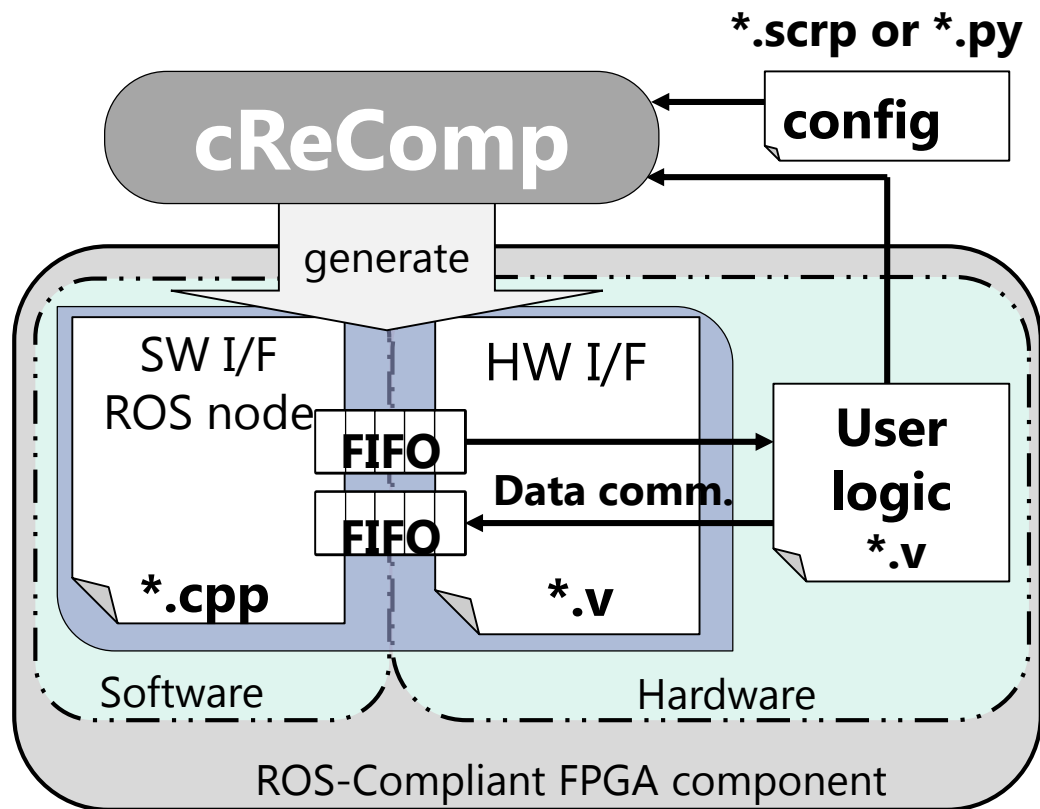
Status: Following human



Heavy task can be offloaded onto FPGA!

(2) ROS-compliant FPGA component on Programmable SoC

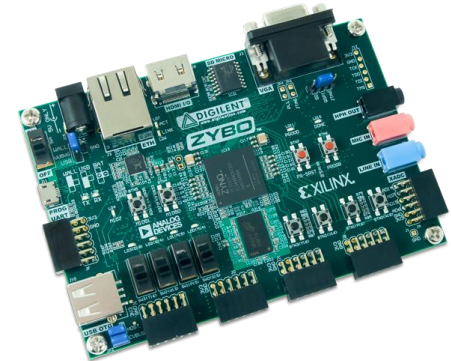
cReComp: automatic generation of ROS component form HDL



- **Target**
- Programmable SoC (Xilinx Zynq)
- **Input**
- Application Logic: HDL
- Configuration File:
 - scrp (Text)
 - Python (using PyVerilog)
- **Output**
- HDL: FIFO buffer control
- C++: ROS node



Zedboard



ZYBO

**Experimental (6 undergraduate/graduates) result:
finished componentization within an hour! (42 minutes maximum)**

Shared at <https://github.com/Kumikomi/cReComp>

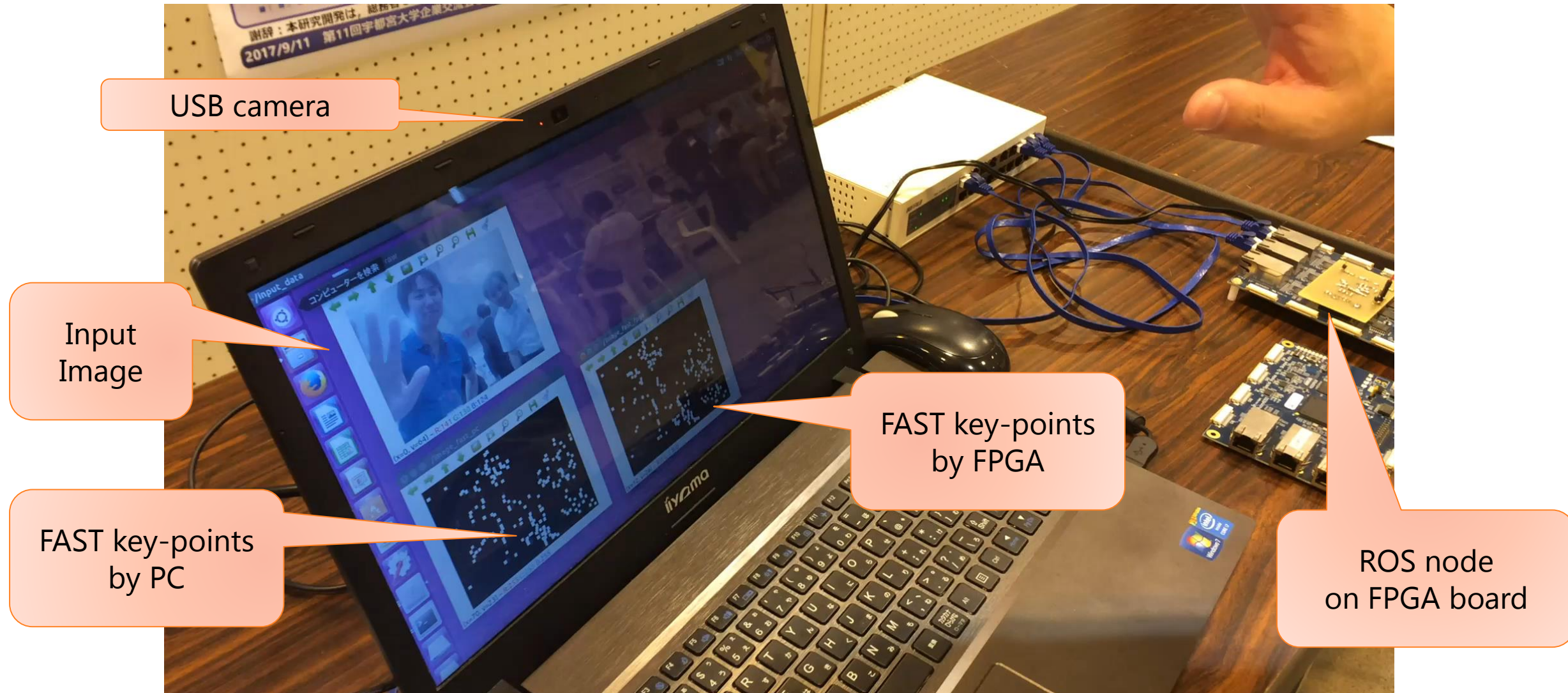
(2) ROS-compliant FPGA component on Programmable SoC Open-Source Packages Available in ROS.org

The screenshot shows the ROS.org website interface for browsing packages in the Jade distribution. The page title is "Browsing packages for jade". A table lists several packages, with "openrerc_motion_sensor" highlighted. The description for this package states: "This package supports a ultra sonic sensor using an **FPGA** board (ZedBoard Xilinx)." Other packages listed include ackermann_msgs, actionlib, openni_launch, and openrerc_pwm.

Shared at
https://github.com/Kumikomi/openrerc_motion_sensor
https://github.com/Kumikomi/openrerc_pwm

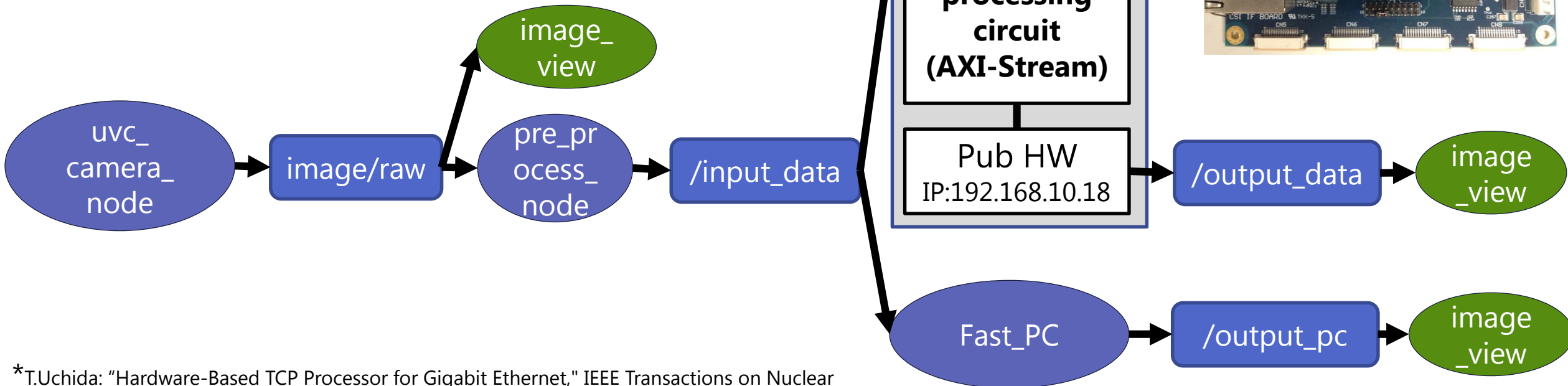
(3) Fully-hardwired ROS-compliant FPGA component

DEMO MOVIE: FPGA-ROS node of FAST key-point detection



(3) Fully-hardwired ROS-compliant FPGA component - structure of the evaluation system

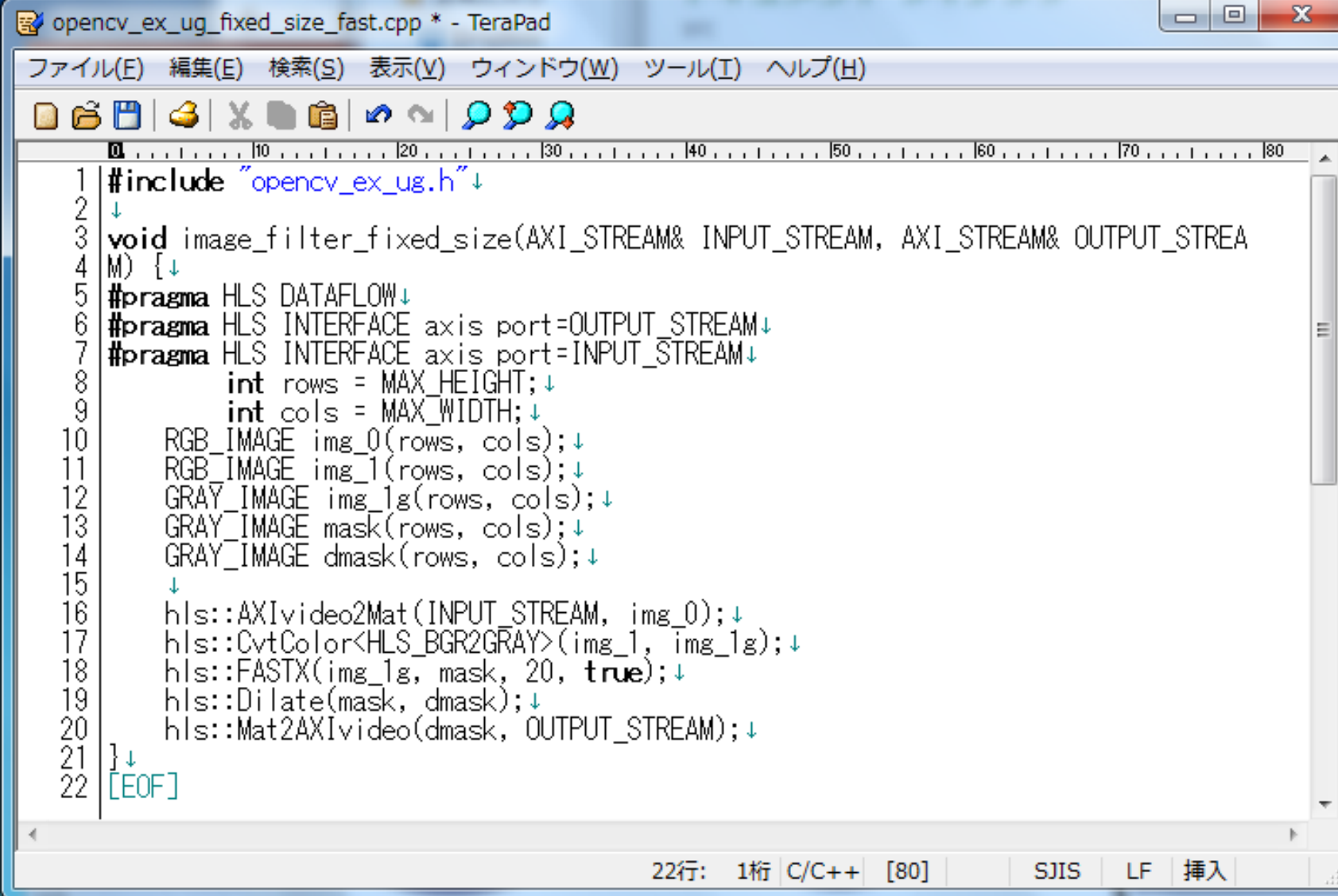
- FPGA is accessed via Gigabit Ethernet
- Image processing circuit
 - HLS (Xilinx VivadoHLS) from C++ (**OpenCV**)
- Subscriber/Publisher HW (HDL)
 - Implemented using **TCP/IP stack** (SiTCP*)



*T.Uchida: "Hardware-Based TCP Processor for Gigabit Ethernet," IEEE Transactions on Nuclear Science, Vol.55, No.SIG 3, pp.1631-1637, 2008.

(3) Fully-hardwired ROS-compliant FPGA component

Source code of Image Processing Circuit – for Xilinx Vivado HLS

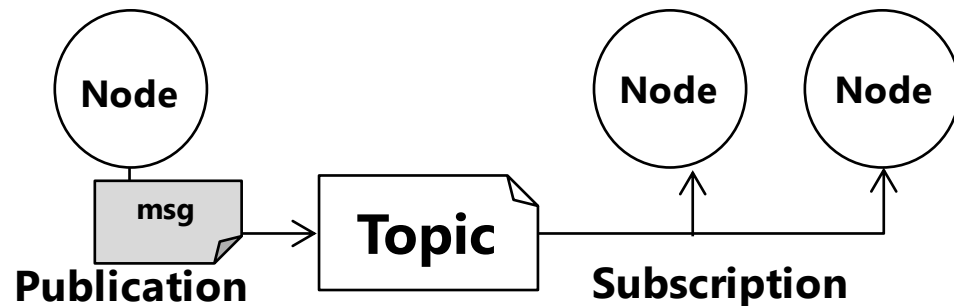


```
opencv_ex_ug_fixed_size_fast.cpp * - TeraPad
ファイル(E) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(I) ヘルプ(H)
1 #include "opencv_ex_ug.h"↓
2 ↓
3 void image_filter_fixed_size(AXI_STREAM& INPUT_STREAM, AXI_STREAM& OUTPUT_STREAM,
4 M) {↓
5 #pragma HLS DATAFLOW↓
6 #pragma HLS INTERFACE axis port=OUTPUT_STREAM↓
7 #pragma HLS INTERFACE axis port=INPUT_STREAM↓
8     int rows = MAX_HEIGHT;↓
9     int cols = MAX_WIDTH;↓
10    RGB_IMAGE img_0(rows, cols);↓
11    RGB_IMAGE img_1(rows, cols);↓
12    GRAY_IMAGE img_1g(rows, cols);↓
13    GRAY_IMAGE mask(rows, cols);↓
14    GRAY_IMAGE dmask(rows, cols);↓
15    ↓
16    hls::AXIvideo2Mat(INPUT_STREAM, img_0);↓
17    hls::CvtColor<HLS_BGR2GRAY>(img_1, img_1g);↓
18    hls::FASTX(img_1g, mask, 20, true);↓
19    hls::Dilate(mask, dmask);↓
20    hls::Mat2AXIvideo(dmask, OUTPUT_STREAM);↓
21 }↓
22 [EOF]
```

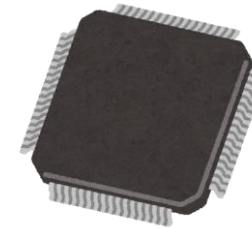
22行: 1桁 C/C++ [80] SJIS LF 挿入

How does the FPGA ROS-node (Hardware Pub/Sub) work?

- ROS wire protocol (TCP/IP packets) i.e. Publish/Subscribe messages
 - Anyone can be a ROS node who talks TCP/IP in ROS manner.
 - It is not the matter if it is Software, Hardware (FPGA) or something!



Software



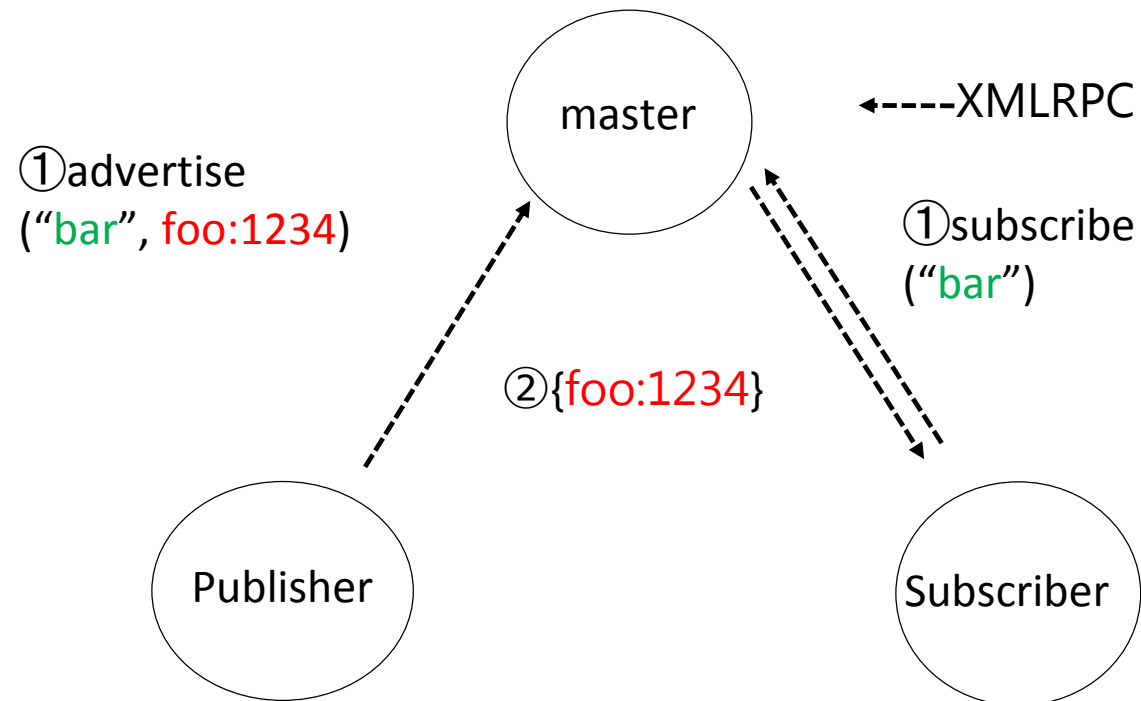
Hardware (FPGA)



something

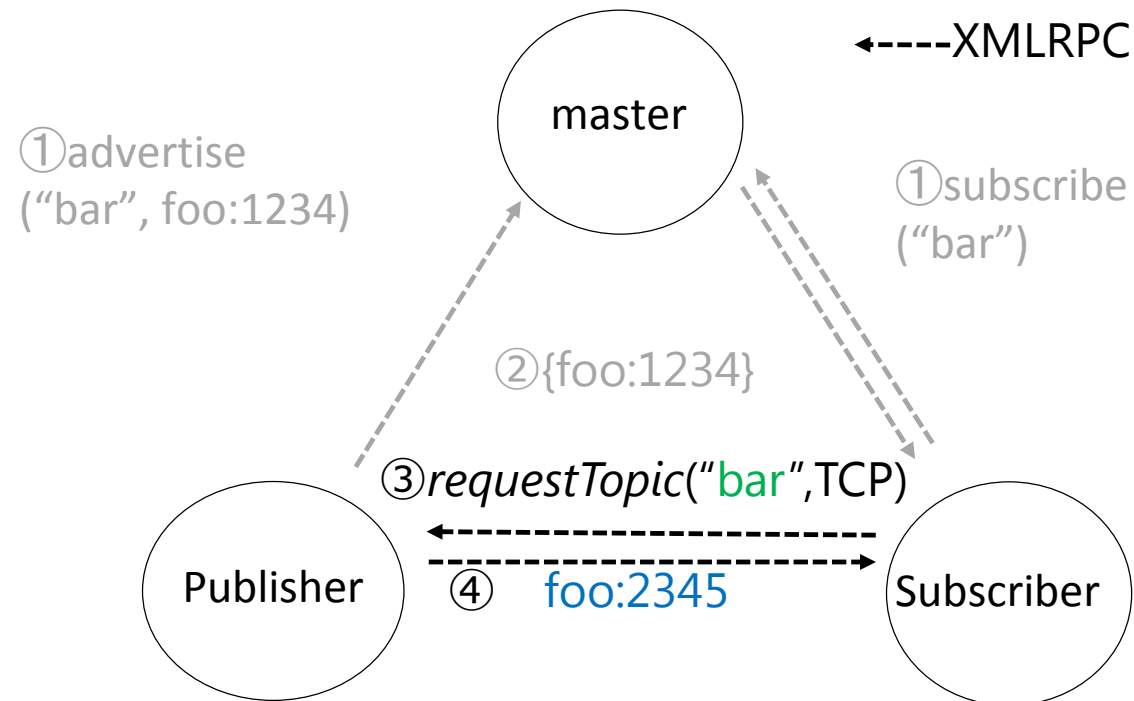
Let's hack the ROS wire protocol.

Procedure of Publish/Subscribe messaging in ROS system (1/3)



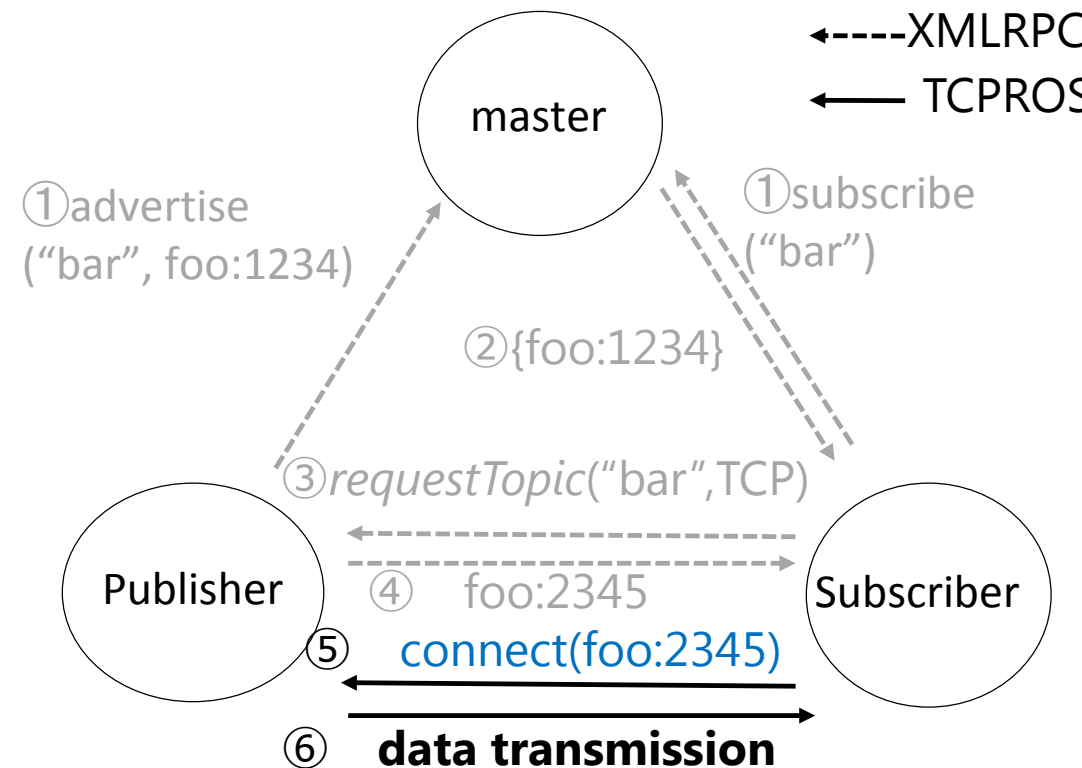
- STEP① and STEP② is communication with master
- master is a name server in ROS system
- **XMLRPC**: remote procedure calling using HTTP as the transport and XML as the encoding

Procedure of Publish/Subscribe messaging in ROS system (2/3)



- STEP③: Subscriber sends a connection request to Publisher by using `requestTopic` (XMLRPC)
- STEP④: Publisher returns host address and port number of TCPROS

Procedure of Publish/Subscribe messaging in ROS system (3/3)

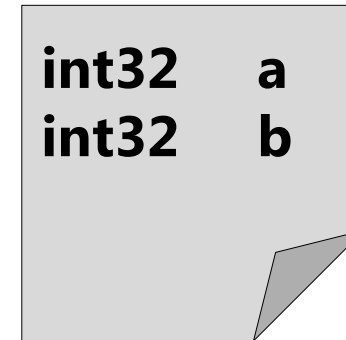


- Binary data transmission starts in **TCPROS protocol**
- STEP⑤: Subscriber establishes a TCP connection
- **STEP⑥: Data transmission repeats**

TCPROS protocol

- Binary data: efficient data transfer
- Easy to handle in hardware
- Example:
 - ROS message = int32 x2

ROS Message



TCP/IP packet (captured by WireShark)

```

52 54 00 60 81 a8 52 54 00 26 a6 ae 08 00 45 00
00 40 fb 14 40 00 40 06 c9 57 c0 a8 7a 73 c0 a8
7a 87 8a 81 df 36 8f de ee 08 61 a8 82 70 80 18
00 eb 76 7e 00 00 01 01 08 0a 00 03 19 ba 00 02
de 55 08 00 00 00 04 00 00 00 05 00 00 00
  
```

Data: message size(int32) +int32+int32

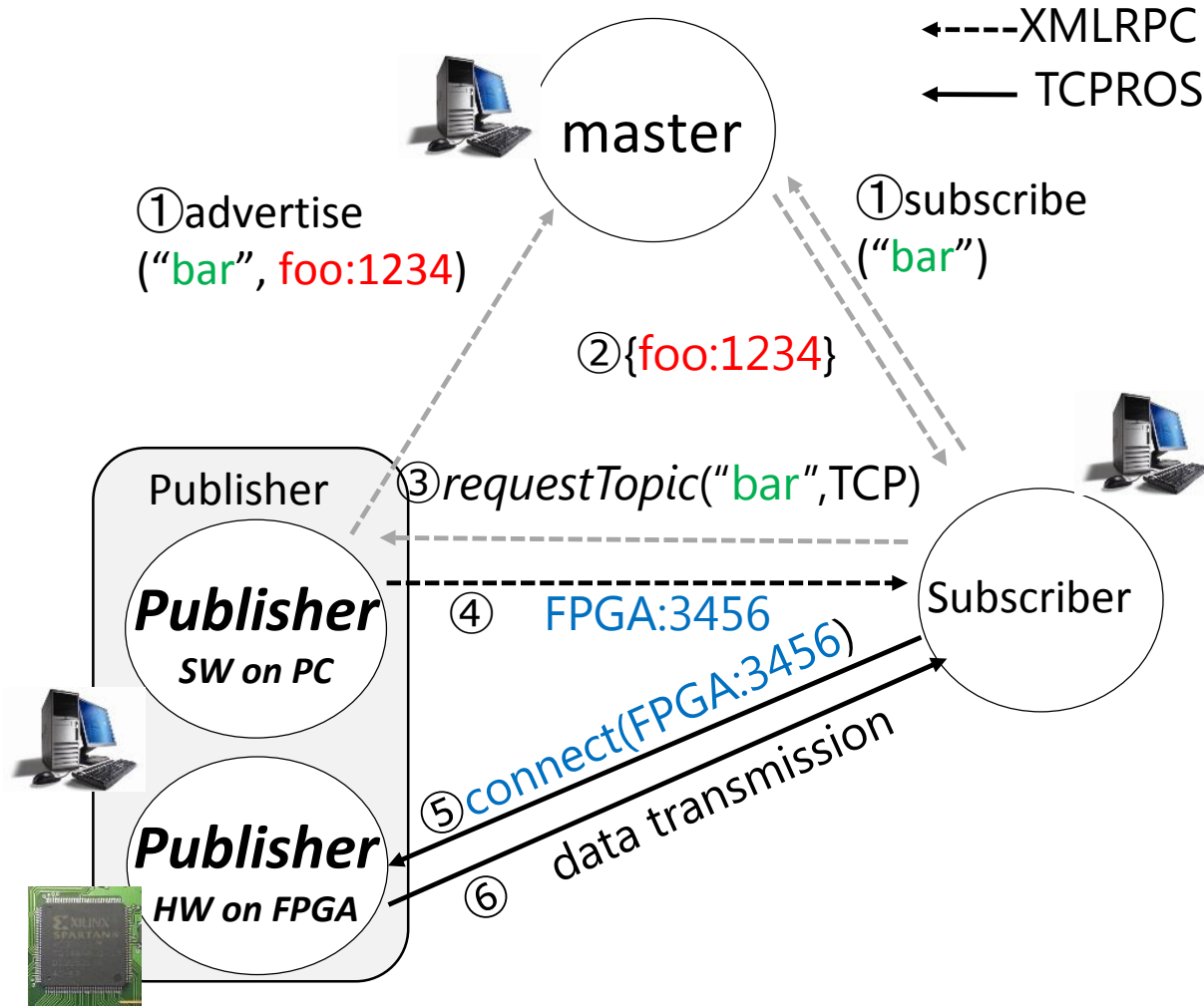
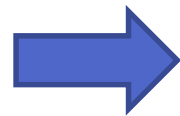
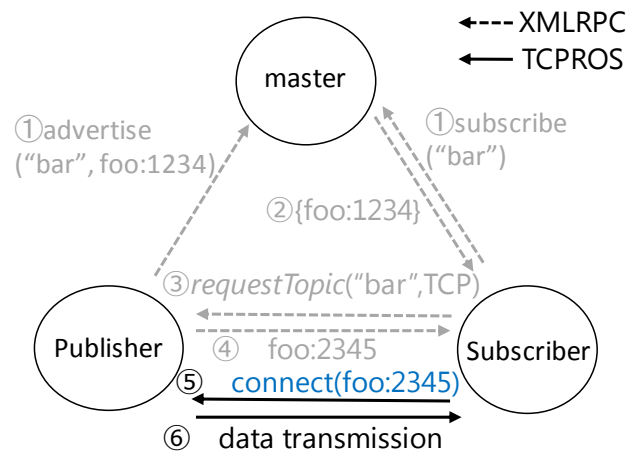
(3) Fully-hardwired ROS-compliant FPGA component

- Separation for Hardwired Publisher Node

Publisher SW
 topic name: "bar"
 hostname: foo
 XMLRPC port number: 1234

Publisher HW
 hostname :FPGA
 TCPROS port number: 3456

Subscriber
 subscribe to "bar"

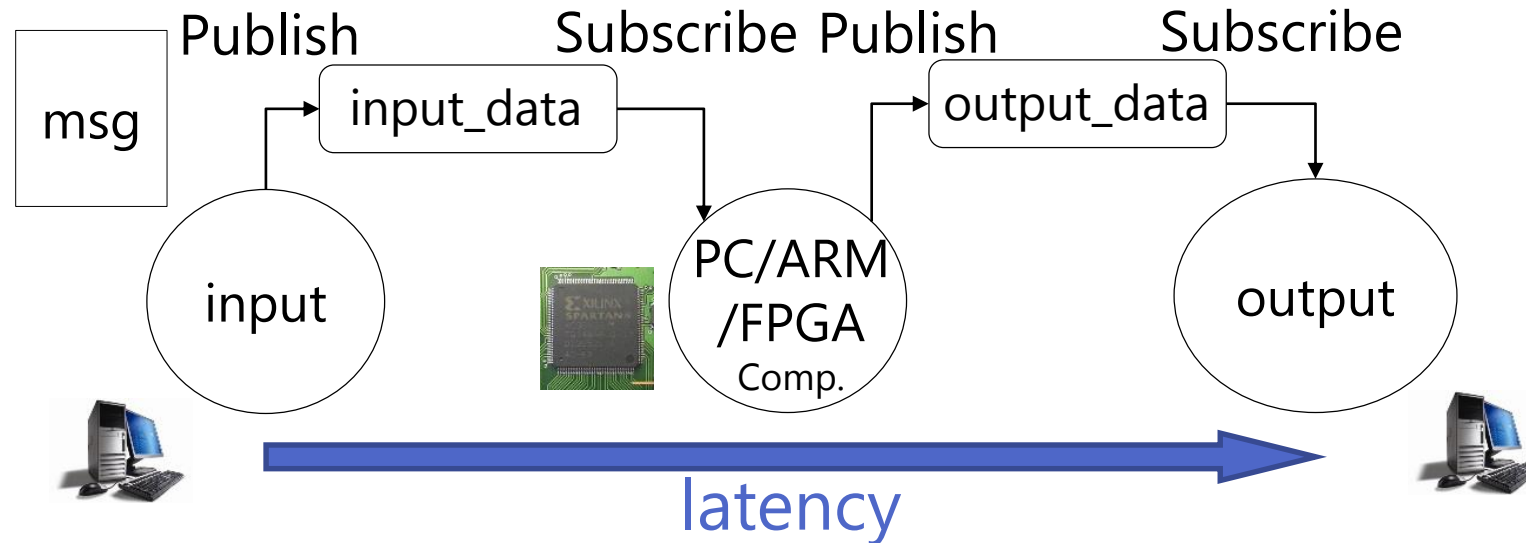


- Publisher HW can be implemented with 1 TCP/IP port!**

(3) Fully-hardwired ROS-compliant FPGA component

- Evaluation system

- Measurement: communication latency
- FPGA component does not process data (relay only)
- Used ROS message type: "sensor_msgs/image"
- Comparison: PC/ARM/FPGA



(3) Fully-hardwired ROS-compliant FPGA component

- Amount of hardware resource

- FPGA: Xilinx Spartan-6, XC6SLX110
 - Board: exTri-CSI (e-trees.Japan, Inc.)*
 - Two SiTCP** instances for Gigabit Ethernet ports

*FPGA does not process data (relay only)



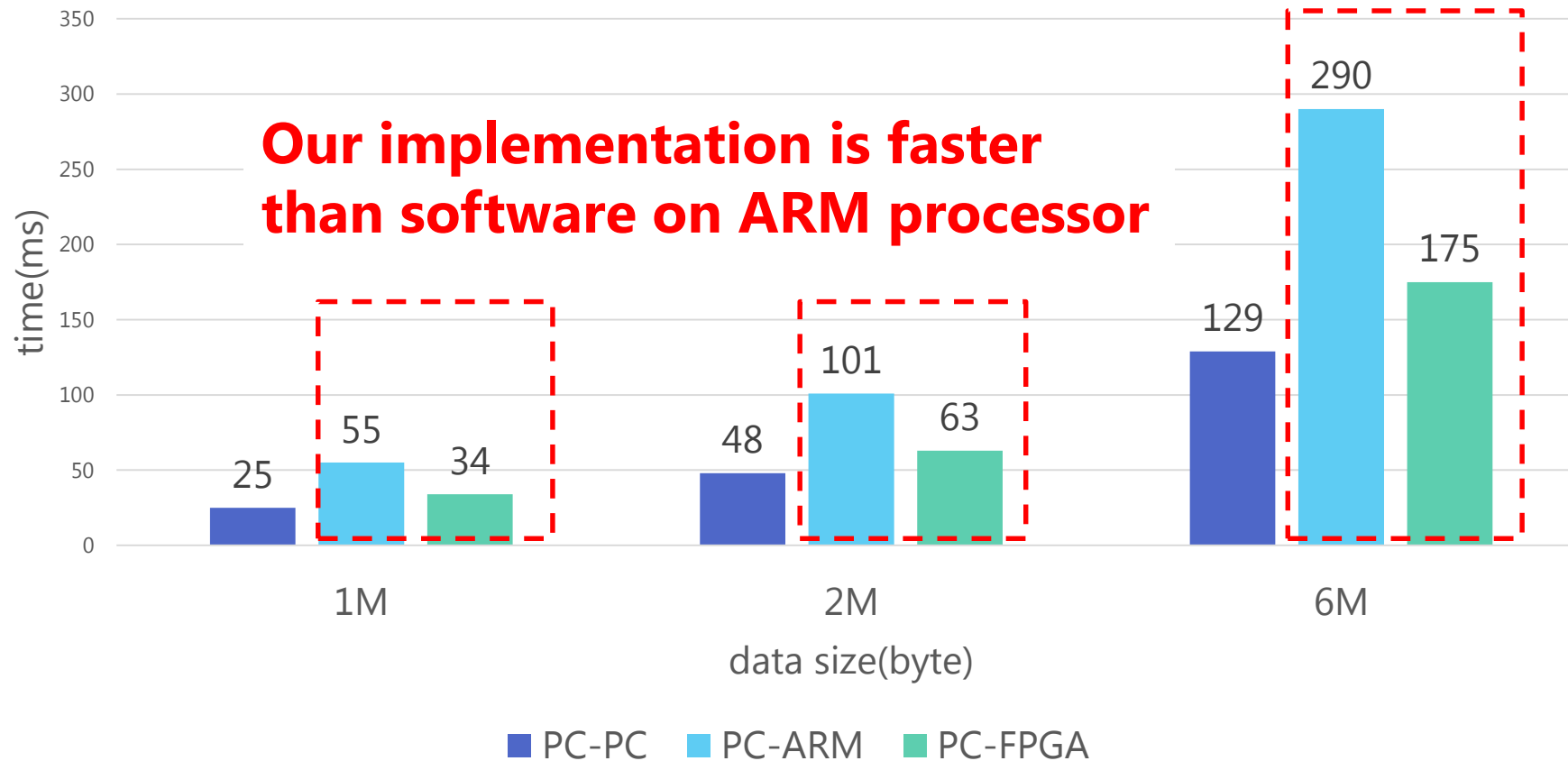
RESOURCE	UTILIZATION
Slice Registers	9253/126576 (7%)
Slice LUTs	7421/63288 (11%)
RAMB16BWER S	54/268 (20%)

* <http://e-trees.jp/products/extri-csi>

**T.Uchida: "Hardware-Based TCP Processor for Gigabit Ethernet," IEEE Transactions on Nuclear Science, Vol.55, No.SIG 3, pp.1631-1637, 2008.

(3) Fully-hardwired ROS-compliant FPGA component

- Evaluation results: Image data transfer (sensor_msg/image)



- **Throughput: 550Mbps**
 - **58% of maximum performance (SiTCP's MAX 949Mbps)**

Summary & Future work

- Proposal: Three types of ROS-compliant FPGA components
- Robot applications of the components
 - Object recognition using DNN architecture
 - SLAM: Distributed Visual-SLAM processing
 - Robot: **Image processing (ORB feature) on FPGA**
 - Cloud: SLAM processing at highly-parallel environment
- Challenges at RoboCup@Home2018!
 - Movie: RoboCup@Home2017
1st place Winner! at domestic standard platform league (TOYOTA HSR)



THANK YOU

References

- **POINTERS to Packages:**

- OpenReroc (Open source Reconfigurable Robot Component) project, which aims to promote the concept of ROS-compliant FPGA component
 - <http://kumikomi.github.io/OpenReroc/>
- Example implementation of Xilinx Zynq (ARM+FPGA) for Indigo packages at ros wiki
 - http://www.ros.org/browse/details.php?distro=indigo&name=openreroc_pwm
 - http://www.ros.org/browse/details.php?distro=indigo&name=openreroc_motion_sensor

- **URLs:**

- Home service robot project team "Hibikino-Musashi" for RoboCup@Home
<http://www.brain.kyutech.ac.jp/~hma/wordpress/about-us/>
- RoboCup@Home Movie <https://www.youtube.com/channel/UCJEeZZiDXijz6PidLiOtwwQ>
- RoboCup@Home Website <http://www.robocupathome.org/>
- Zynq-7000 All Programmable SoC, Xilinx:
<http://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html>

References (2/2)

- [1] **Takeshi Ohkawa**, Kazushi Yamashina, Takuya Matsumoto, Kanemitsu Ootsu, Takashi Yokota, "Architecture Exploration of Robot System using ROS-Compliant FPGA Component", Proc. 27th IEEE International Symposium on Rapid System Prototyping (RSP), pp.72-78, Oct. 2016. DOI: <http://dx.doi.org/10.1145/2990299.2990312>
- [2] **Yuhei Sugata**, **Takeshi Ohkawa**, Kanemitsu Ootsu and Takashi Yokota, "Acceleration of publish/subscribe messaging in ROS-Compliant FPGA Component," The 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies (HEART2017), 7-9 June 2017 (PDF is not available yet) <http://www.ruhr-uni-bochum.de/heart2017/program.html>
- [3] Kazushi Yamashina, **Takeshi Ohkawa**, Kanemitsu Ootsu and Takashi Yokota, "Proposal of ROS-compliant FPGA Component for Low-Power Robotic Systems - case study on image processing application," Proceedings of 2nd International Workshop on FPGAs for Software Programmers, FSP2015, pp. 62-67, 2015. <https://arxiv.org/ftp/arxiv/papers/1508/1508.07123.pdf>
- [4] Kazushi Yamashina, Hitomi Kimura, **Takeshi Ohkawa**, Kanemitsu Ootsu, Takashi Yokota, "cReComp: Automated Design Tool for ROS-Compliant FPGA Component," In proc. IEEE 10th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc-16), 2016.
- [5] A. Suzuki, T. Morie, **H. Tamukoh**, "FPGA Implementation of Autoencoders Having Shared Synapse Architecture," Proc. of the 23rd Int. Conf. on Neural Information Processing (ICONIP2016), Lecture Notes in Computer Science, Vol. 9947, pp.231-239, 2016.
- [6] S. Yokota, J. Li, Y. Ogishima, H. Kubo, **H. Tamukoh**, and M. Sekine, "Self-Learning of Feature Regions for Image Recognition," Journal of Computer Sciences and Applications, Vol.3, No.1, pp.1-10, 2015
- [7] **Yutaro Ishida**, et al., "Approach to accelerate the development of practical home service robots - RoboCup @Home DSPL", The 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2017), 2017.