ETRI
한국전자통신연구원
Electronics and Telecommunications Research Institute

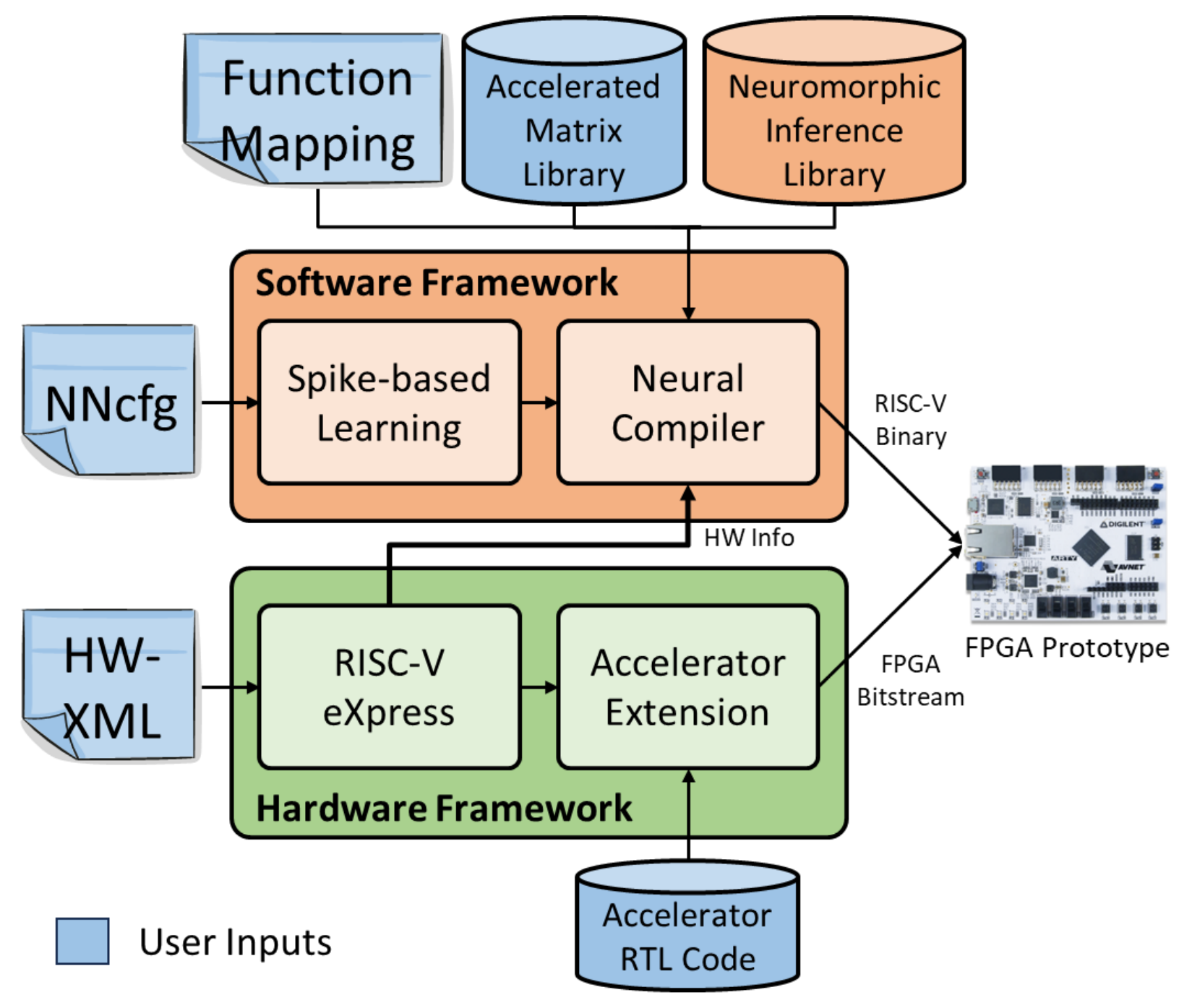
NPX: Automating Neuromorphic Processor Design from Spike-Based Learning to FPGA Prototyping

FPL 2025 Demo Night

Kyuseung Han, Hyeonguk Jang, Sukho Lee, Sung-Eun Kim, Kyudong Hwang, and Jae-Jin Lee
Electronics and Telecommunications Research Institute, Daejeon, South Korea

Neuromorphic Processor eXpress

- An extensible framework for developing lightweight neuromorphic processors
- Automates the entire design flow from spike-based learning to FPGA prototyping
- Demonstration of our FPL 2025 paper
  - “NeuGEMM: A Reordering-Free Unified GEMM-Conv2D Accelerator for Lightweight Neuromorphic Processors”
- Overall Framework



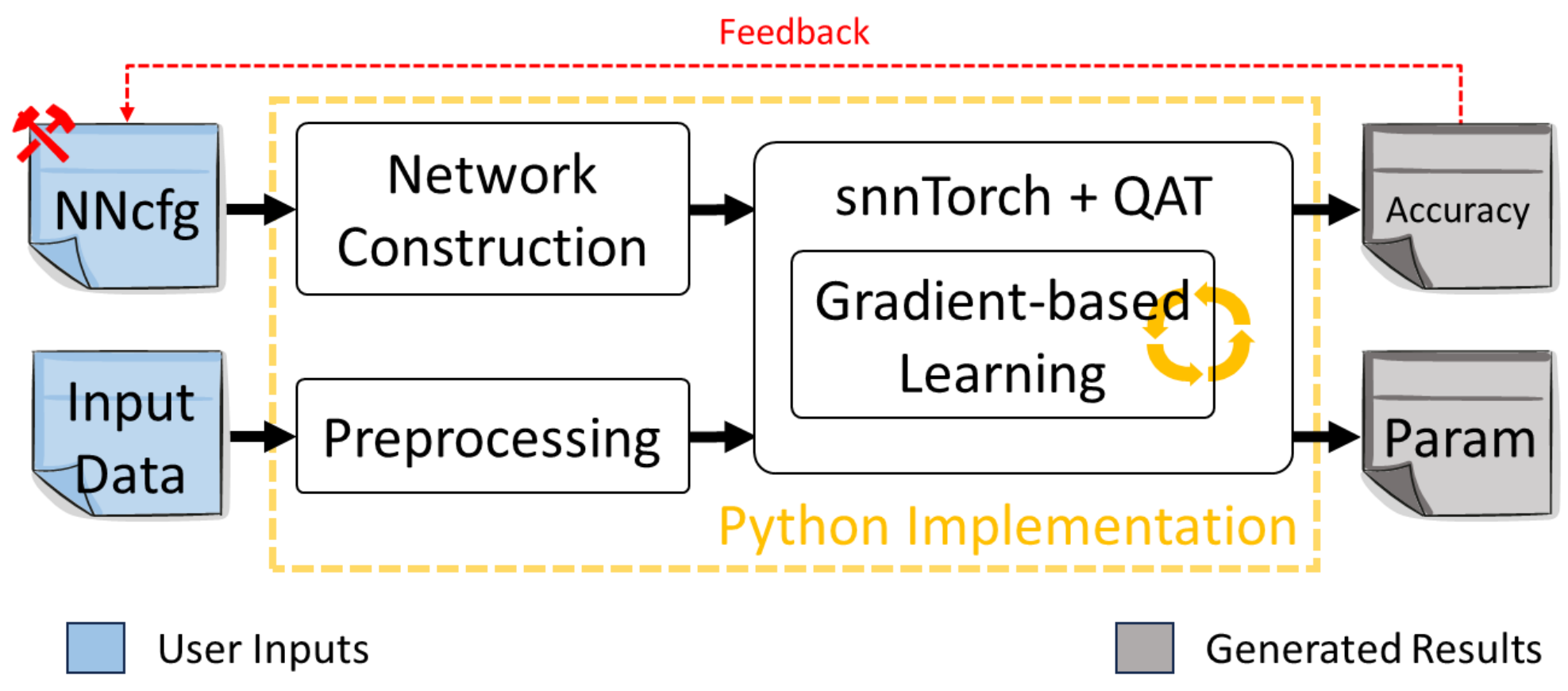
NN Configuration: NNcfg

- A text format originally used in Darknet (open source neural network framework written in C)
- Modified to support SNNs
- Used in both training (snnTorch) and inference (NIL)

```
[global]
neuron_type=q8
reset_mechanism=subtract

[Conv2d]
out_channels=12
kernel_size=5
[MaxPool2d]
kernel_size=2
[Leaky]

[Conv2d]
out_channels=32
kernel_size=5
[MaxPool2d]
kernel_size=2
[Leaky]

[Flatten]
[Linear]
out_features=10
[Leaky]
```

Spike-Based Learning

❖ **Overall Learning Process**



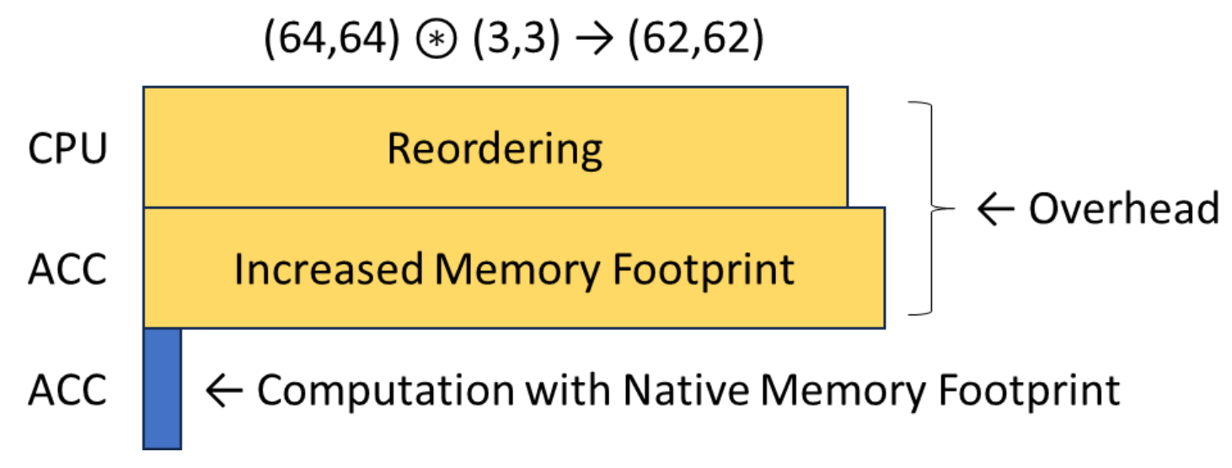
❖ **snnTorch**
- An extension of PyTorch for SNNs
- https://snntorch.readthedocs.io

❖ **Quantization-Aware Training (QAT)**
- Tailored for NPX processors
- Produces integer parameters for efficient inference

❖ **Network Construction from NNcfg**
- Parser converts NNcfg into snnTorch networks

NeuGEMM Accelerator

❖ **Data Reordering in Lightweight Processors**
- Transpose for GEMM & im2col for Conv2D
- Significant overhead due to limited CPU and memory bandwidth
- Execution time breakdown for Conv2D using im2col

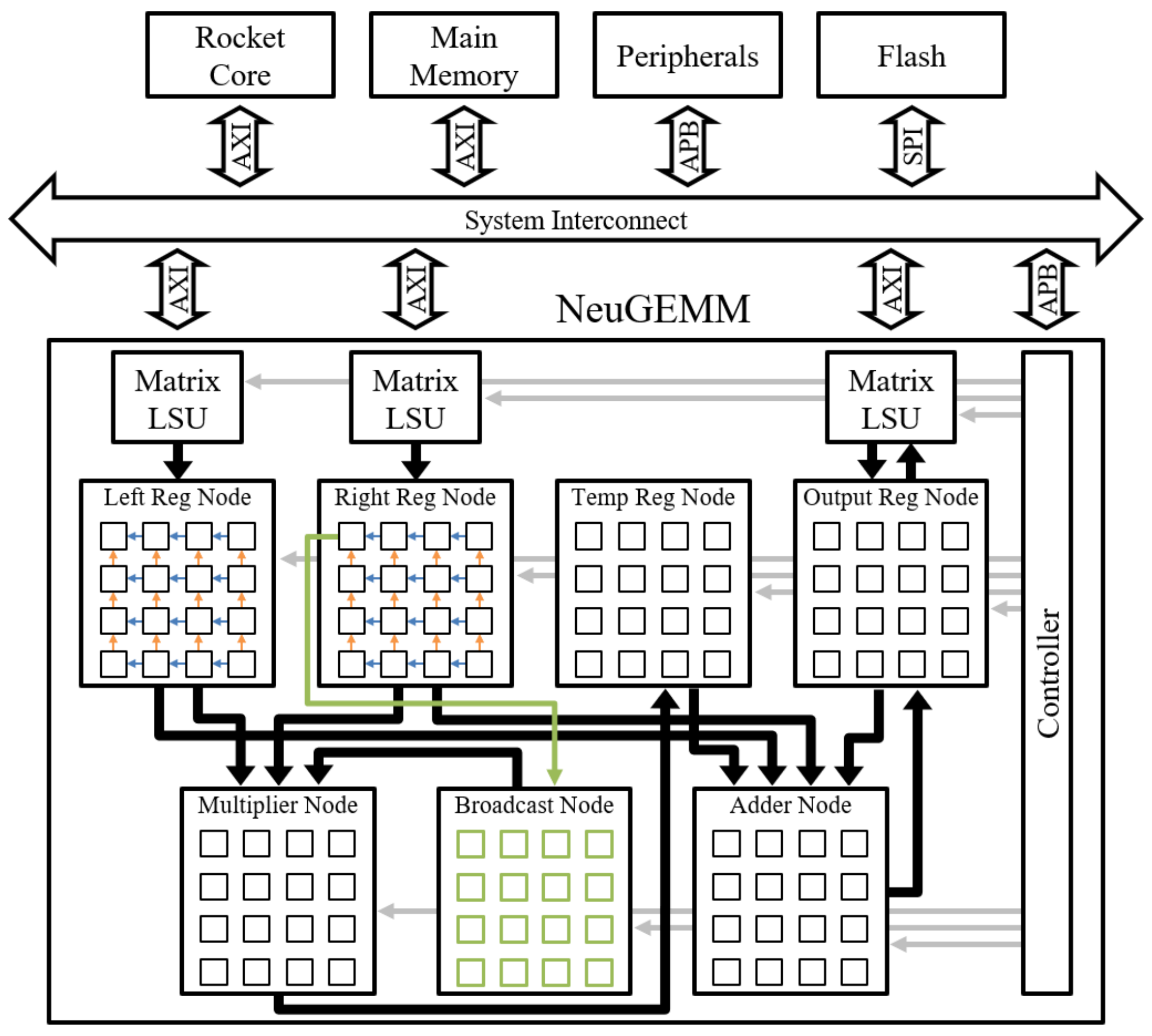


❖ **Reordering-Free Accelerator**
- Directly accesses arrays in the native C layout
- Supports Conv2D with minimal reconfigurability
- Results in more internal data movement overhead
- Balances workload across CPU, memory, and accelerators

❖ **Co-designed with Software**
- MatInfo struct describes matrices in the original C array layout
- NIL is built upon the MatInfo
- Matrix LSUs access C arrays according to MatInfo

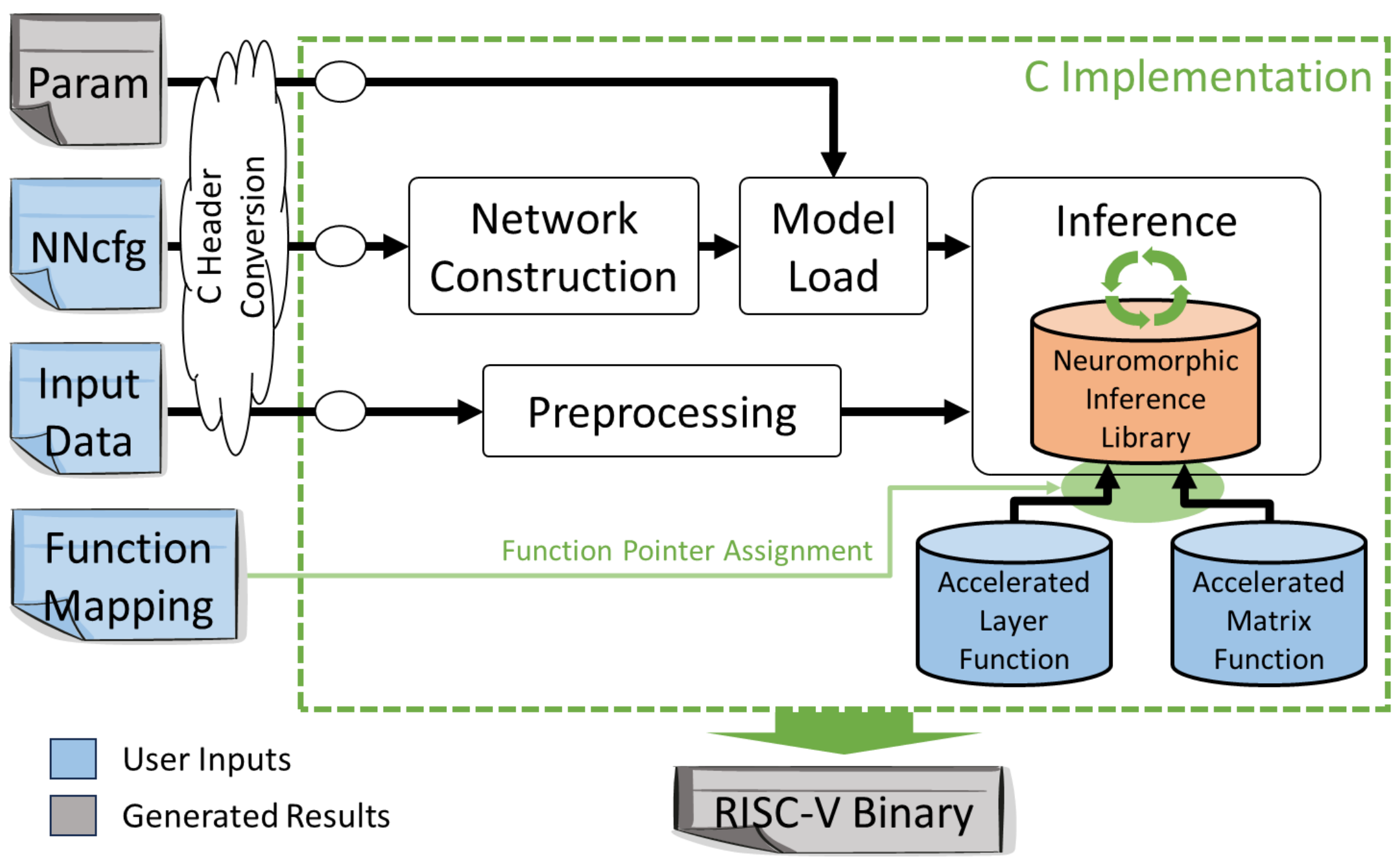
```
typedef struct
{
  void *addr;
  int stride;
  int num_row;
  int num_col;
  int datatype;
} MatInfo;
```

❖ **Unified Architecture for GEMM and Conv2D Acceleration**



Neural Compiler

❖ **RISC-V Binary Generation from Trained Model**



❖ **Post Process**
- Conversion of trained models (network structure and parameters) into C header files

❖ **Neuromorphic Inference Library (NIL)**
- C structs for supported layers
- NNcfg parser for layer structs
- Preprocessing functions for supported datasets
- Baseline forward functions for layer structs
  - Formulated using virtual matrix operations
- Baseline inference functions for networks
  - Sequential layer execution (-O0)
  - Based on virtual forward functions

❖ **Function Mapping**
- Replaces baseline functions with your accelerated ones

Neuromorphic Processor

❖ **Processor Design with RISC-V eXpress (RVX)**
- Automatic generation of processor RTL from HW-XML

```
<rvx>
 <platform>
   <name>starc_neugemm</name>
   <status>described</status>
   <spec>
     <define>
       <name>sram_size</name>
       <value>64kB</value>
     </define>
     <define>
       <name>include_slow_dram</name>
       <value>True</value>
     </define>
     <define>
       <name>num_spi</name>
       <value>2</value>
     </define>
   </spec>

   <ip_instance>
     <name>i_dca_neugemm00</name>
     <library_name>dca_neugemm</library_name>
     <parameter>
       <id>MATRIX_SIZE_PARA</id>
       <value>8</value>
     </parameter>
   </ip_instance>

   <ip_instance>
     <name>i_main_core</name>
     <library_name>rvc_rocket_big</library_name>
   </ip_instance>

 </platform>
</rvx>
```

- RTL Simulation Support
  - Automatically generates simulation scripts
  - Supports Cadence Xcelium and Siemens QuestaSim
- FPGA Prototyping Support
  - Automatically generates AMD Vivado scripts
  - Includes preconfigured Vivado components for supported FPGA boards (e.g. slow_dram)
- Accelerator Extension
  - Supports attachment of your own accelerators
  - NeuGEMM is provided as a built-in accelerator

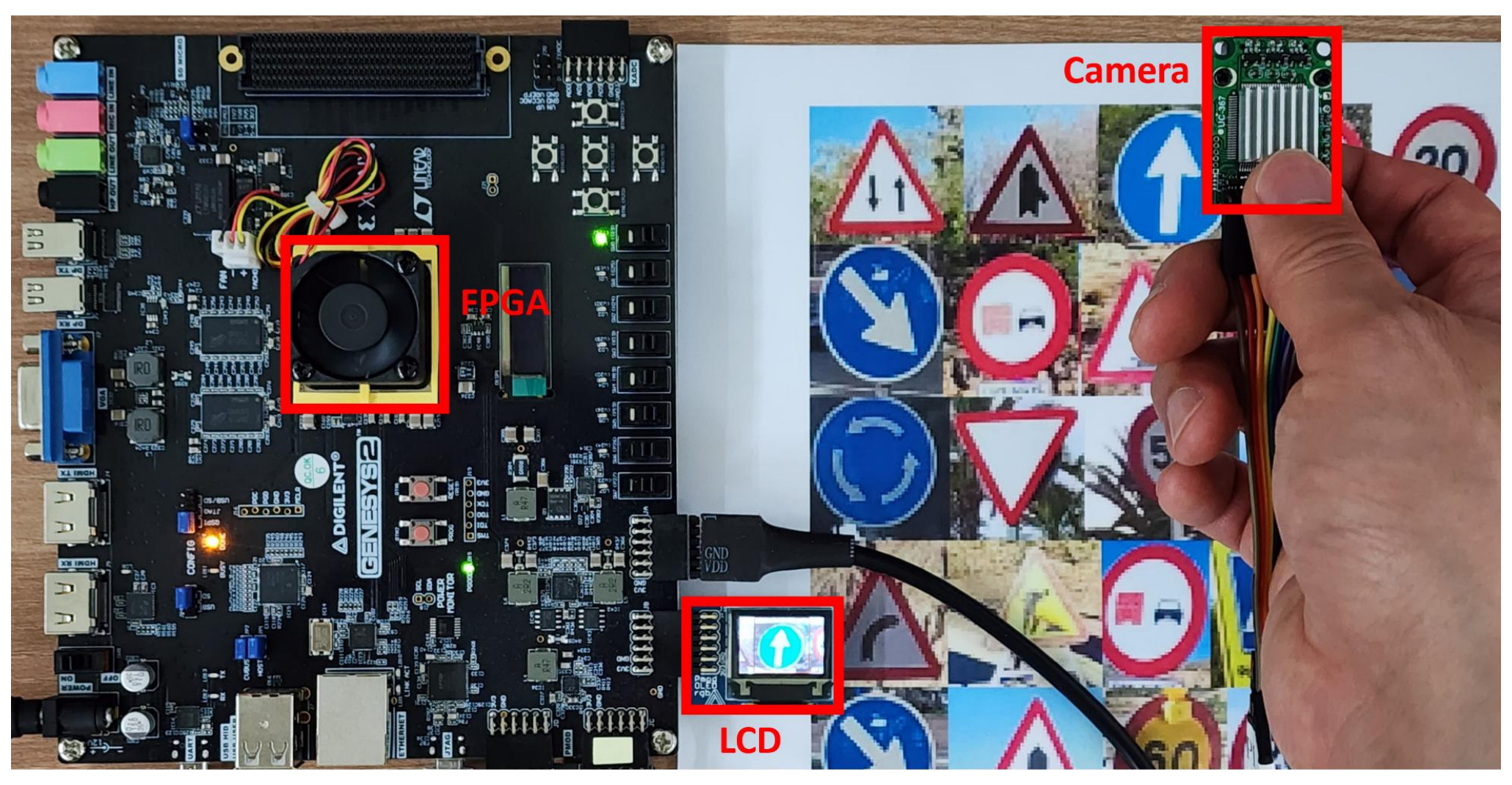
A Case Study: Traffic Sign Recognition System

❖ **Customization of the Baseline Processor**
- HW-XML configured to include two SPI ports for camera and display

❖ **Camera and Display Connection to FPGA Board**
- Digilent Genesys2 FPGA board
- Arducam 5MP Plus OV5642 camera
- Digilent OLEDrgb display

❖ **Development of Application Software**
- Handles camera and display via SPI
- Preprocesses the captured image for network input

❖ **33× Faster Inference**
- 202,418 ms (baseline) → 6,096 ms (with NeuGEMM)



https://riscvexpress.github.io/npx
Contact: ceicarus@etri.re.kr (Jae-Jin Lee)