Data Sheet

CU2026S

Chengwen Chu, Yizheng Tang, Linxiao Wu, Chenchen Xu

Systolic-Array Accelerator for Rock–Paper–Scissors Gesture Recognition

FEATURES

- Fixed LeNet-like CNN for 64×64×1 int8 hand-gesture images (Rock / Paper / Scissors / optional Stop).
- 3×3 INT8 systolic convolution array (9-PE core)
- Support for two 3×3 convolution layers (Cin=1→Cout=4, Cin=4→Cout=8) plus 1803→3/4 fully connected (FC) layer.
- Total compute per inference ≈82 kMAC at 8-bit precision.
- On-chip SRAM ≈50 KB for input image, feature buffers, line buffers, and weights.
- SPI interface: 4 KB image input.
- Hierarchical hardware control: top-level ASIC FSM and CNN-layer FSM.
- Target core clock frequency: 10 MHz for bring-up, 25–50 MHz in 65 nm (TBD after PNR).
- Small logic and SRAM footprint suitable for student tape-out projects (exact gate count TBD).

APPLICATIONS

- Rock–Paper–Scissors gesture-recognition game with FPGA front-end and HDMI/VGA UI.
- Low-resolution (64×64) hand-gesture classifiers for embedded HMI or toys.
- Example platform for teaching CNN accelerators, systolic arrays, and FPGA–ASIC co-design.
- Generic 64×64 image classifier where a small, fixed CNN is sufficient.

GENERAL DESCRIPTION

The Team5-CNN chip is a pure-hardware CNN accelerator optimized for 64×64 grayscale hand-gesture images. The device receives a 4 KB image over an SPI interface from an FPGA front-end, stores it in an on-chip input buffer, and performs all convolutional and fully connected computations using a shared 3x3 INT8 systolic array. The fixed network topology consists of two 3×3 convolution layers with ReLU and 2×2 pooling, followed by a fully connected layer that produces 3 or 4 gesture logits (Rock / Paper / Scissors / optional Stop).

A hierarchical control scheme coordinates the computation. A top-level ASIC FSM handles SPI transactions, weight loading, and the start/end of inference, while a CNN FSM sequences Conv1, Pool1, Conv2, Pool2, FC, and Argmax using ping-pong feature buffers and compact window buffers for pooling. The fully connected layer flattened activations and weights through the systolic array to accumulate four logits in parallel.

The chip targets at least 10 inferences per second for a total compute rate of ≈0.82 MMAC/s at 10 MHz clock frequency, assuming one inference per image frame from the FPGA. All CNN parameters (activations and weights) are quantized to 8-bit integers. The design focuses on low area and simple integration with an FPGA, making it suitable as a teaching/experimental ASIC accelerator core.

FUNCTIONAL BLOCK DIAGRAM

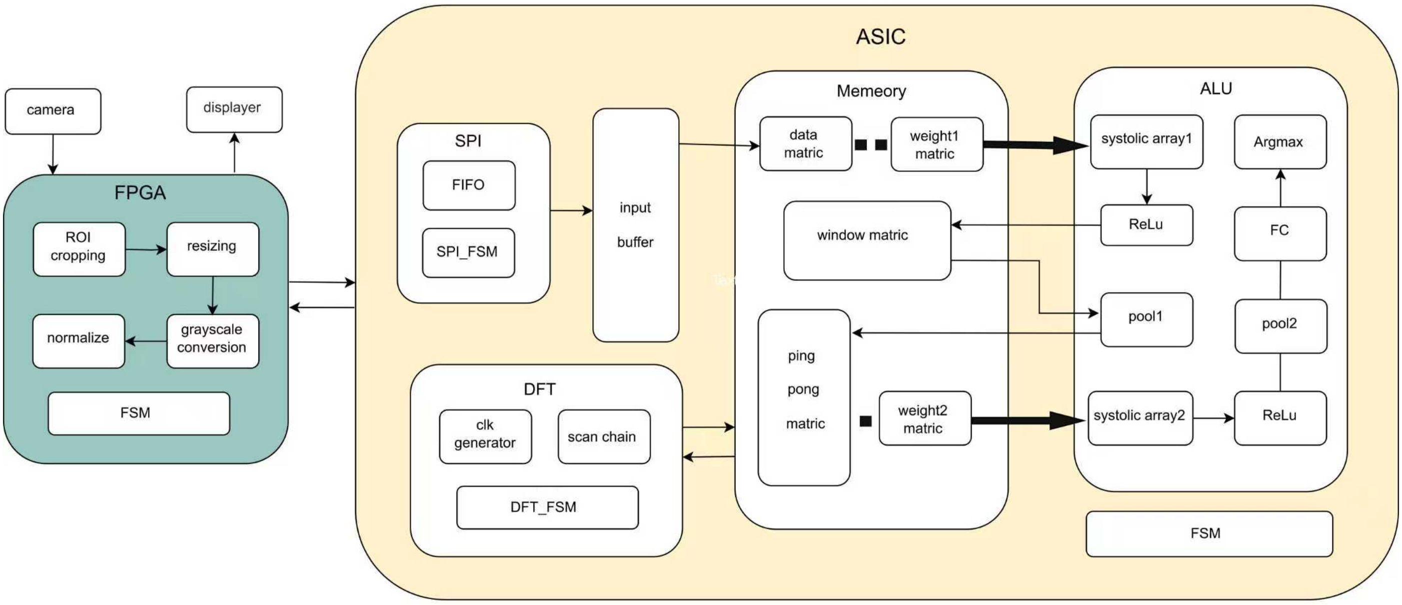
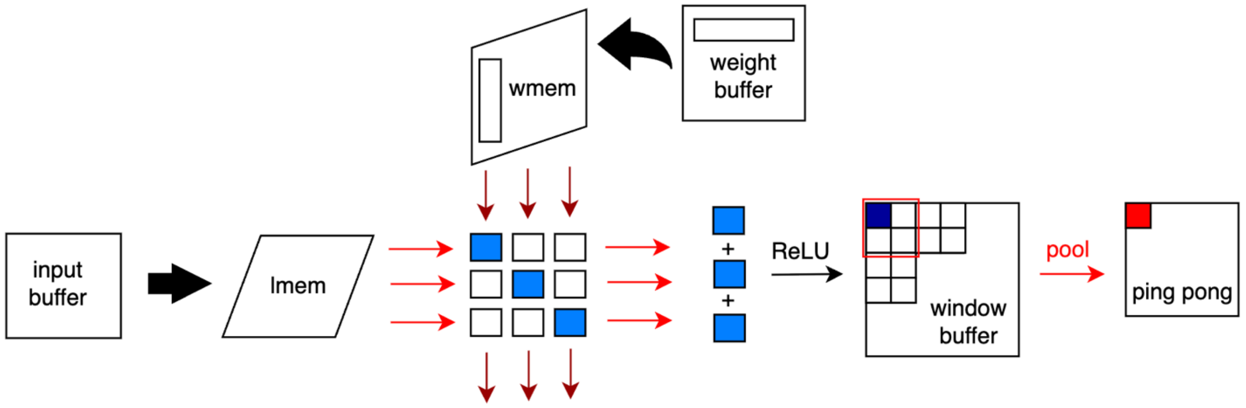


Figure. 1

DATAFLOW



SPECIFICATION

Table.1 On-Chip Memory Blocks

Block	Size	Purpose
Input img64 buffer	4KB	Stores the 64×64 int8 input image
Feature buffers (ping-pong)	5.5KB	Intermediate CNN feature maps for Conv/Pool.
Window buffer1 & 2	2.8KB	Stores ReLU outputs
Weights (SRAM)	7.1KB	Store weights for Conv1, Conv2, FC
Lmem / wmem	0.015KB	Rhombus buffers feeding the 3×3 systolic MAC array
fmem	1.8KB	Flattens the ping-pong feature maps for FC
Total	≈50 KB	Sufficient for one complete inference

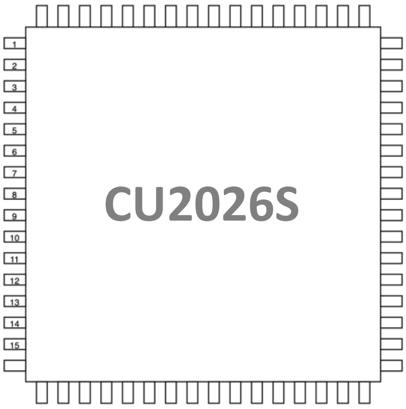
Table.2 CNN Layer Configuration

Layer	Configuration	Input → Output Size	Compute Role
Input	Grayscale image	64×64×1 (int8)	CNN input frame.
Conv1	3×3, Cin=1, Cout=4, stride 1	64×64×1 → 62×62×4	Low-level feature extraction.
Pool1	2×2 max-pool, stride 2	62×62×4 → 31×31×4	Spatial down-sampling.
Conv2	3×3, Cin=4, Cout=8, stride 1	31×31×4 → 30×30×8	Higher-level feature extraction.
Pool2	2×2 max-pool, stride 2	30×30×8 → 15×15×8	Further reduction.
FC	Fully connected	1803 → 3 (or 4)	Final classification (logits).

Table.3 System-Level Data Interfaces

Stage / Interface	Data Format	Data Size / Rate	Design Implication
Camera → FPGA	640×480 @ 30 fps, RGB565	~18.4 MB/s	Handled entirely by FPGA I/O and preprocessing.
FPGA preprocessing out	64×64 grayscale (int8)	4 KB per frame	Reduced data size for ASIC inference.
FPGA → ASIC (SPI)	Serialized img64 buffer	4 KB / inference	One 64×64 frame is transferred per inference.
Required SPI bandwidth	–	≥10 KB/s (10 fps target)	Minimum bandwidth for real-time operation.
Implemented SPI	SPI @ 5 MHz	~500 KB/s	>×50 margin over the requirement.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



Pin Name	Direction	Description
1 PIN	I/O	General-purpose pin / reserved
2 CLK_EXT	In	External system clock for the ASIC core
3 RST_N	In	Active-low global reset
4 SPI_SCLK	In	SPI serial clock from FPGA (master)
5 SPI_CSN	In	SPI chip select (active low)
6 SPI_MOSI	In	SPI master-out slave-in data line
7 SPI_MISO	Out	SPI master-in slave-out data line
8 SCAN_EN	In	Scan enable (DFT mode select)
9 SCAN_IN	In	Scan chain serial input
10 SCAN_OUT	Out	Scan chain serial output
11 SCAN_CLK	In	Scan clock input
12 VDD_CORE	In	Core logic power supply
13 VDD_IO	In	I/O power supply (SPI and control pins)
14 VDD_TEST	In	Test / scan power supply
15 VSS	In	Ground reference

FSM CONTROL DIAGRAM

The chip is orchestrated by two cooperating FSMs: an ASIC system FSM and a CNN accelerator FSM.

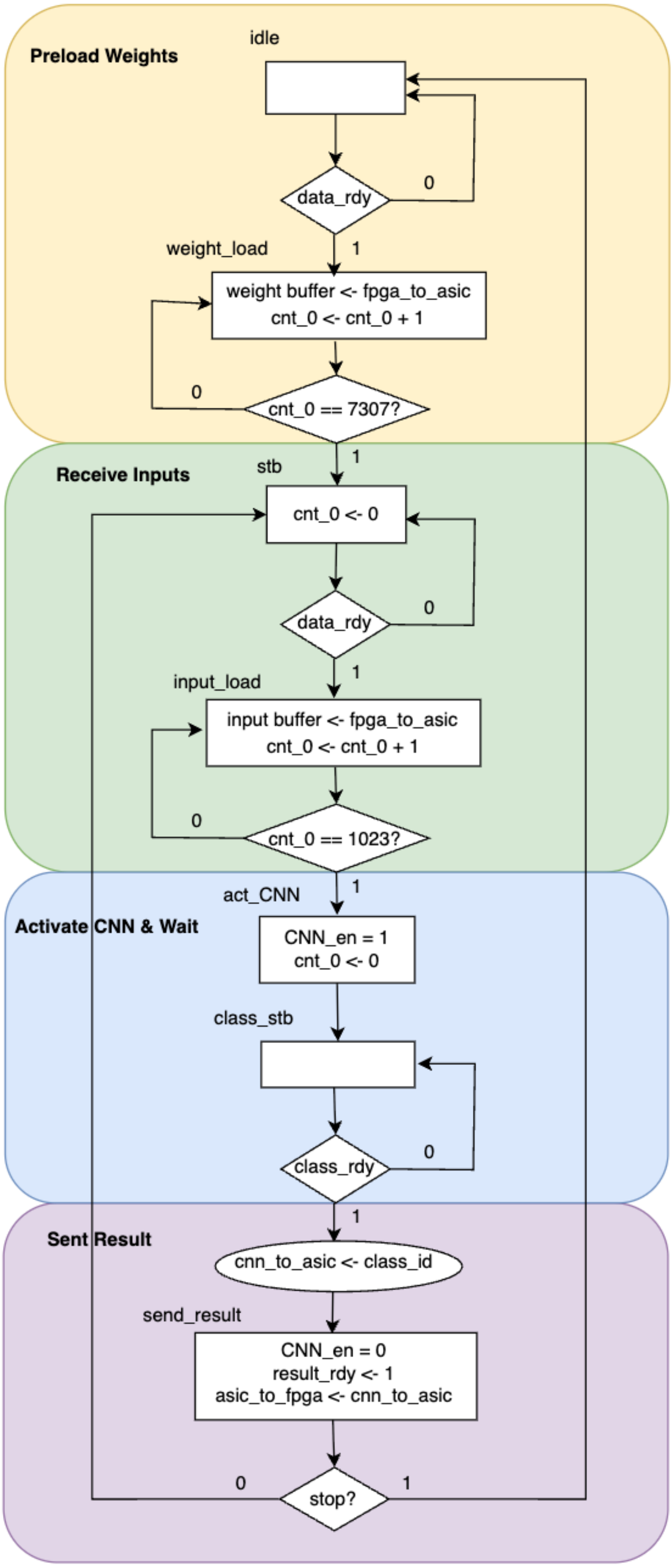


Figure.2 ASIC FSM

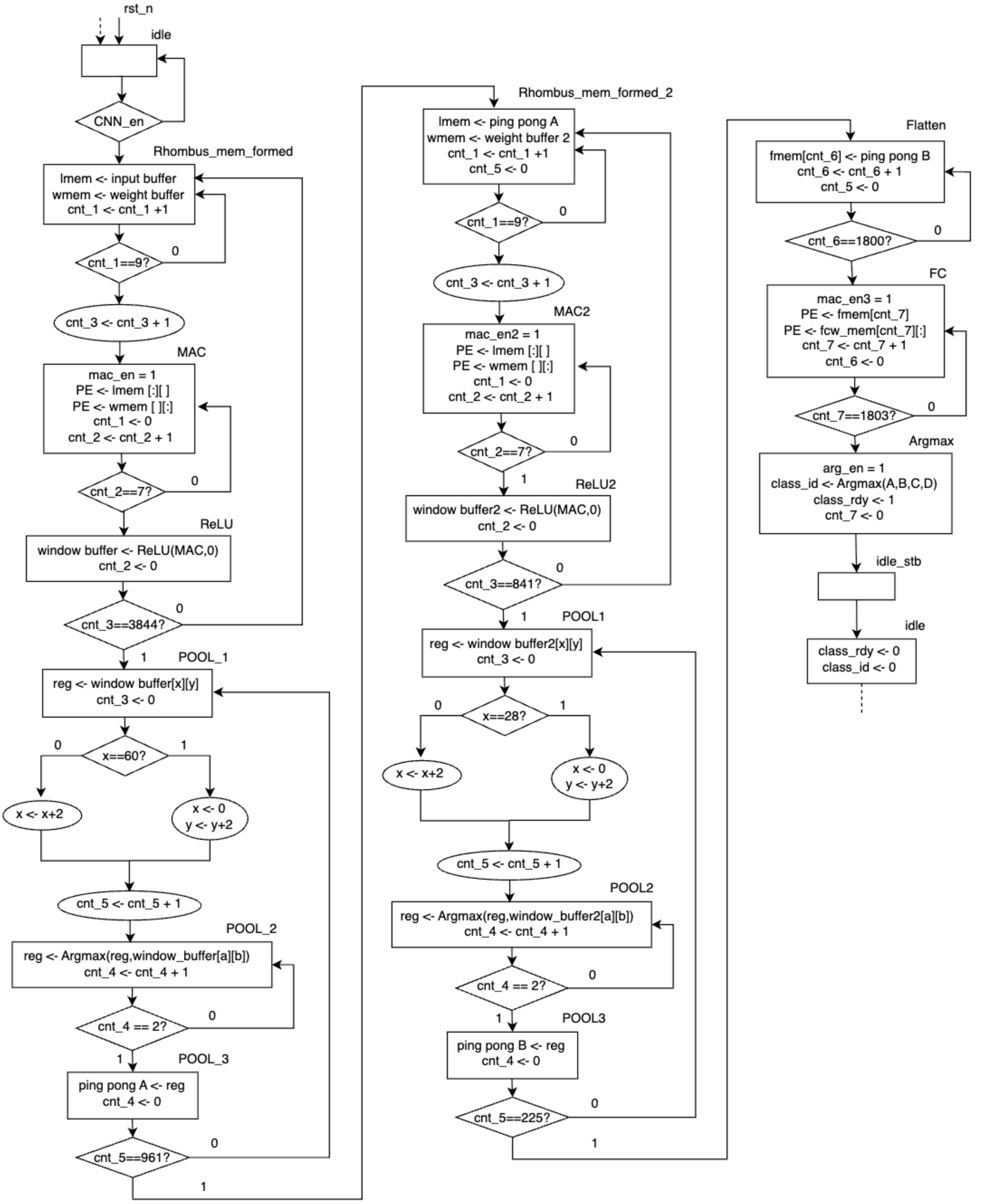


Figure.3 CNN FSM