An Energy-Efficient CNN Accelerator Specialized for PYNQ-Z2 FPGA Device

Chao Chia Lin, Chih Lun Chen, Min Chia Chen
National Taiwan University of Science and Technology
Department of Electronic and Computer Engineering

Introduction

The aim of this study is to build a low power CNN accelerator specialized for PYNQ-Z2. The primary objectives include maximizing resource utilization and minimizing off-chip communication. We built a completed CNN model which consists of Convolution, Max Pooling and Fully Connected layers by pure logical circuits. The result was verified by the equivalent software-based model, the hardware accelerator demonstrates a fast inference time of approximately 100 FPS, which successfully meeting the real-time processing requirements.

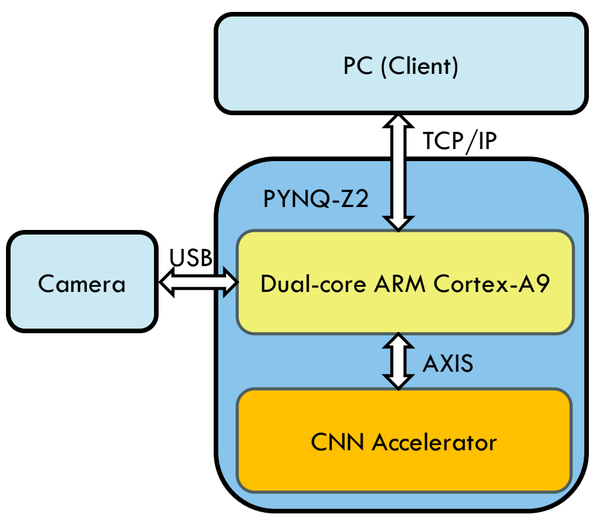

Figure 1: System architecture

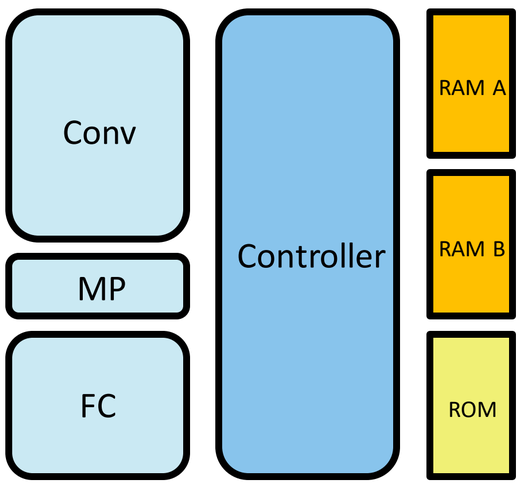

Figure 2: PL architecture

Convolution Strategy

Convolutional Neural Networks (CNNs) have emerged as a cornerstone in deep learning architectures, demonstrating remarkable efficacy across diverse applications. In this study, we first trained a basic CNN model which performs gender classification. Base on the framework of convolution layer, we construct our hardware accelerator accordingly as shown in the following figure:

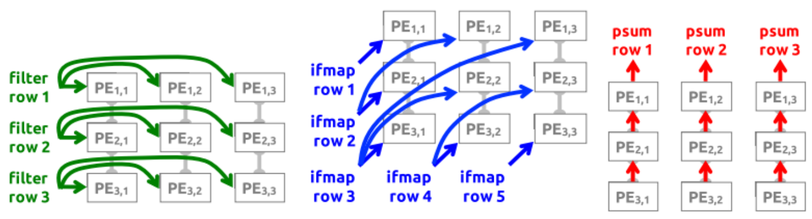

Figure 3: Dataflow in PEs

Efficient partial sum (psum) accumulation, minimizing movement, becomes essential to conserve storage space and memory read/write energy. However, achieving maximum input data reuse while immediately reducing psums is challenging, as psums generated by Multiply and Accumulate (MAC) operations using the same filter or ifmap value are not readily reducible. The RS dataflow employs a systematic approach, optimizing for all data types simultaneously. This involves completing the psum of the output feature map at the earliest possible instance. Through time interleaving of filter and ifmap rows, each Processing Element (PE) can process multiple 1-D primitives from different channels and accumulating psums.

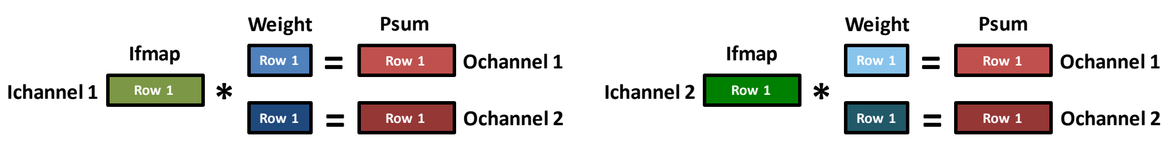

Figure 4: Minimize data movement strategy

Our Convolution circuit is constructed to be size-flexible for every layer, by setting the parameters as input from Controller. Beside Convolution reuse, RAM reuse is another cost saving strategy by reducing the number of RAM.

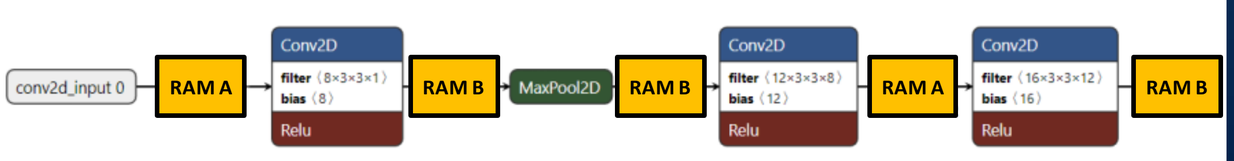

Figure 5: RAM reuse strategy

Fully Connected Layer Strategy

The first layer after flattening require the most computation resources and memory cost. It is not possible to load every feature map and weight into the chip and perform cross-multiplication at the same time. To balance the hardware requirement between layers, we apply partial sum strategy, which load a small amount of data every stage and accumulate the multiplication result. For the last two layers, we reuse the same hardware and infer each output node immediately without any accumulation.

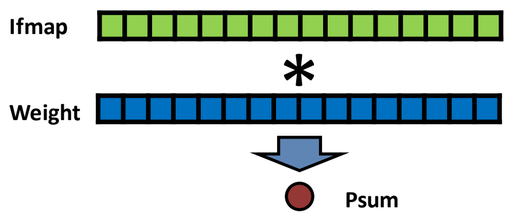

Figure 6: Resource reuse in FC

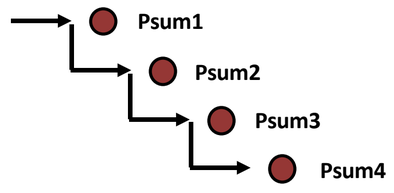

Figure 7: Psum accumulation

Result

The implemented architecture on the PYNQ-Z2 development board has undergone synthesis, yielding the following resource utilization metrics: Slice LUTs at 41%, Slice Registers at 7%, Block RAM at 30%, and DSP at 25%. A maximum clock frequency of **50MHz** was achieved, successfully meeting the specified timing constraints. The obtained results indicate that the total inference time for a single frame is 267,000 cycles, which is equivalent to 0.5ms.

Name	Slice LUTs (53200)	Slice Registers (106400)	F7 Muxes (26600)	F8 Muxes (13300)	Slice (13300)	LUT as Logic (53200)	LUT as Memory (17400)	Block RAM Tile (140)	DSPs (220)	Bonded IOPADs (130)	BUFGCTRL (32)
N Lab_final_bd_wrapper	40.65%	16.88%	4.61%	3.37%	49.41%	40.34%	0.94%	30.36%	25.45%	100.00%	3.13%
I Lab_final_bd_i (Lab_final_bd)	40.65%	16.88%	4.61%	3.37%	49.41%	40.34%	0.94%	30.36%	25.45%	0.00%	3.13%

Figure 8: Total utilization

NN_1 (NN_1_imp_1SXG1K6)	36.33%	13.77%	4.61%	3.37%	42.56%	36.33%	0.00%	28.93%	25.45%	0.00%	0.00%
blk_ram_temp (Lab_final_	0.02%	<0.01%	0.00%	0.00%	0.10%	0.02%	0.00%	11.43%	0.00%	0.00%	0.00%
blk_rom_inimg (Lab_final_	0.03%	<0.01%	0.00%	0.00%	0.10%	0.03%	0.00%	11.43%	0.00%	0.00%	0.00%
blk_rom_other_weight (La	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.36%	0.00%	0.00%	0.00%
blk_rom_weight (Lab_final	0.04%	<0.01%	0.03%	0.00%	0.08%	0.04%	0.00%	5.71%	0.00%	0.00%	0.00%
ConV1_0 (Lab_final_bd_C	25.38%	10.96%	4.58%	3.37%	30.21%	25.38%	0.00%	0.00%	3.64%	0.00%	0.00%
FC_2_0 (Lab_final_bd_FC	8.93%	1.98%	0.00%	0.00%	10.58%	8.93%	0.00%	0.00%	21.82%	0.00%	0.00%
MP1_0 (Lab_final_bd_MP1	1.69%	0.80%	0.00%	0.00%	2.08%	1.69%	0.00%	0.00%	0.00%	0.00%	0.00%
MUX_mem_out_0 (Lab_fir	0.24%	0.02%	0.00%	0.00%	0.53%	0.24%	0.00%	0.00%	0.00%	0.00%	0.00%

Figure 9: Utilization of each component in CNN

Setup

Worst Negative Slack (WNS):	1.136 ns	Worst Hold Slack (WHS):	0.033 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	29575	Total Number of Endpoints:	29575

Hold

Pulse Width

Worst Pulse Width Slack (WPWS):	9.500 ns
Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0
Total Number of Endpoints:	14787

All user specified timing constraints are met.

Figure 10: Timing report

Intra-Clock Paths - sys_clk - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	1.136	23	9	8	NN_bd_i/FC_2_0..._reg[22][5]/C	NN_bd_i/FC_2_0/...cti_reg[7][4]/D	18.797	11.649	7.148	20.0	sys_clk
Path 2	1.183	23	9	8	NN_bd_i/FC_2_0..._reg[22][5]/C	NN_bd_i/FC_2_0/...cti_reg[7][7]/D	18.747	11.649	7.098	20.0	sys_clk
Path 3	1.201	23	9	8	NN_bd_i/FC_2_0..._reg[22][5]/C	NN_bd_i/FC_2_0/...cti_reg[7][2]/D	18.732	11.649	7.083	20.0	sys_clk
Path 4	1.253	23	9	8	NN_bd_i/FC_2_0...uf_reg[3][6]/C	NN_bd_i/FC_2_0/...cti_reg[3][4]/D	18.685	11.628	7.057	20.0	sys_clk
Path 5	1.258	23	9	8	NN_bd_i/FC_2_0...uf_reg[3][6]/C	NN_bd_i/FC_2_0/...cti_reg[3][5]/D	18.682	11.628	7.054	20.0	sys_clk
Path 6	1.308	23	9	8	NN_bd_i/FC_2_0...uf_reg[3][6]/C	NN_bd_i/FC_2_0/...cti_reg[3][1]/D	18.631	11.628	7.003	20.0	sys_clk
Path 7	1.313	23	9	8	NN_bd_i/FC_2_0...uf_reg[3][6]/C	NN_bd_i/FC_2_0/...cti_reg[3][3]/D	18.628	11.628	7.000	20.0	sys_clk
Path 8	1.323	23	9	8	NN_bd_i/FC_2_0..._reg[22][5]/C	NN_bd_i/FC_2_0/...cti_reg[7][0]/D	18.608	11.649	6.959	20.0	sys_clk
Path 9	1.328	23	9	8	NN_bd_i/FC_2_0..._reg[22][5]/C	NN_bd_i/FC_2_0/...cti_reg[7][1]/D	18.605	11.649	6.956	20.0	sys_clk
Path 10	1.367	24	9	8	NN_bd_i/FC_2_0..._reg[16][5]/C	NN_bd_i/FC_2_0/...cti_reg[5][3]/D	18.534	11.867	6.667	20.0	sys_clk

Figure 11: Critical paths

Position	Clock Name	Period (ns)	Rise At (ns)	Fall At (ns)	Add Clock	Source Objects	Source File
1	sys_clk	20.000	0.000	10.000	☐	[get_ports sys_clk]	cc.xdc
Double click to create a Create Clock constraint							

Figure 12: Clock constraint

Future work

It is important to note that we did not consider the data transferring network, which is crucial in most of the computational application. They can be achieved with Network on Chip (NoC) and efficient communication system between PEs. Additionally, future iterations could explore the integration of more advanced models, such as ResNet or YOLO, to broaden the versatility of the CNN accelerator.