DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

A Technology-Independent Toolflow for Automating AI Deployment on FPGAs for On-board Satellite Applications

Presentation for SpacE FPGA Users Workshop (SEFUW) 2023

📍 European Space Research and Technology Centre (ESTEC)

Authors: Tommaso Pacini (Presenter)
Emilio Rapuano
Pietro Nannipieri
Prof. Luca Fanucci

Presentation Outline

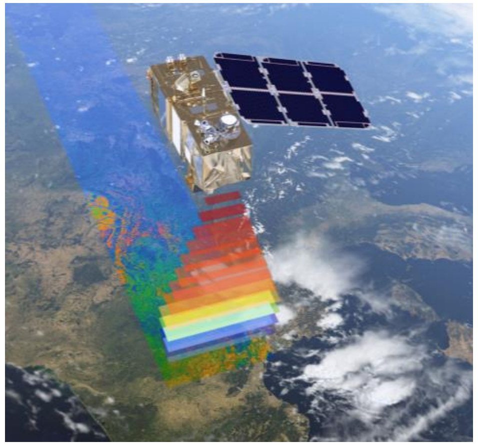
- Research Context
- DNN-to-FPGA Toolflows
- FPG-AI: an Automation Toolflow for CNNs
- Conclusions

Presentation Outline

- Research Context
- DNN-to-FPGA Toolflows
- FPG-AI: an Automation Toolflow for CNNs
- Conclusions

Artificial Intelligence for Space Missions

- **Growing interest in AI for space applications:**
  - Weather & Atmospheric Monitoring
  - Object Detection & Tracking
  - Ground Classification
  - FDIR for Reliability
  - Autonomous Spacecraft Navigation



- **AI Acceleration: Ground Station vs On-board Processing**

GROUND STATION
DATA PROCESSING

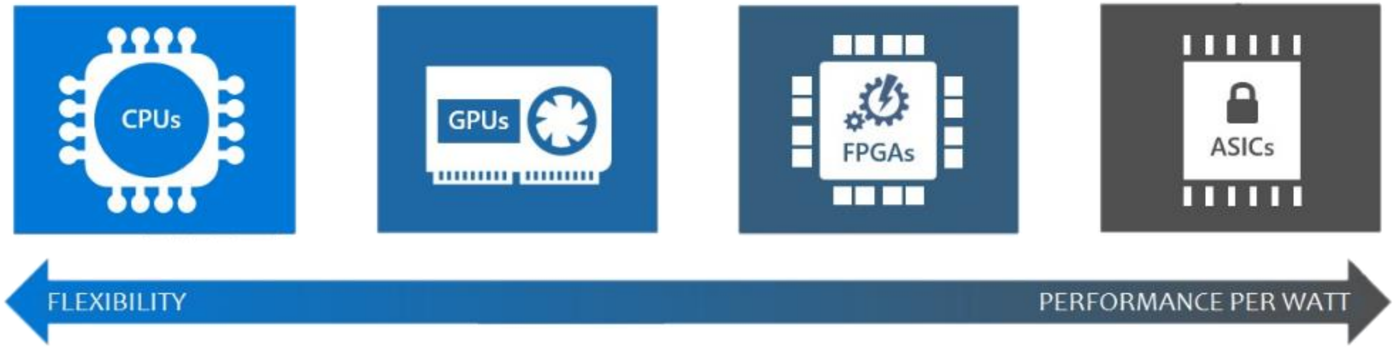
- Bandwidth constraints
- Increased latency
- Unconstrained resources
- Security requirements

ON-BOARD
DATA PROCESSING

- Computation near data source
- Real-time Processing
- Limited resources
- Power constraints

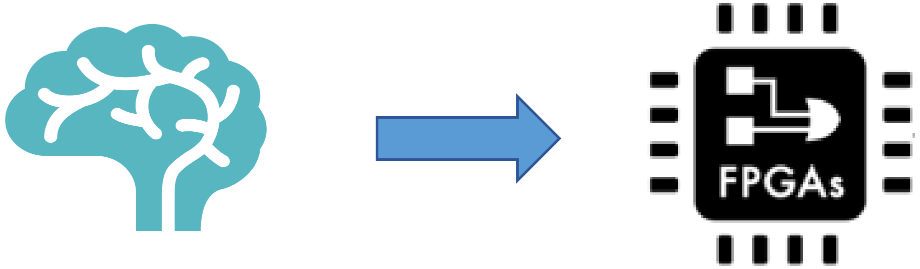
AI Accelerators Onboard Satellites

- AI implementation on-the-edge tricky owing to design trade-offs
- FPGAs offer valuables trade-offs between performance and flexibility
  - More flexible than ASICs
  - Specifity leading to better energy efficiency than CPUs/GPUs
  - Possibility of rad-hard technology
  - Limited resources budget
  - High design effort and long time-to-market



Research Objective

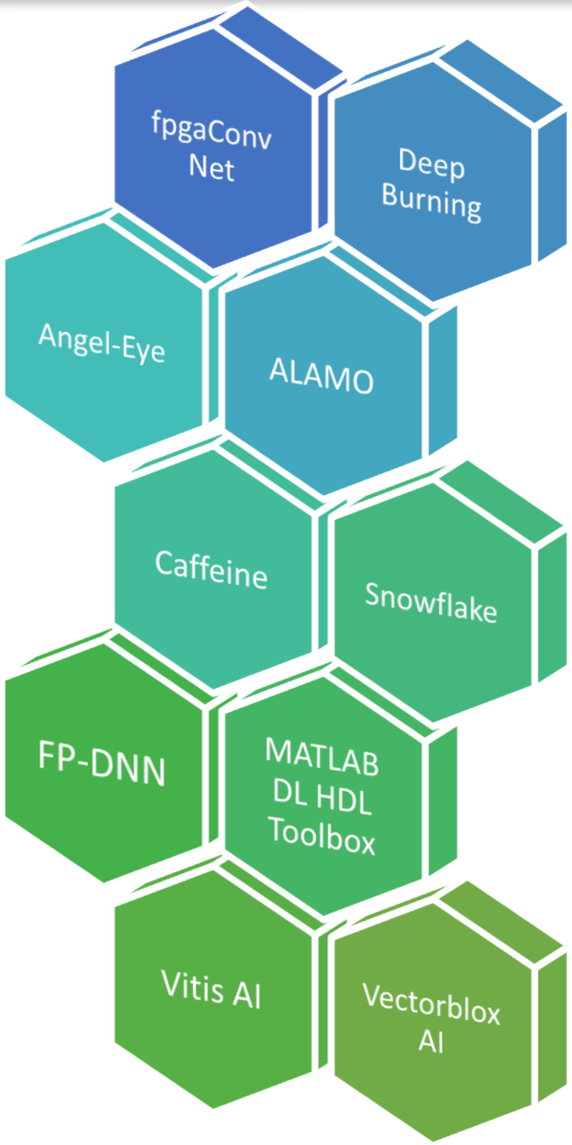
To create **automation frameworks** to enable a wide range of users without specific expertise to accelerate Deep Neural Networks (DNNs) on FPGAs with reduced development times



Presentation Outline

- Research Context
- DNN-to-FPGA Toolflows
- FPG-AI: an Automation Toolflow for CNNs
- Conclusions

State of the Art of Automatic Toolflows for DNNs Acceleration on FPGA



fpgaConvNet

Deep Burning

Angel-Eye

ALAMO

Caffeine

Snowflake

FP-DNN

MATLAB DL HDL Toolbox

Vitis AI

Vectorblox AI

Metrics for DNN-to-FPGA Toolflows

1. **Interface:**
   Input library, Supported layers
2. **Model Compression:**
   In-training vs Post-training
3. **Hardware Architecture:**
   High-Level Synthesis (HLS) vs Register-Transfer Logic (RTL)
4. **Device Portability:**
   Standalone FPGAs, SoCs
5. **Design Space Exploration & Automation:**
   FPGA-dependent vs (CNN+FPGA)-dependent

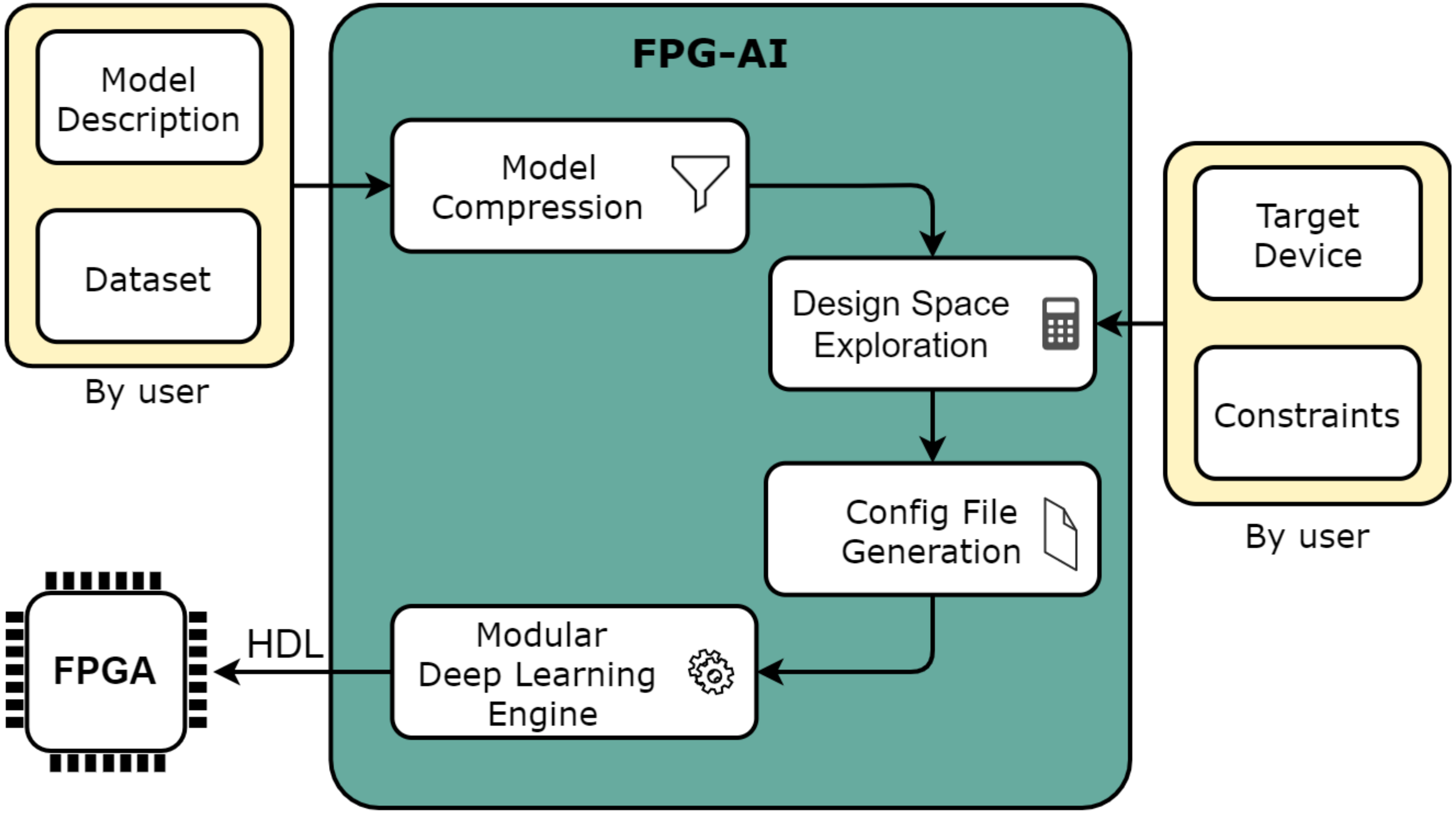
Beyond the state of the art

- Fully handcrafted RTL architecture for **vendor/technology-independent design**
- Hardware accelerators with **no CPU** processing
- Improved Design Space Exploration
  - **Detailed analytical model** for precise timing/resources estimations before synthesis
  - Very **fine-grained quantization** analysis
  - New constraints (accuracy, resource):
    enhanced **user control** over the final design

Presentation Outline

- Research Context
- DNN-to-FPGA Toolflows
- FPG-AI: an Automation Toolflow for CNNs
- Conclusions

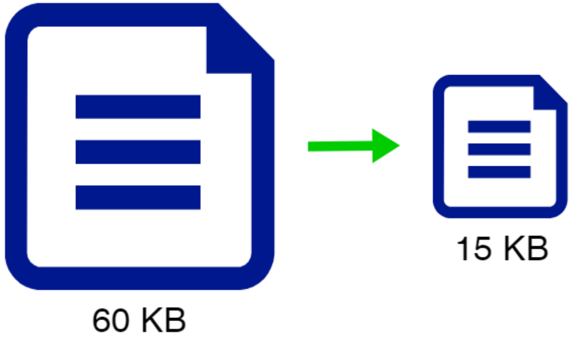
FPG-AI: an Automation Toolflow for CNNs



T. Pacini, E. Rapuano, L. Fanucci: "FPG-AI: a Technology-Independent Framework for the Automation of CNN Deployment on FPGAs", submitted to IEEE Access Journal, March 2023

FPG-AI: Model Compression

- **Post-training** approach: no need for expertise in CNN model design
- **Dynamic quantization** applied on input and weights
- **Truncation** at layers’ output for further memory footrpint reduction
- **Layer folding**: Batch Normalization, Average Pooling
- **Fully fixed-point** quantization for hardware efficiency (timing/area/power) with respect to floating-point machines



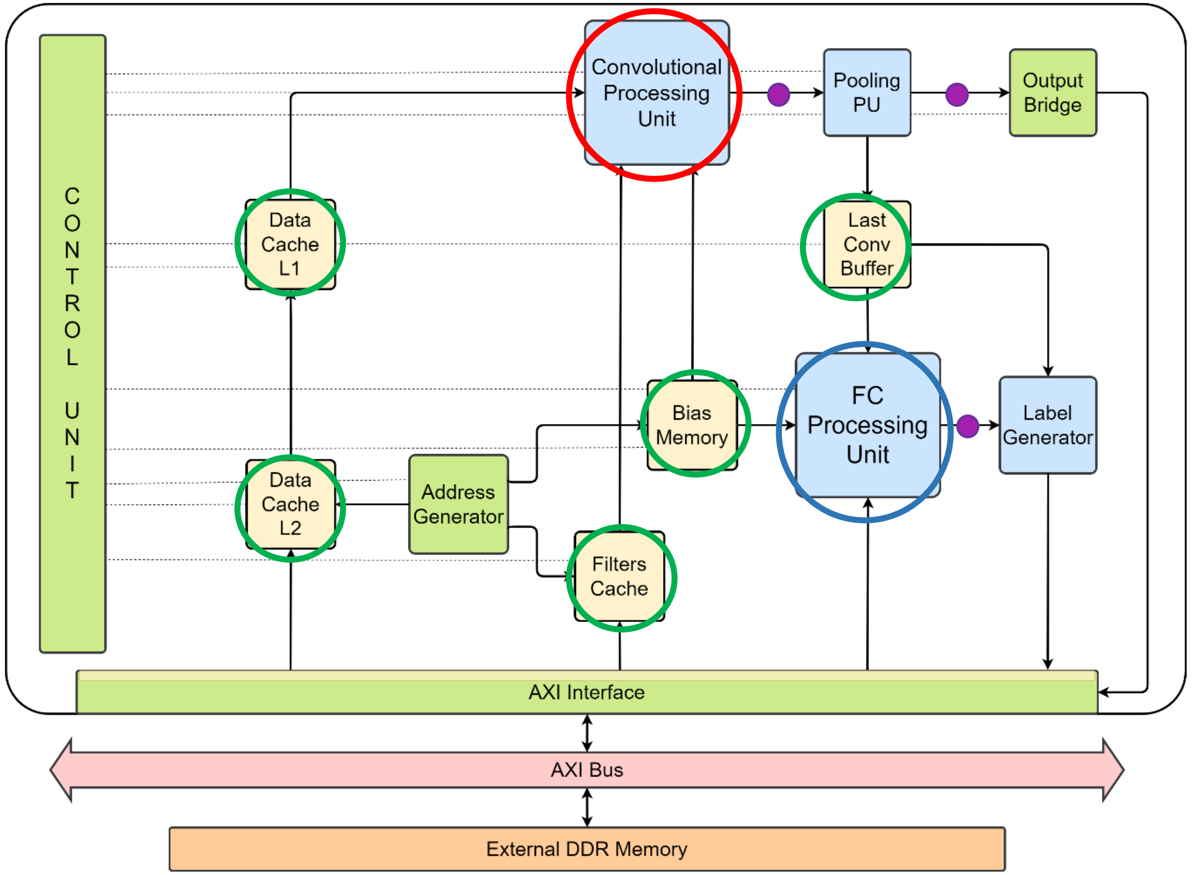
Modular Deep Learning Engine (MDE)

- **High portability**: no third-party IPs used
- **High scalability** in terms of DSP/On-chip Memory utilization
- **Easily configurable** through a file (.vhd)
  - **Model parameters:**

    Input shape, # Layers, Layers type, # Classes, etc.
  - **Quantization parameters:**

    Input/weights bitwidths, Truncation and Saturation bits
  - **Architectural parameters:**

    # MAC units (N_{PE}), Parallel channels (PCh_{in}), $N_{neurons}$, Memory primitives for each IP, etc.



Design Space Exploration (DSE)

- Detailed analytical model of the HW architecture
- Choice of the configurable parameters through iterations
- DSE depending on:
  - CNN model
  - Target FPGA
  - User’s constraints (optional)

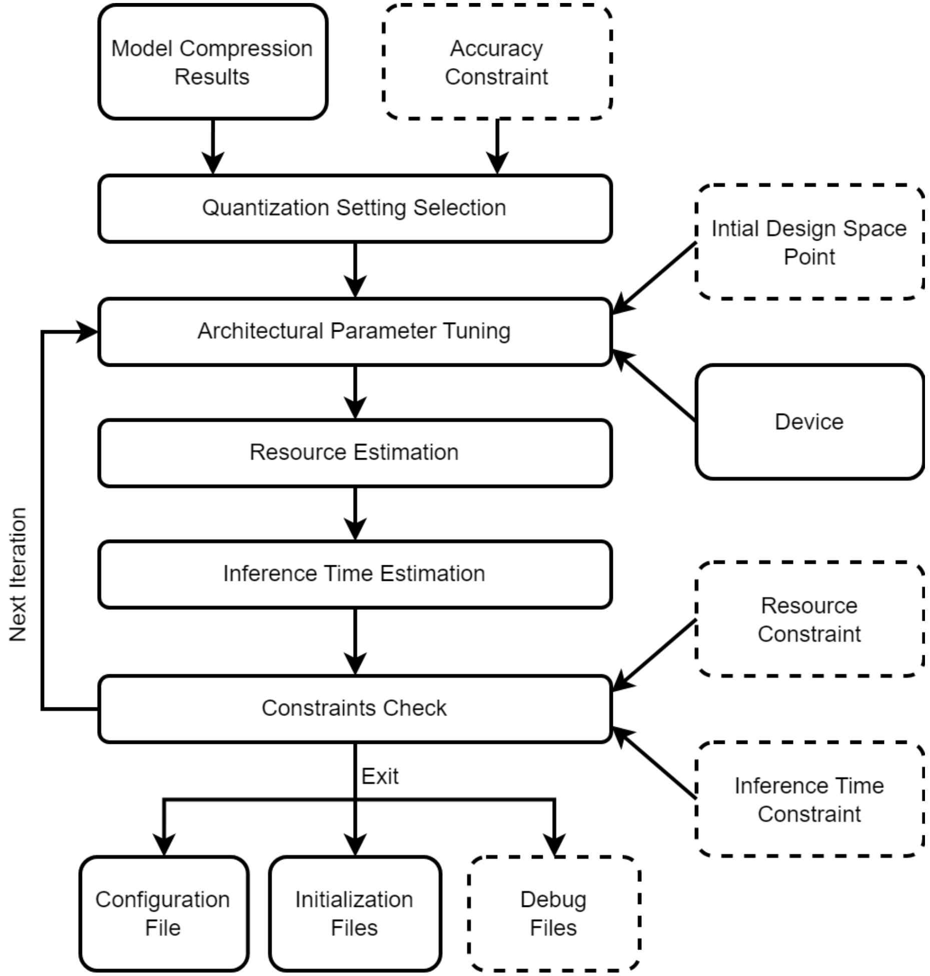
Initial values

Minimum accuracy

Maximum inference time

Resources limitation

- Outputs:
  - HW configuration file
  - Memories initialization files
  - Textual debug files (optional)



Model Compression Results

Accuracy Constraint

Quantization Setting Selection

Intial Design Space Point

Architectural Parameter Tuning

Device

Resource Estimation

Inference Time Estimation

Next Iteration

Resource Constraint

Constraints Check

Inference Time Constraint

Exit

Configuration File

Initialization Files

Debug Files

Implementation Results

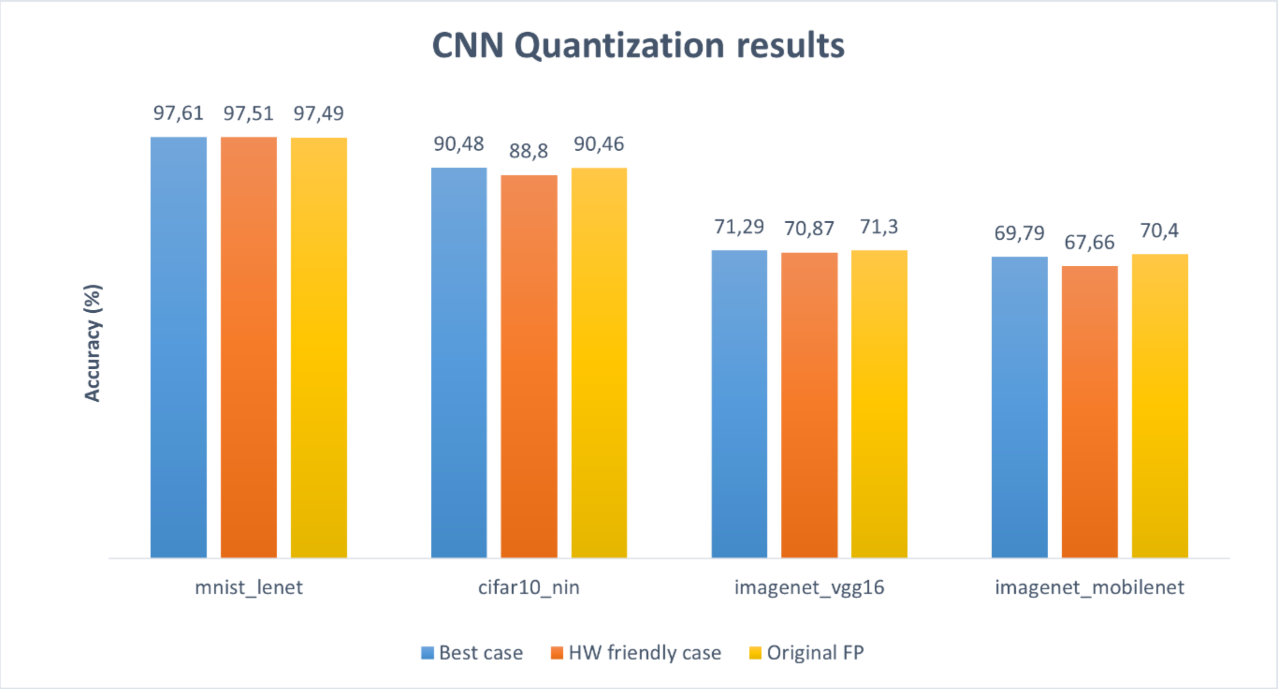
- Heterogeneous set of FPGA families to demonstrate portability (rad-hard FPGAs considered):
  - **Xilinx** Zynq US+, Kintex US, Zynq-7000
  - **Microsemi** PolarFire, RTG4
  - **Intel** Arria 10, Stratix V
- Tool tested on popular CNN models and datasets:

Model	Dataset	Input shape	#Classes
LeNet	MNIST	28x28x1	10
CloudScout	Sentinel-2	512x512x3	2
NiN	CIFAR10	32x32x3	10
VGG16	ImageNet	224x224x3	1000
MobileNet	ImageNet	224x224x3	1000

Model Compression Results

- Model Compression results as a list of cases
- Quantization/truncation settings evaluated:
  - **Best case**: highest accuracy
  - **HW-friendly case**: resource efficiency

- General conclusions:
  - Up to **75% memory footprint reduction**
  - Comparable **(-2%) or even higher accuracy** with respect to original floating-point model



Toolflows Implementation Results: Comparison

- Default solutions by FPG-AI: maximum degree of parallelism
- Comparison between same CNN-FPGA pairs where possible

Model	LeNet			NiN		VGG16			MobileNet-V1	
Toolflow	fpgaConvNet	DeepBurning	**FPG-AI**	DeepBurning	**FPG-AI**	Ad-hoc acc.	Caffeine	**FPG-AI**	Ad-hoc acc.	**FPG-AI**
Device	XC7Z020	XC7Z045	XC7Z045	XC7Z045	XC7Z045	KU060	KU060	KU060	XC7Z045	XC7Z045
Precision	16 FXP	N/A	≤8b FXP	N/A	≤8b FXP	16b FXP	16b FXP	≤12b FXP	16b FXP	≤10b FXP
DSP	5	12	434	42	611	2123	1058	2338	608	586
RAM [Mb]	0.33	0.22	1.75	3.25	10.16	27.63	26.79	22.57	36.91	13.92
Frequency [MHz]	100	100	100	100	80	263	200	66.2	250	80
Inference Time [ms]	1.08	6.7	0.09	54.4	11.04	42.68	110.7	520.8	17.98	156.1
Timing Eff. [GOP/s]	0.48	0.08	5.94	6.79	33.46	689.6	266	29.87	63.4	7.30
RAM Eff. [GOP/s/Mb]	1.46	0.35	3.39	2.09	3.29	24.96	9.93	1.32	1.72	0.52
DSP Eff. [GOP/s/DSP]	0.096	0.006	0.014	0.162	0.055	0.325	0.251	0.013	0.104	0.012
Accuracy [%]	95.4	97.3	97.5	79.5	88.9	N/A	N/A	70.9	N/A	67.7

Toolflows Implementation Results: Comparison

- Default solutions by FPG-AI: maximum degree of parallelism
- Comparison between same CNN-FPGA pairs where possible

Model	LeNet			NiN		VGG16			MobileNet-V1	
Toolflow	fpgaConvNet	DeepBurning	**FPG-AI**	DeepBurning	**FPG-AI**	Ad-hoc acc.	Caffeine	**FPG-AI**	Ad-hoc acc.	**FPG-AI**
Device	XC7Z020	XC7Z045	XC7Z045	XC7Z045	XC7Z045	KU060	KU060	KU060	XC7Z045	XC7Z045
Precision	16 FXP	N/A	≤8b FXP	N/A	≤8b FXP	16b FXP	16b FXP	≤12b FXP	16b FXP	≤10b FXP
DSP	5	12	434	42	611	2123	1058	2338	608	586
RAM [Mb]	0.33	0.22	1.75	3.25	10.16	27.63	26.79	22.57	36.91	13.92
Frequency [MHz]	100	100	100	100	80	263	200	66.2	250	80
Inference Time [ms]	1.08	6.7	0.09	54.4	11.04	42.68	110.7	520.8	17.98	156.1
Timing Eff. [GOP/s]	0.48	0.08	5.94	6.79	33.46	689.6	266	29.87	63.4	7.30
RAM Eff. [GOP/s/Mb]	1.46	0.35	3.39	2.09	3.29	24.96	9.93	1.32	1.72	0.52
DSP Eff. [GOP/s/DSP]	0.096	0.006	0.014	0.162	0.055	0.325	0.251	0.013	0.104	0.012
Accuracy [%]	95.4	97.3	97.5	79.5	88.9	N/A	N/A	70.9	N/A	67.7

Presentation Outline

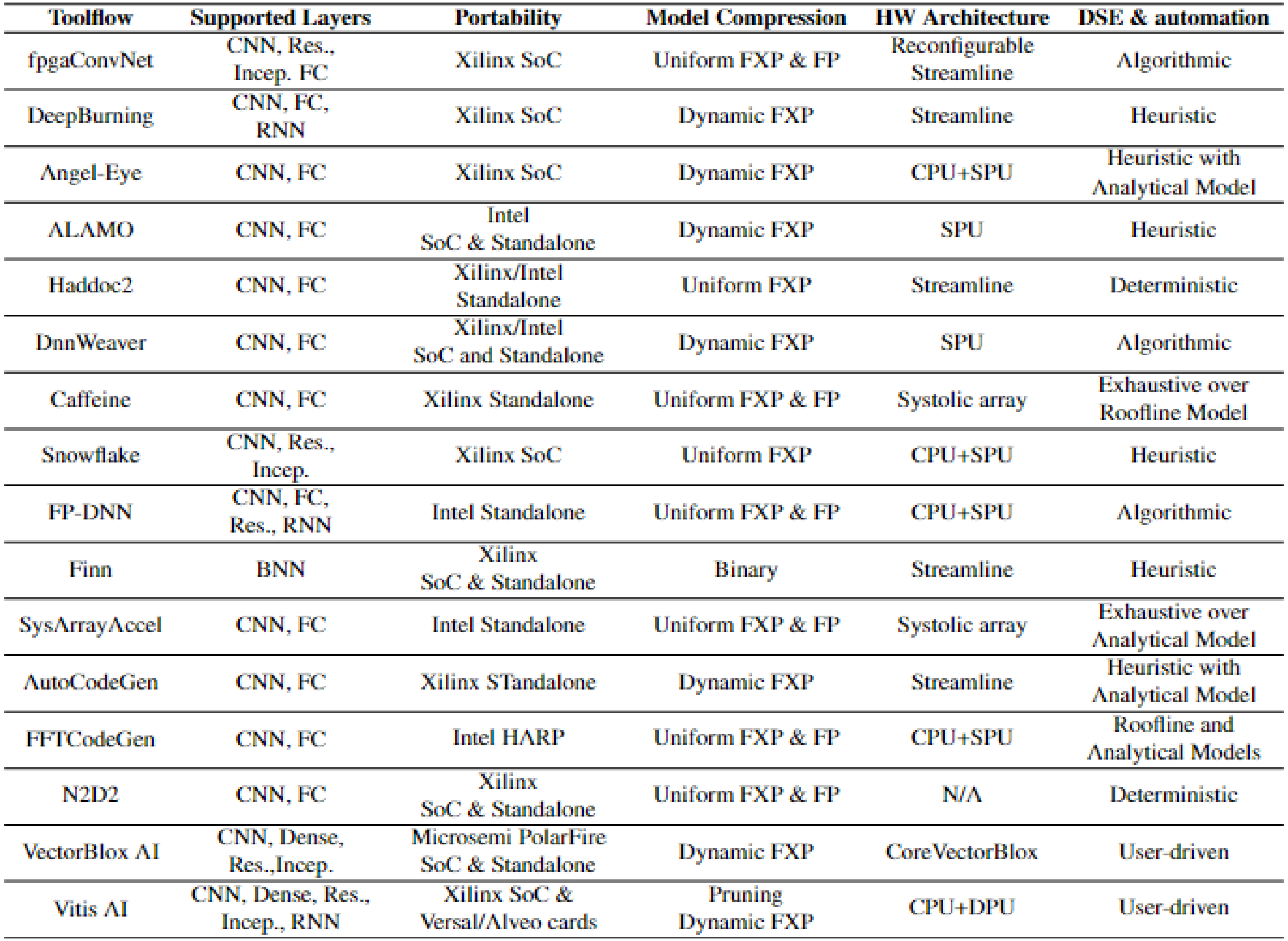
- Research Context
- DNN-to-FPGA Toolflows
- FPG-AI: an Automation Toolflow for CNNs
- Conclusions

FPG-AI: Conclusions

- Very detailed and layer-wise quantization analysis
- Optimized RTL architecture with **no CPU processing**
- Finer Design Space Exploration
  - **Accurate estimations** with analytical model
  - **Enhanced user control** over the final design
  - **New constraints**: accuracy, resources limitations
- **Enhanced portability**: any target FPGA device
- On-going developments:
  - Extension to Recurrent Neural Networks (RNNs)
  - Extension to NanoXplore technology

Thank you for
the attention

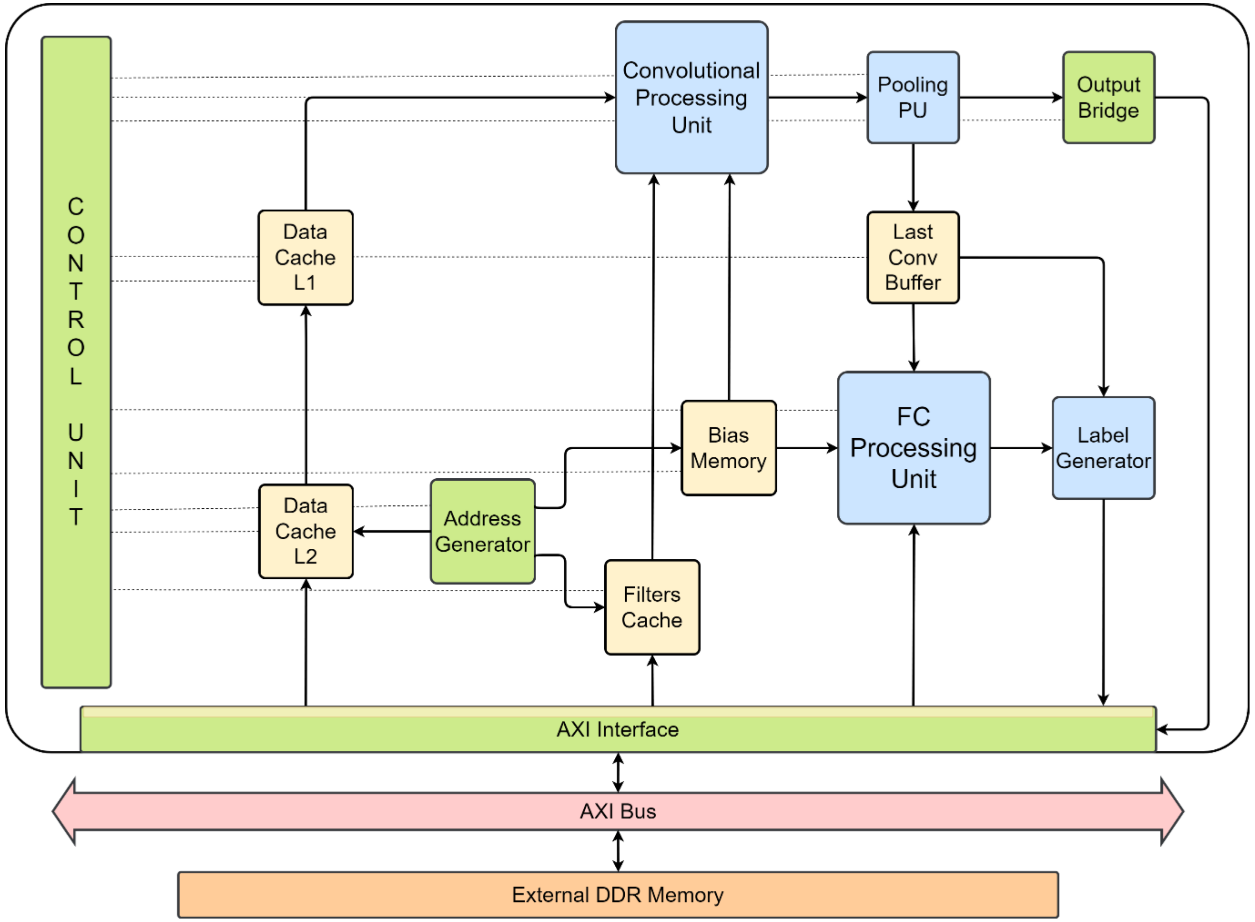
Backup slides



Toolflow	Supported Layers	Portability	Model Compression	HW Architecture	DSE & automation
fpgaConvNet	CNN, Res., Incep. FC	Xilinx SoC	Uniform FXP & FP	Reconfigurable Streamline	Algorithmic
DeepBurning	CNN, FC, RNN	Xilinx SoC	Dynamic FXP	Streamline	Heuristic
Angel-Eye	CNN, FC	Xilinx SoC	Dynamic FXP	CPU+SPU	Heuristic with Analytical Model
ALAMO	CNN, FC	Intel SoC & Standalone	Dynamic FXP	SPU	Heuristic
Haddoc2	CNN, FC	Xilinx/Intel Standalone	Uniform FXP	Streamline	Deterministic
DnnWeaver	CNN, FC	Xilinx/Intel SoC and Standalone	Dynamic FXP	SPU	Algorithmic
Caffeine	CNN, FC	Xilinx Standalone	Uniform FXP & FP	Systolic array	Exhaustive over Roofline Model
Snowflake	CNN, Res., Incep.	Xilinx SoC	Uniform FXP	CPU+SPU	Heuristic
FP-DNN	CNN, FC, Res., RNN	Intel Standalone	Uniform FXP & FP	CPU+SPU	Algorithmic
Finn	BNN	Xilinx SoC & Standalone	Binary	Streamline	Heuristic
SysArrayAccel	CNN, FC	Intel Standalone	Uniform FXP & FP	Systolic array	Exhaustive over Analytical Model
AutoCodeGen	CNN, FC	Xilinx STandalone	Dynamic FXP	Streamline	Heuristic with Analytical Model
FFTCodeGen	CNN, FC	Intel HARP	Uniform FXP & FP	CPU+SPU	Roofline and Analytical Models
N2D2	CNN, FC	Xilinx SoC & Standalone	Uniform FXP & FP	N/A	Deterministic
VectorBlox AI	CNN, Dense, Res.,Incep.	Microsemi PolarFire SoC & Standalone	Dynamic FXP	CoreVectorBlox	User-driven
Vitis AI	CNN, Dense, Res., Incep., RNN	Xilinx SoC & Versal/Alveo cards	Pruning Dynamic FXP	CPU+DPU	User-driven

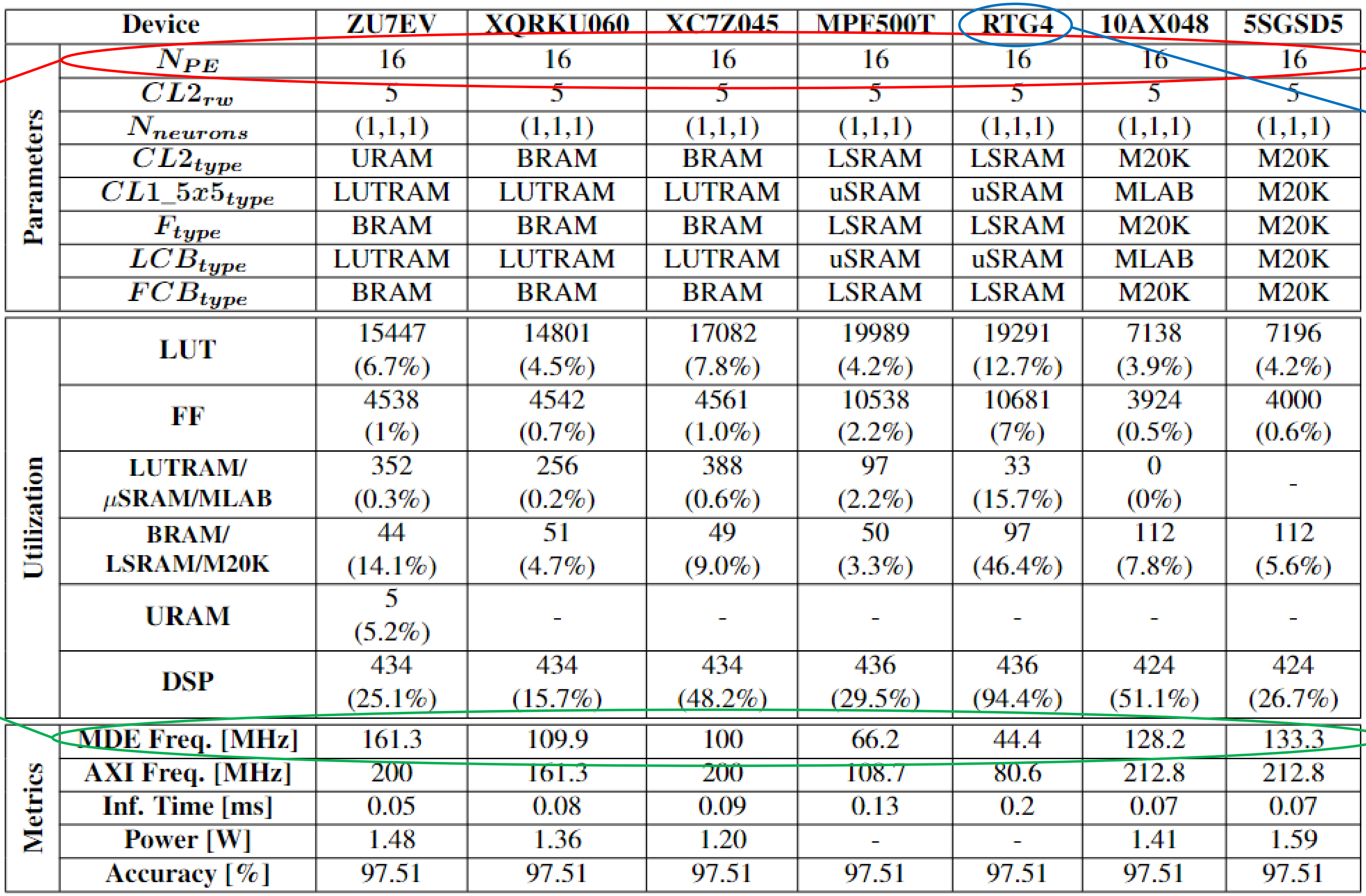
FPG-AI Architecture: Modular Deep-Learning Engine (MDE)

- **Multiple Processing Units** time-shared among layers
- **Use of external memory** (AXI-interfaced) for data and weights
- **Custom Cache systems** for efficient resource usage
- **Optimal scheduling strategies** for inference time reduction



Implementation Results - LeNet

No variation on N_{PE}: maximum parallelization always possible

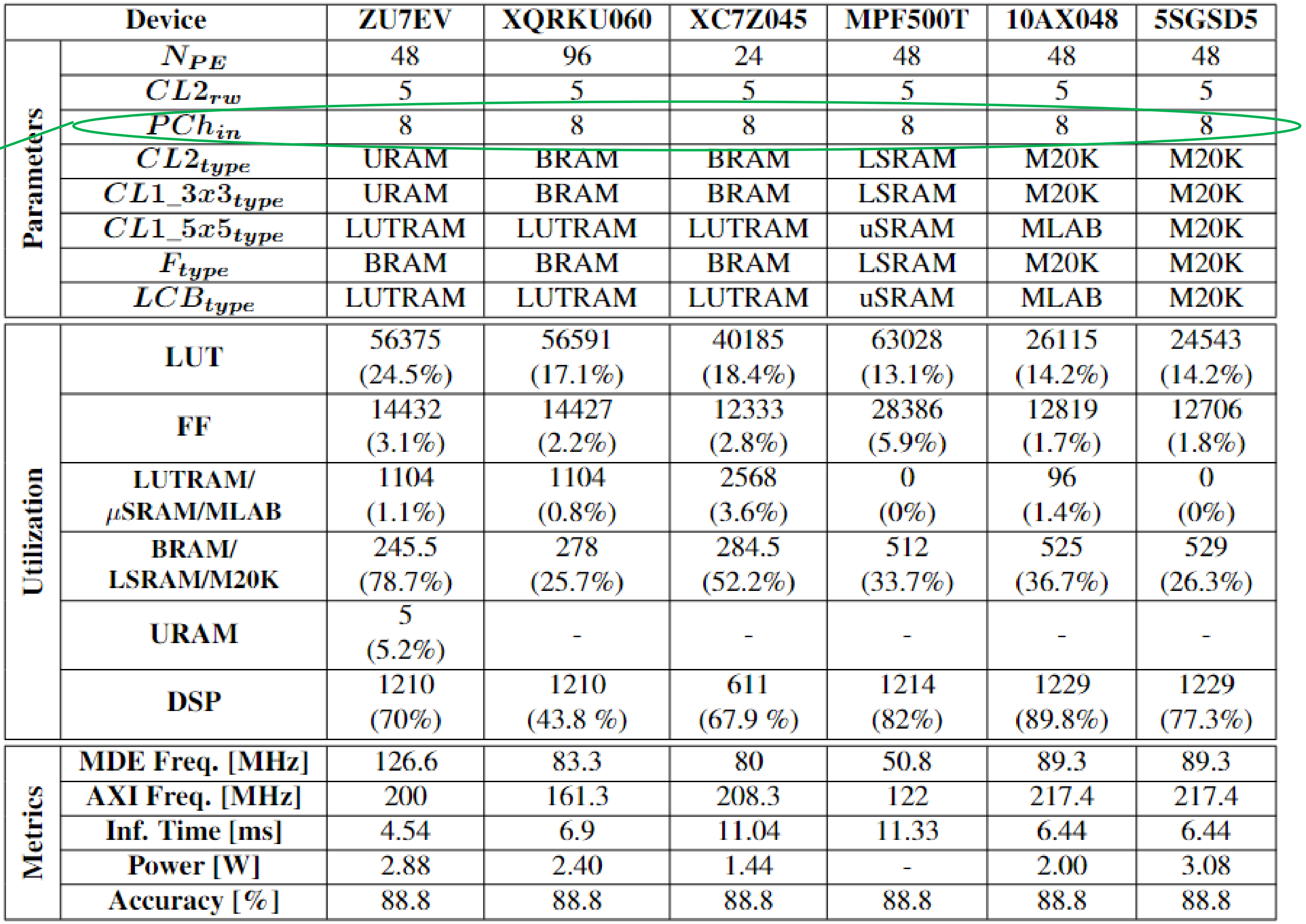


	Device	ZU7EV	XQRKU060	XC7Z045	MPF500T	RTG4	10AX048	5SGSD5
Parameters	N_{PE}	16	16	16	16	16	16	16
	$CL2_{rw}$	5	5	5	5	5	5	5
	$N_{neurons}$	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
	$CL2_{type}$	URAM	BRAM	BRAM	LSRAM	LSRAM	M20K	M20K
	$CL1\_5x5_{type}$	LUTRAM	LUTRAM	LUTRAM	uSRAM	uSRAM	MLAB	M20K
	F_{type}	BRAM	BRAM	BRAM	LSRAM	LSRAM	M20K	M20K
	LCB_{type}	LUTRAM	LUTRAM	LUTRAM	uSRAM	uSRAM	MLAB	M20K
	FCB_{type}	BRAM	BRAM	BRAM	LSRAM	LSRAM	M20K	M20K
Utilization	LUT	15447 (6.7%)	14801 (4.5%)	17082 (7.8%)	19989 (4.2%)	19291 (12.7%)	7138 (3.9%)	7196 (4.2%)
	FF	4538 (1%)	4542 (0.7%)	4561 (1.0%)	10538 (2.2%)	10681 (7%)	3924 (0.5%)	4000 (0.6%)
	LUTRAM/ μSRAM/MLAB	352 (0.3%)	256 (0.2%)	388 (0.6%)	97 (2.2%)	33 (15.7%)	0 (0%)	-
	BRAM/ LSRAM/M20K	44 (14.1%)	51 (4.7%)	49 (9.0%)	50 (3.3%)	97 (46.4%)	112 (7.8%)	112 (5.6%)
	URAM	5 (5.2%)	-	-	-	-	-	-
	DSP	434 (25.1%)	434 (15.7%)	434 (48.2%)	436 (29.5%)	436 (94.4%)	424 (51.1%)	424 (26.7%)
Metrics	MDE Freq. [MHz]	161.3	109.9	100	66.2	44.4	128.2	133.3
	AXI Freq. [MHz]	200	161.3	200	108.7	80.6	212.8	212.8
	Inf. Time [ms]	0.05	0.08	0.09	0.13	0.2	0.07	0.07
	Power [W]	1.48	1.36	1.20	-	-	1.41	1.59
	Accuracy [%]	97.51	97.51	97.51	97.51	97.51	97.51	97.51

Unique model small enough to be implemented on RTG4

Highest frequencies reached

Implementation Results - NiN



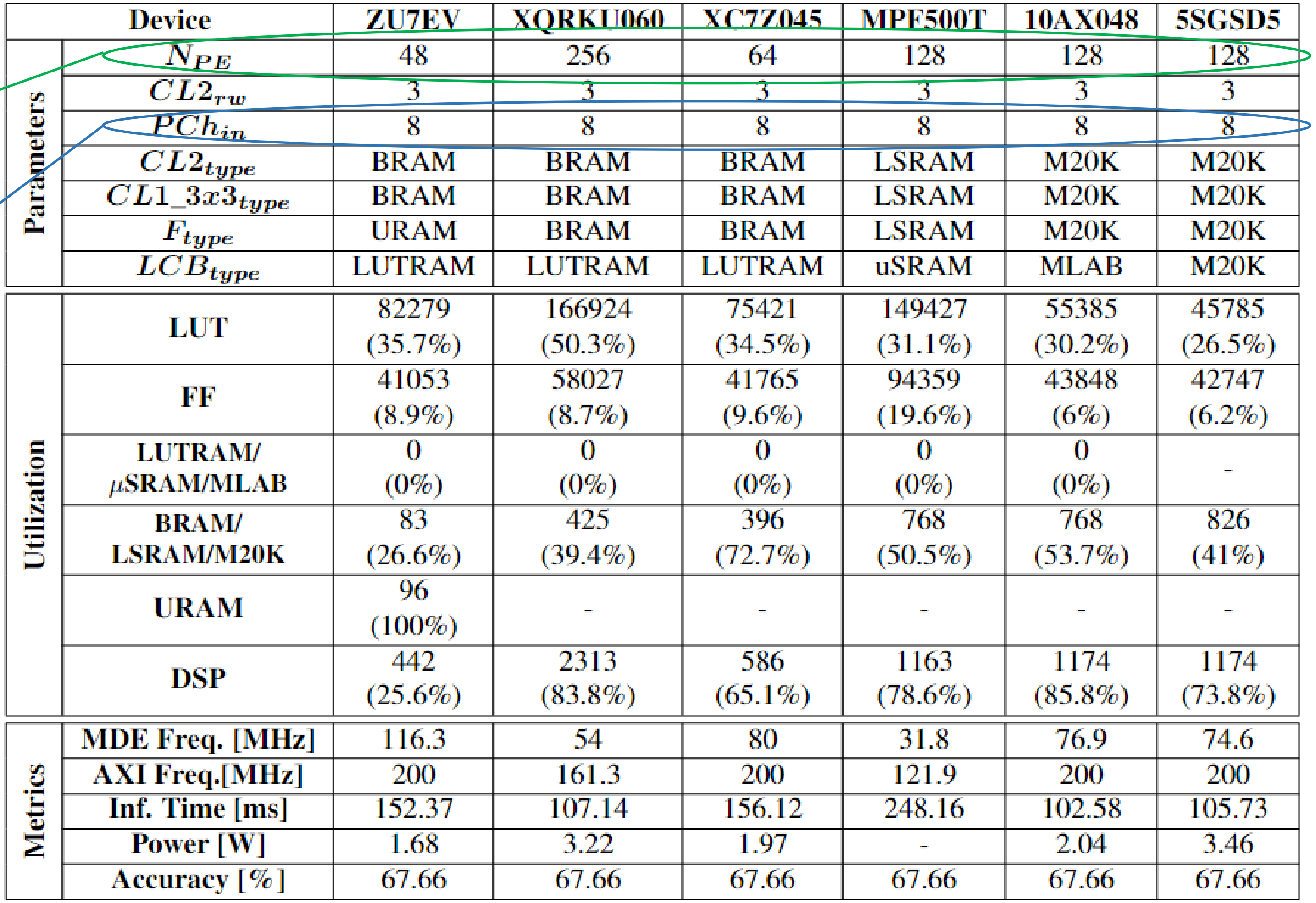
	Device	ZU7EV	XQRKU060	XC7Z045	MPF500T	10AX048	5SGSD5
Parameters	N_{PE}	48	96	24	48	48	48
	$CL2_{rw}$	5	5	5	5	5	5
	PCh_{in}	8	8	8	8	8	8
	$CL2_{type}$	URAM	BRAM	BRAM	LSRAM	M20K	M20K
	$CL1\_3x3_{type}$	URAM	BRAM	BRAM	LSRAM	M20K	M20K
	$CL1\_5x5_{type}$	LUTRAM	LUTRAM	LUTRAM	uSRAM	MLAB	M20K
	F_{type}	BRAM	BRAM	BRAM	LSRAM	M20K	M20K
	LCB_{type}	LUTRAM	LUTRAM	LUTRAM	uSRAM	MLAB	M20K
Utilization	LUT	56375 (24.5%)	56591 (17.1%)	40185 (18.4%)	63028 (13.1%)	26115 (14.2%)	24543 (14.2%)
	FF	14432 (3.1%)	14427 (2.2%)	12333 (2.8%)	28386 (5.9%)	12819 (1.7%)	12706 (1.8%)
	LUTRAM/ μSRAM/MLAB	1104 (1.1%)	1104 (0.8%)	2568 (3.6%)	0 (0%)	96 (1.4%)	0 (0%)
	BRAM/ LSRAM/M20K	245.5 (78.7%)	278 (25.7%)	284.5 (52.2%)	512 (33.7%)	525 (36.7%)	529 (26.3%)
	URAM	5 (5.2%)	-	-	-	-	-
	DSP	1210 (70%)	1210 (43.8 %)	611 (67.9 %)	1214 (82%)	1229 (89.8%)	1229 (77.3%)
Metrics	MDE Freq. [MHz]	126.6	83.3	80	50.8	89.3	89.3
	AXI Freq. [MHz]	200	161.3	208.3	122	217.4	217.4
	Inf. Time [ms]	4.54	6.9	11.04	11.33	6.44	6.44
	Power [W]	2.88	2.40	1.44	-	2.00	3.08
	Accuracy [%]	88.8	88.8	88.8	88.8	88.8	88.8

Advantage for
1x1 layers

Implementation Results - MobileNet

Widest N_{PE} variation

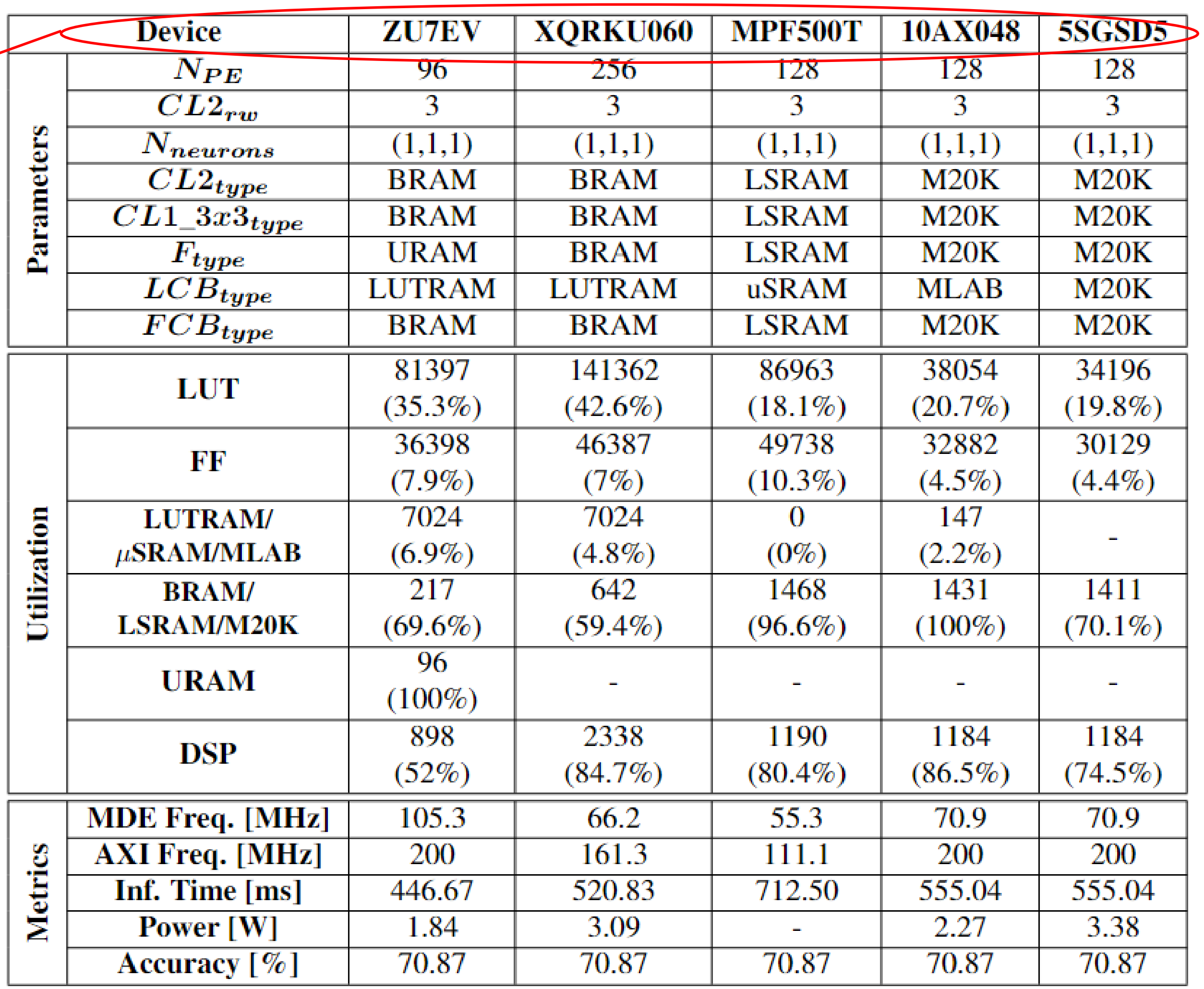
Advantage for 1x1 layers



	Device	ZU7EV	XQRKU060	XC7Z045	MPF500T	10AX048	5SGSD5
Parameters	N_{PE}	48	256	64	128	128	128
	$CL2_{rw}$	3	3	3	3	3	3
	PCh_{in}	8	8	8	8	8	8
	$CL2_{type}$	BRAM	BRAM	BRAM	LSRAM	M20K	M20K
	$CL1\_3x3_{type}$	BRAM	BRAM	BRAM	LSRAM	M20K	M20K
	F_{type}	URAM	BRAM	BRAM	LSRAM	M20K	M20K
	LCB_{type}	LUTRAM	LUTRAM	LUTRAM	uSRAM	MLAB	M20K
Utilization	**LUT**	82279 (35.7%)	166924 (50.3%)	75421 (34.5%)	149427 (31.1%)	55385 (30.2%)	45785 (26.5%)
	FF	41053 (8.9%)	58027 (8.7%)	41765 (9.6%)	94359 (19.6%)	43848 (6%)	42747 (6.2%)
	LUTRAM/ μSRAM/MLAB	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	-
	BRAM/ LSRAM/M20K	83 (26.6%)	425 (39.4%)	396 (72.7%)	768 (50.5%)	768 (53.7%)	826 (41%)
	URAM	96 (100%)	-	-	-	-	-
	DSP	442 (25.6%)	2313 (83.8%)	586 (65.1%)	1163 (78.6%)	1174 (85.8%)	1174 (73.8%)
Metrics	**MDE Freq. [MHz]**	116.3	54	80	31.8	76.9	74.6
	AXI Freq.[MHz]	200	161.3	200	121.9	200	200
	Inf. Time [ms]	152.37	107.14	156.12	248.16	102.58	105.73
	Power [W]	1.68	3.22	1.97	-	2.04	3.46
	Accuracy [%]	67.66	67.66	67.66	67.66	67.66	67.66

Implementation Results – VGG16

Hardest model to fit FPGA resources



	Device	ZU7EV	XQRKU060	MPF500T	10AX048	5SGSD5
Parameters	N_{PE}	96	256	128	128	128
	$CL2_{rw}$	3	3	3	3	3
	$N_{neurons}$	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
	$CL2_{type}$	BRAM	BRAM	LSRAM	M20K	M20K
	$CL1\_3x3_{type}$	BRAM	BRAM	LSRAM	M20K	M20K
	F_{type}	URAM	BRAM	LSRAM	M20K	M20K
	LCB_{type}	LUTRAM	LUTRAM	uSRAM	MLAB	M20K
	FCB_{type}	BRAM	BRAM	LSRAM	M20K	M20K
Utilization	**LUT**	81397 (35.3%)	141362 (42.6%)	86963 (18.1%)	38054 (20.7%)	34196 (19.8%)
	FF	36398 (7.9%)	46387 (7%)	49738 (10.3%)	32882 (4.5%)	30129 (4.4%)
	LUTRAM/ µSRAM/MLAB	7024 (6.9%)	7024 (4.8%)	0 (0%)	147 (2.2%)	-
	BRAM/ LSRAM/M20K	217 (69.6%)	642 (59.4%)	1468 (96.6%)	1431 (100%)	1411 (70.1%)
	URAM	96 (100%)	-	-	-	-
	DSP	898 (52%)	2338 (84.7%)	1190 (80.4%)	1184 (86.5%)	1184 (74.5%)
Metrics	**MDE Freq. [MHz]**	105.3	66.2	55.3	70.9	70.9
	AXI Freq. [MHz]	200	161.3	111.1	200	200
	Inf. Time [ms]	446.67	520.83	712.50	555.04	555.04
	Power [W]	1.84	3.09	-	2.27	3.38
	Accuracy [%]	70.87	70.87	70.87	70.87	70.87