Bernhard Vogginger, Florian Kelber, Shambhavi Balamuthu Sampath, Johannes Partzsch, Christian Mayr
Chair of Highly-Parallel VLSI Systems and Neuro-Microelectronics

Performance models and energy-optimal scheduling of DNNs on many-core hardware with dynamic power management

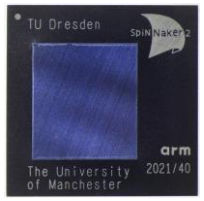
CODAI 2022 Workshop

Sponsored by:

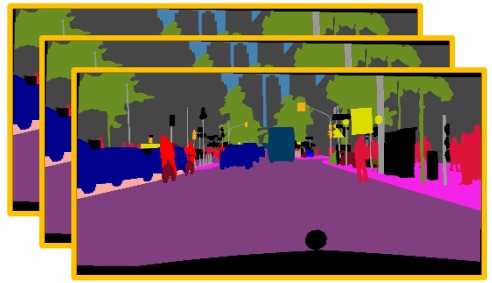
Introduction

Challenge:

Real-time analysis of input (e.g. images at 30 fps) on embedded DNN chip with limited power/energy budget



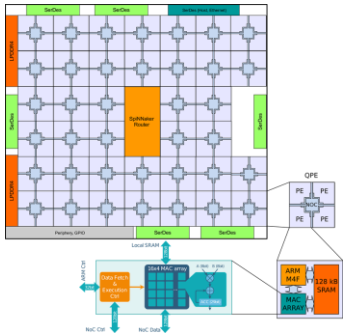
Many-core hardware with
Dynamic Voltage and Frequency Scaling (DVFS)



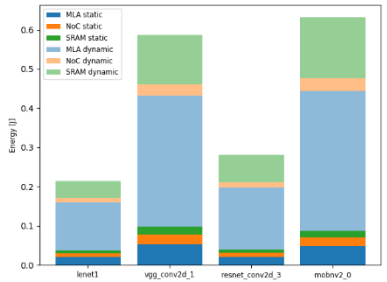
Examples from Cityscapes Dataset [Cordts 2016]

Outline of the presentation

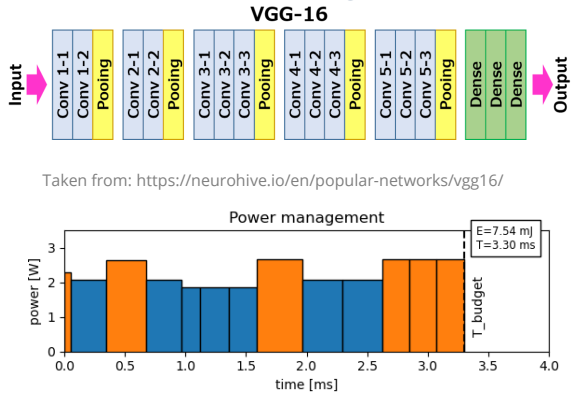
Hardware Description



Performance Models



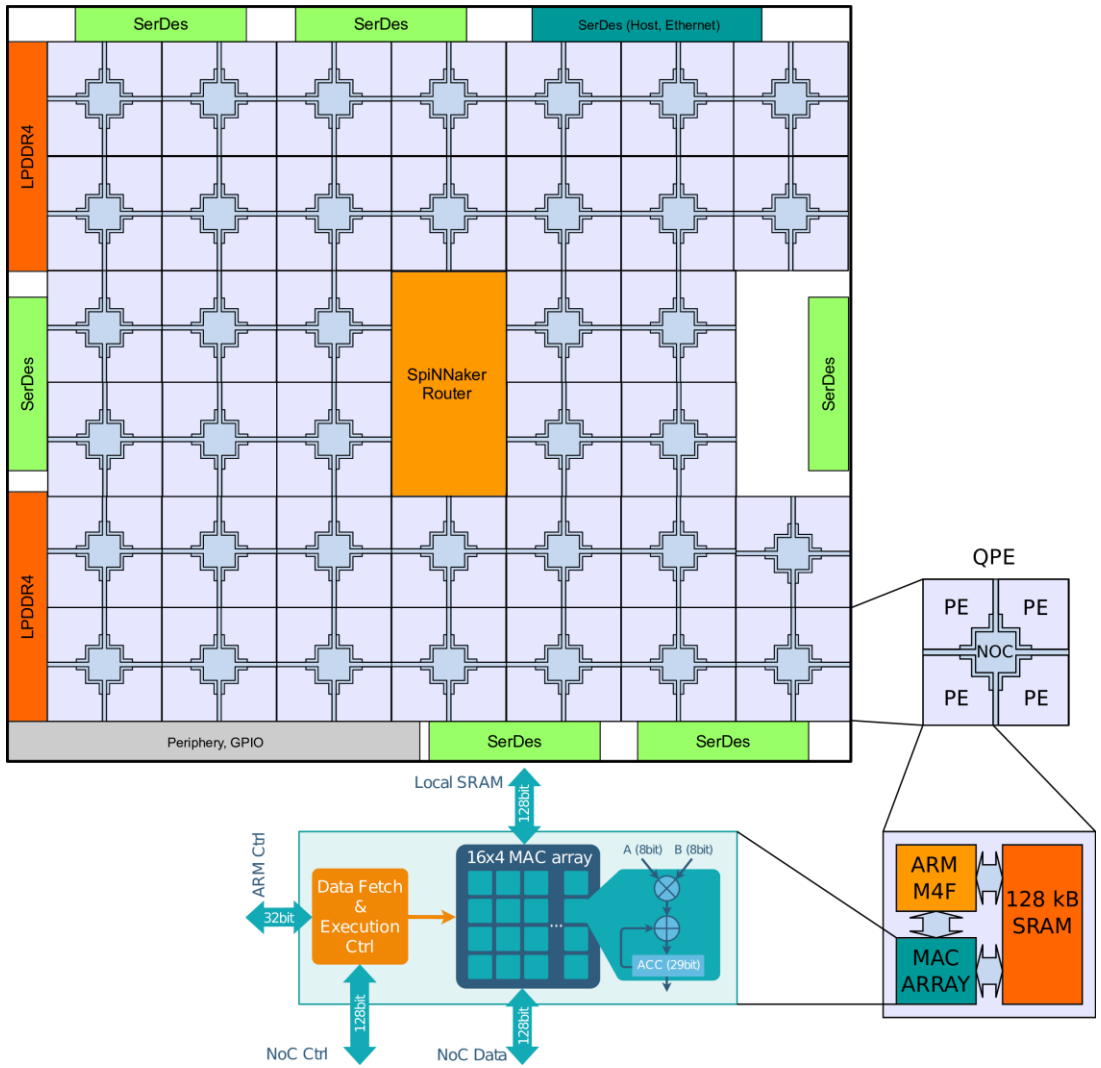
Power management



Taken from: https://neurohive.io/en/popular-networks/vgg16/

SpiNNaker2 many-core hardware

The SpiNNaker2 Chip Architecture



- **Advancement from SpiNNaker 1 chip:** "Spiking Neural Network Architecture" by University of Manchester (Prof. Steve Furber)
- **152 processing elements**:
  - Arm Cortex M4F core for flexible processing
  - 128 KB SRAM
  - **64 unit MAC array** for DNN inference
- 2GByte LPDDR4 per chip
- Network-on-Chip (NoC) for on-chip comm.
- SpiNNaker router & chip-to-chip links for packet-based communication
- **22FDX** CMOS GLOBALFOUNDRIES (Tape-out 6/2021)

The SpiNNaker2 Processing Element (Jib1 prototype version)

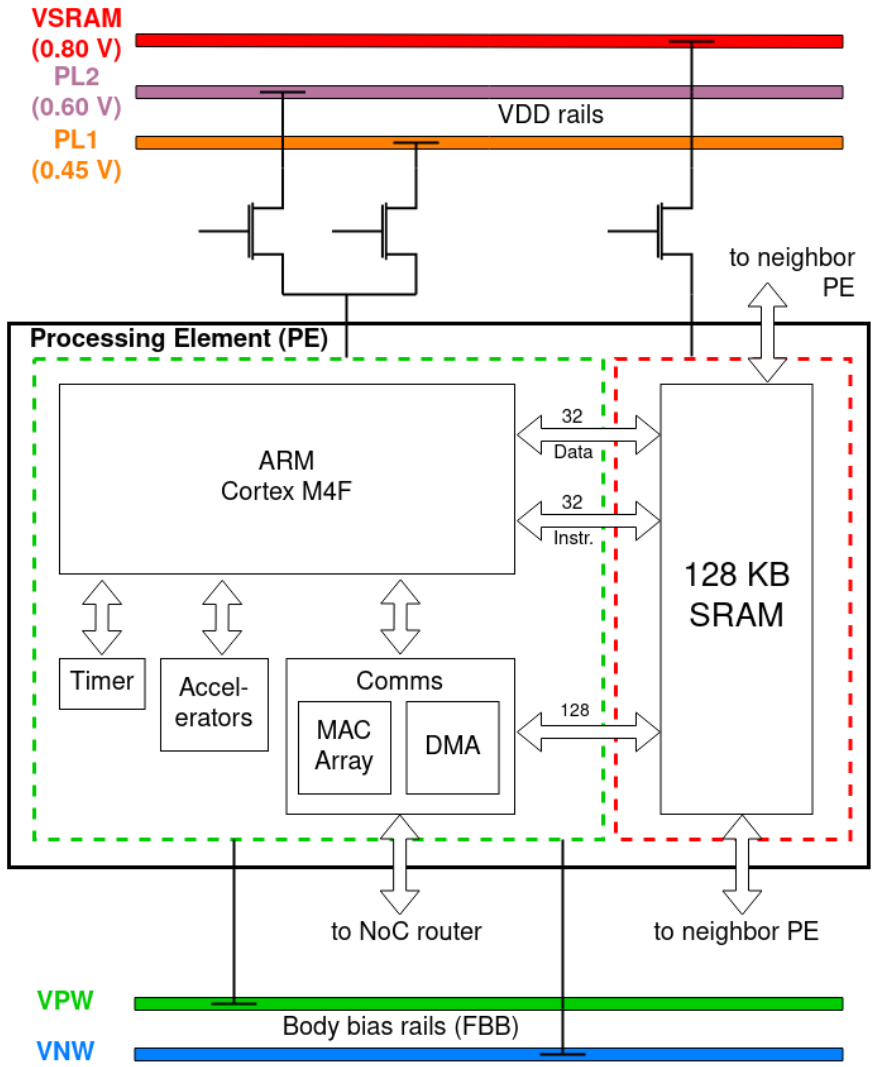
ABB (adaptive body-biasing) IP by Racyics allows operation of core down to 0.45 V

Power management (DVFS):

- Select **from 2 global supply lanes** for core logic: typical range: 0.45 - 0.6 V
- Select **frequency** per PE typical range: 50 - 400 MHz
- Switch performance level (supply lane + f_{clk}) within tens of nanoseconds [Hoeppner 2019]

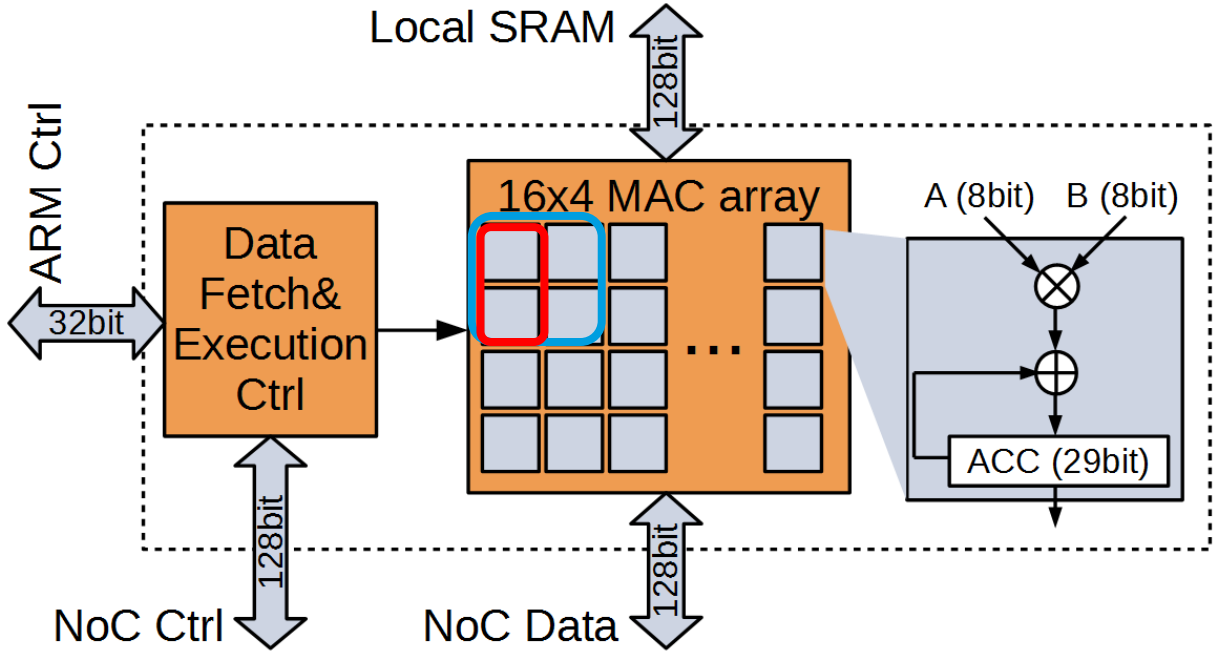
Other power domains are fixed

- Network-on-Chip: 0.6 V, 300 MHz (typical)
- SRAM: 0.8 V, accessed with PE clock



Integrated MAC Accelerator

- 16x4 MAC array per core
- operation modes:
  - matrix multiplication
  - 2D convolution
- ReLU and quantization to 8, 16 or 32 bit
- Combine 2 or 4 MAC units for 8x16 bit or 16bx16b integer operations:
  - Allows to trade-off resolution and throughput (performance)



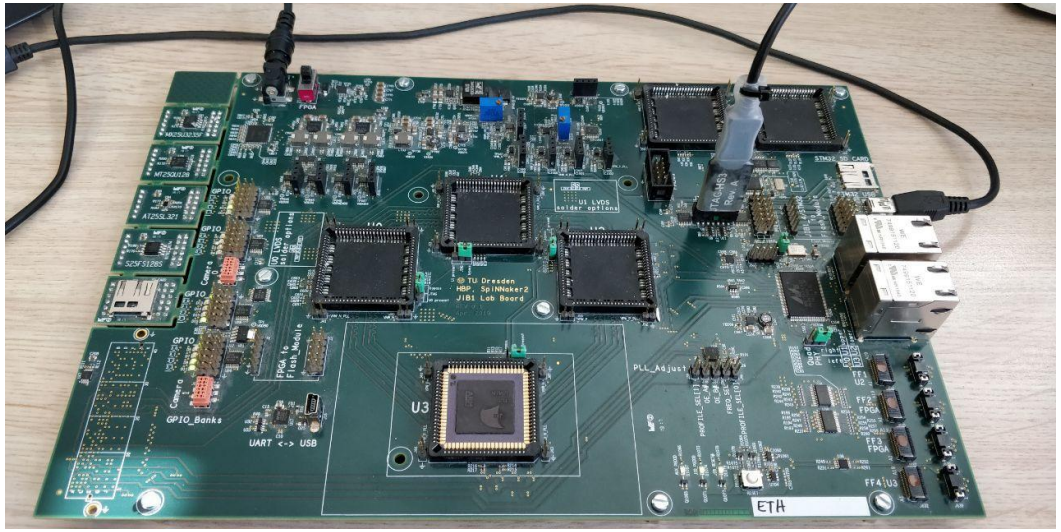
- Only 7 % of silicon area of PE
- Makes a SpiNNaker2 a competetive DNN inference chip:
  - 4.5 TOPS max throughput
  - 2.1 TOPS/W max energy efficiency

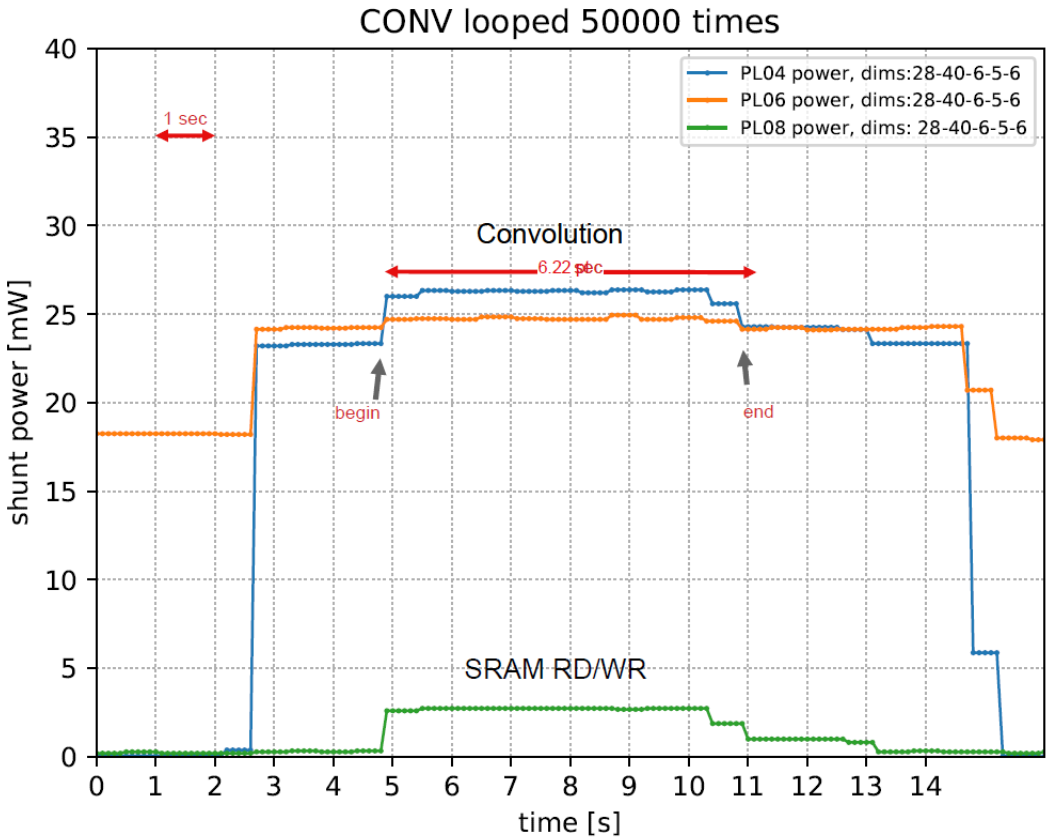
Performance Models

Time and energy model of SpiNNaker2 Machine Learning Accelerator

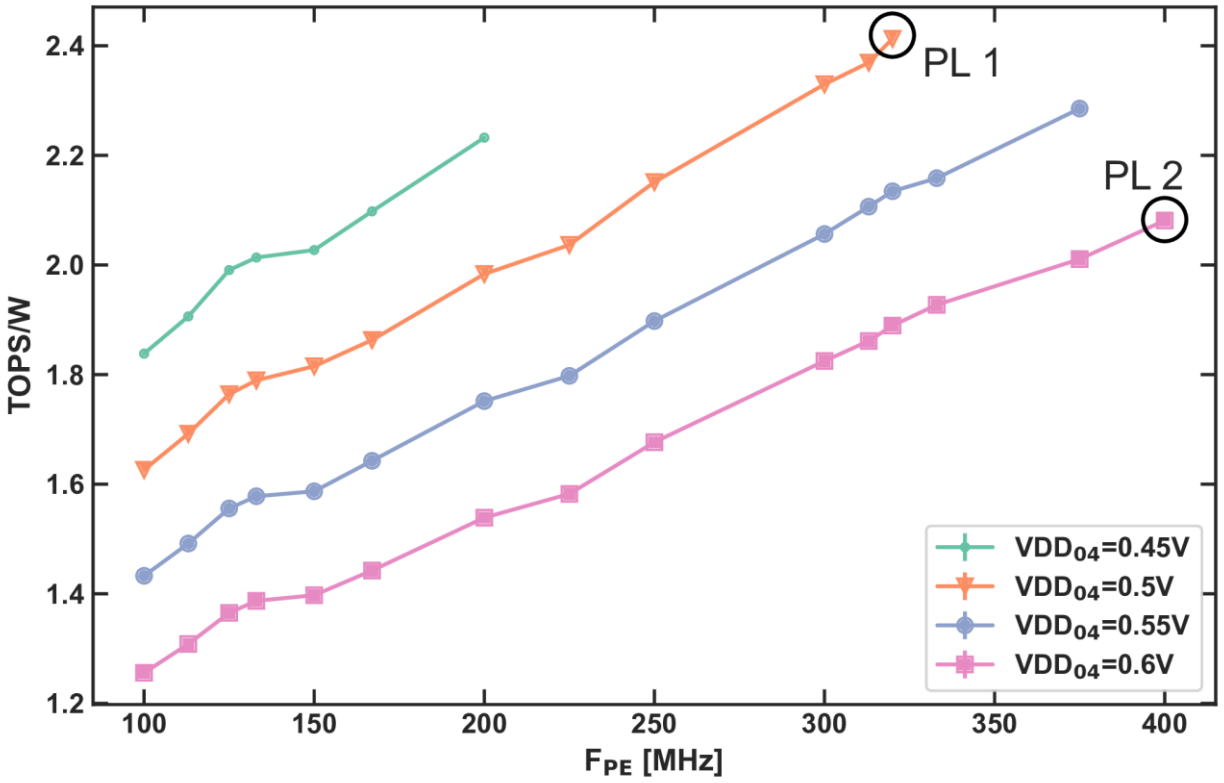
Measurement on prototype chip

- Prototype chip with 8 PEs (2 QPEs)
- Same Conv2D layer running in the loop on all 8 PEs using the machine learning accelerator
- Measure power for all supply lanes over time





Measurements: Sweep of voltage and frequency



Two performance levels selected for further studies

PL 1: VDD_{04}=0.5 V, f_{clk} =320 MHz for least energy

PL 2: VDD_{04}=0.6 V, f_{clk} =400 MHz for highest speed

Time model for Conv2D layer on machine learning accelerator

Modelling approach:

Time depends on Conv2D parameters x_{conv}, fit parameters Θ_conv, and clock frequency f_{clk}.

$$t_{\text{conv}} = t_{\text{conv}}(\boldsymbol{x}_{\text{conv}}, \boldsymbol{\theta}_{\text{conv}}, f_{\text{clk}})$$

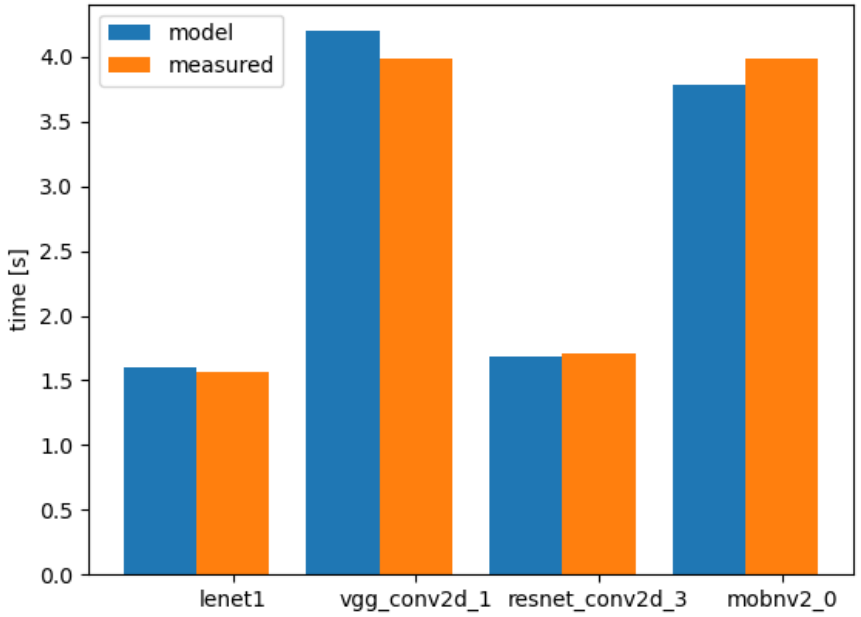
Grey box model for clock cycles based on :

$$\text{clks}_{\text{conv}} = \text{clks}_{\text{init}} + \left\lceil \frac{W_{\text{i}} - W_{\text{w}} + 1}{16} \right\rceil (H_{\text{i}} - H_{\text{w}} + 1) \cdot (aW_{\text{w}}H_{\text{w}}C_{\text{i}} + \text{clks}_{\text{wb}}) \left\lceil \frac{C_{\text{o}}}{4} \right\rceil b \quad ,$$

Calculate actual time considering f_{clk}.

$$t_{\text{conv}} = \frac{\text{clks}_{\text{conv}}}{f_{\text{clk}}}$$

Model vs. Measurement



- f_{clk} =400 MHz
- Total time for 100K loops of Conv2D layer
- Relative time error: **< 9 %**

Energy models

Simple model based on static and dynamic power

$$E_{\text{tot}} = E_{\text{dynamic}} + E_{\text{static}}$$

$$E_{\text{static}} = P_{\text{static}} \cdot t$$

$$E_{\text{dynamic}} = \sum_{actions} E_{\text{action}} \cdot N_{\text{action}}$$

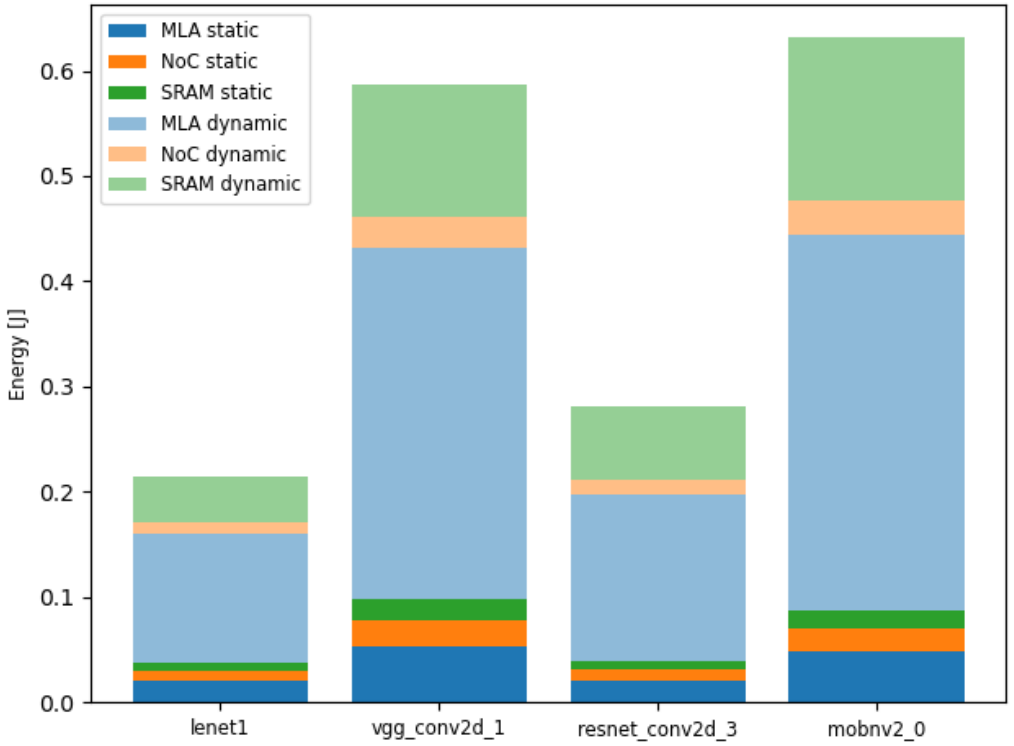
$$E_{\text{dynamic,SRAM}} = E_{\text{read32}} \cdot N_{\text{read32}} + E_{\text{write32}} \cdot N_{\text{write32}}$$

$$E_{\text{dynamic,NoC}} = E_{\text{NoC-read}} \cdot N_{\text{NoC-read}}$$

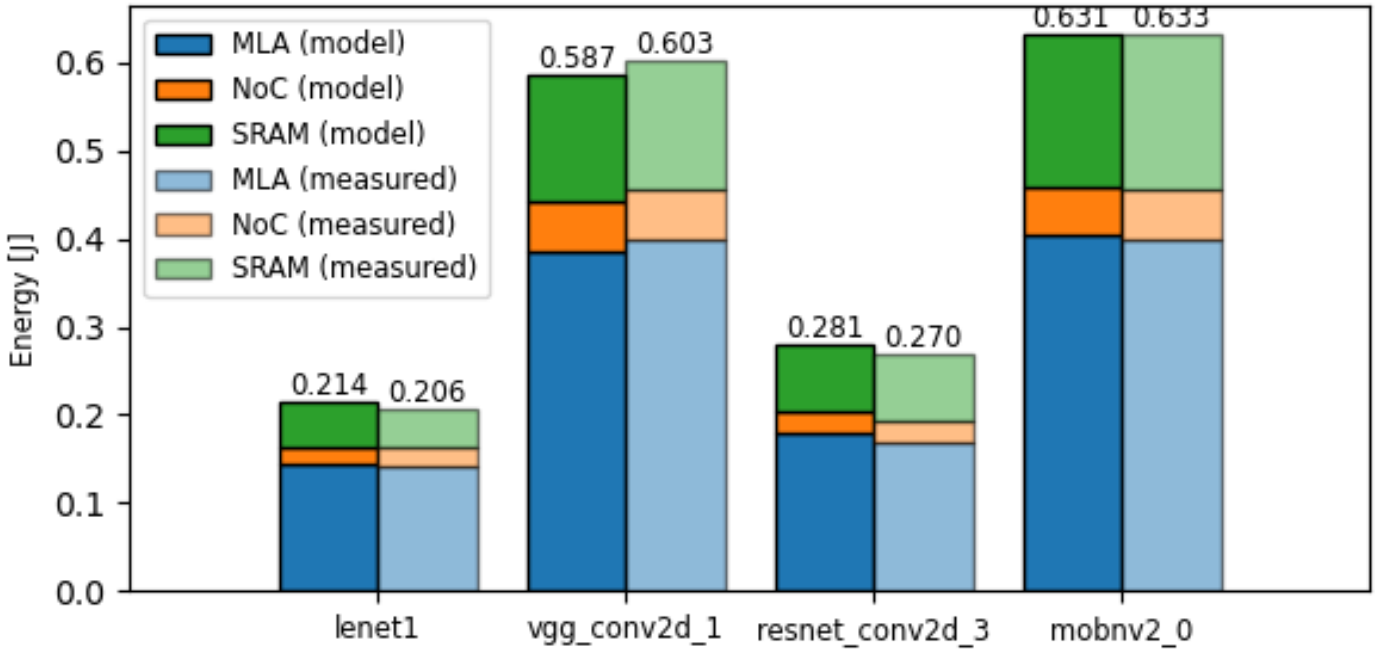
$$E_{\text{dynamic,MLA}} = E_{\text{MLA-compute}} \cdot N_{\text{MLA-compute}}$$

- Action counts calculated based on accelerator operation
- Approach similar to Accelergy [Wu et el. 2019], but with explicit handling of static power

Breakdown of energy contributions (fitted model)



Comparison of energy model and measurements



- **PL 2:** VDD_{04}=0.6 V, f_{clk} =400 MHz
- **Energy** for 100K loops of Conv2D layer run on 8 PEs
- **Relative error** of model vs. Measurement: **< 5 %**

Code Example

```python
from mla_performance_models.models import Conv2DModel
from mla_performance_models.config import Conv2DConfig

# set model parameters
conv2d_model = Conv2DModel(
        clks_init=42,
        clks_wb=16.0,
        a=1.6239,
        b=1.0,
        clk_frequency=400e6,
        static_power_04=1.579e-3,
        static_power_06=0.746e-3,
        static_power_08=0.576e-3,
        E_noc_read_128=20.11e-12,
        E_compute_clock_04=55.17e-12,
        E_sram_read_32=3.94e-12,
        E_sram_write_32=4.34e-12)

# Conv2D parameters
conv2d_cfg = Conv2DConfig()
conv2d_cfg.in_width = 32
conv2d_cfg.in_height = 32
conv2d_cfg.in_channels = 1
conv2d_cfg.filter_width = 5
conv2d_cfg.filter_height = 5
conv2d_cfg.stride_x = 1
conv2d_cfg.stride_y = 1
conv2d_cfg.out_channels = 6

# predict time and energy
time = conv2d_model.time(conv2d_cfg)
energy = conv2d_model.energy(conv2d_cfg)

print("Time [s]:", time)
print("Energy [J]:", energy)
```

Console output:

```
Time [s]: 1.59523e-05
Energy [J]: 2.6800374230000005e-07
```

Power management concepts

for energy-optimal scheduling of DNNs on SpiNNaker2

Overall concept

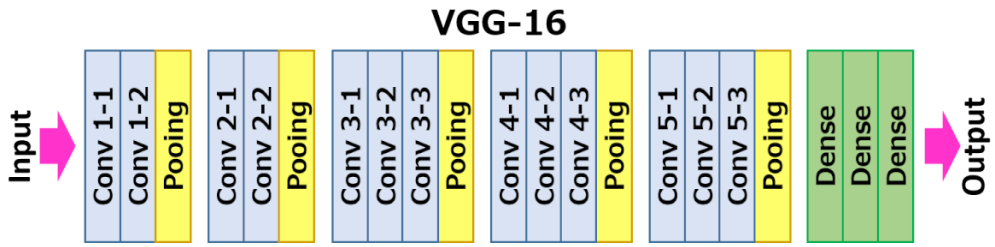
Objective: Energy-optimal mapping of DNNs on SpiNNaker2

Mapping approach:

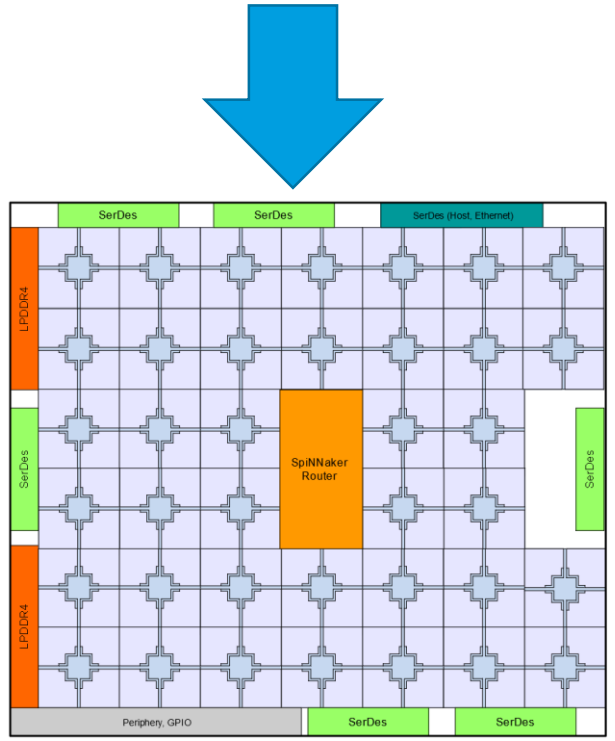
- Layers have to be split into many smaller parts:
  6 MB of largest Conv layer vs. **96 KB** of SRAM per PE for input and output feature maps
- Speedup through data reuse, see [Kelber & Wu et al. 2020]

Dynamic power management:

- Two performance levels:
  - **PL 1 (**0.5 V, 320 MHz) for least energy
  - **PL 2** (0.6 V, 400 MHz) for highest speed
- For each atomic task (small part of layer on MLA), choose PL
- Use performance models to predict time and energy

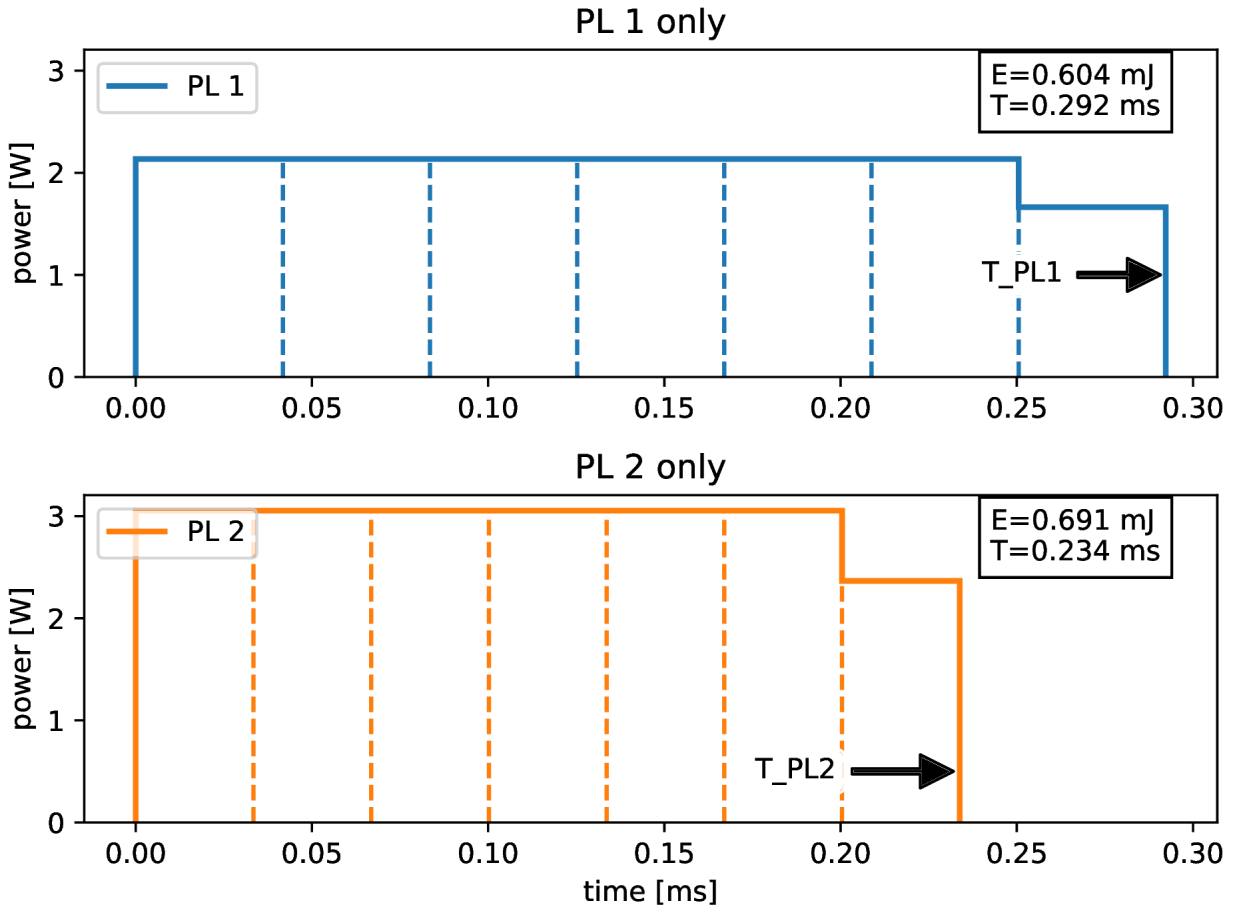


Taken from: https://neurohive.io/en/popular-networks/vgg16/



Example: VGG Conv-1-2 Layer run on SpiNNaker2

- Conv 1-2 Layer has to be split into 1024 parts:
  - 6 loops with 152 parts
  - 1 loop with 112 parts
- Two performance levels:
  - **PL 1 (**0.5 V, 320 MHz) for least energy
  - **PL 2** (0.6 V, 400 MHz) for highest speed
- **Predicted results:**
  - PL 1 requires 11 % less energy
  - PL 2 is 20 % faster



Power management scenarios

Scenarios where switching performance levels is useful:

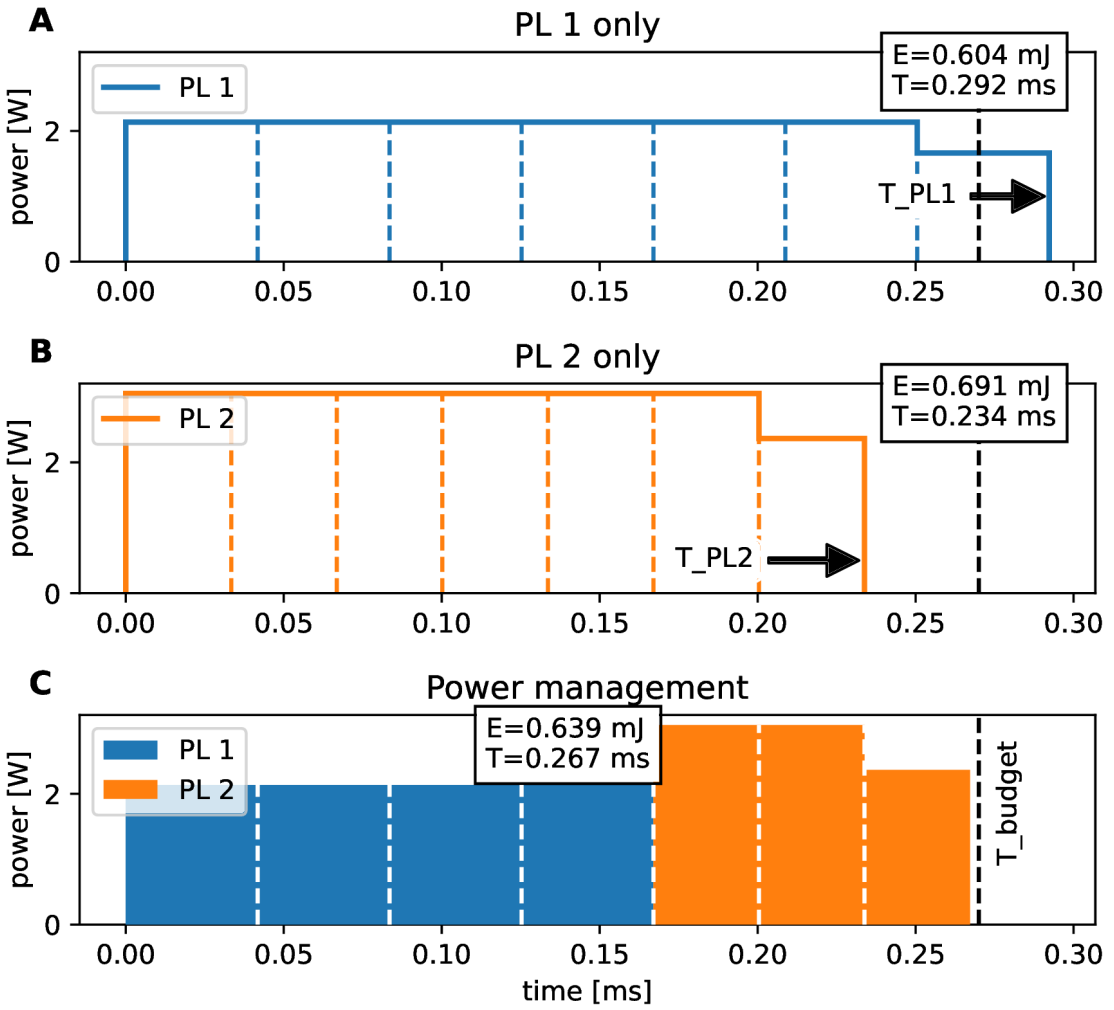
1. **Speed:** DNN should be processed as fast as possible.
2. **Energy:** DNN should be processed with least energy.
3. **Limited time:** Time for processing the DNN is limited, e.g., when images arrive at a certain frame rate.
4. **Limited power:** The power the chip can draw is limited, e.g., by the power supply unit.

Limited Time: single layer (VGG Conv 1-2)

Scenario:

- $T_{PL\,2}$ < Time budget < $T_{PL\,1}$
- Here: time budget is 0.27 ms

- Nr. of loops per PL is automatically tuned such that finish time is just smaller than budget
- Predicted energy saving: 7.4 %

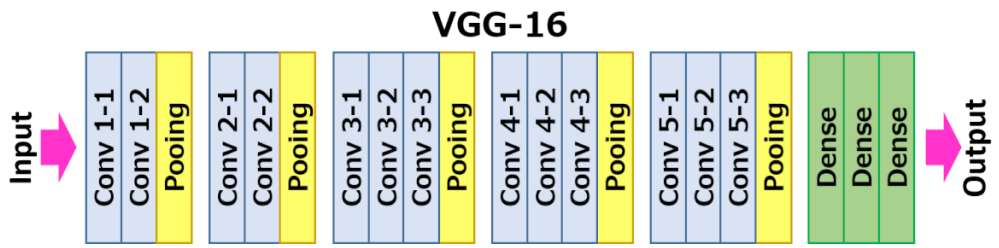


Limited Time: all convolutional layers

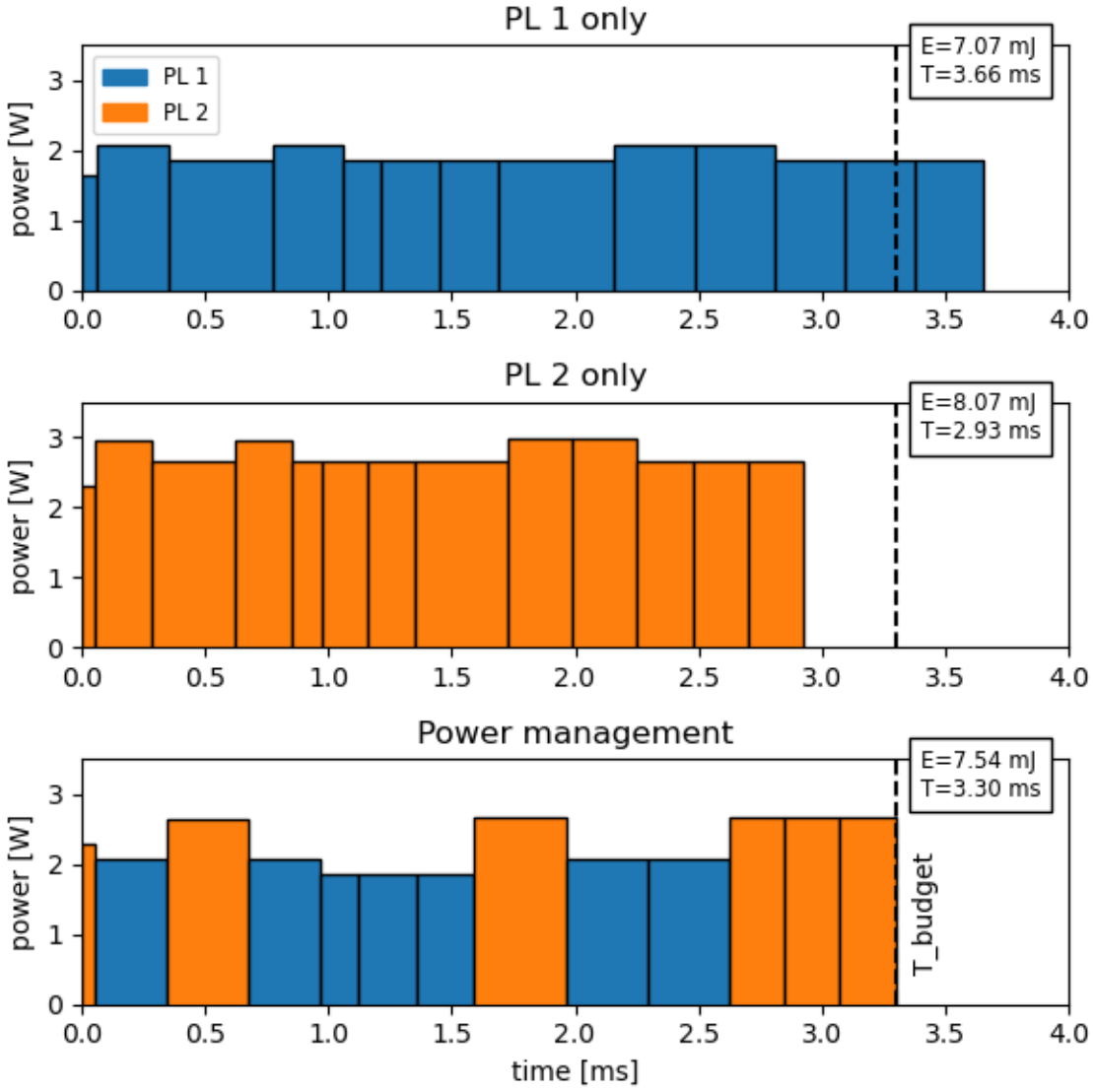
Scenario:

- $T_{PL\,2}$ < Time budget < $T_{PL\,1}$
- Here: time budget is 3.3 ms

- Each layer processed in one PL, automatically tuned to finish just within time budget
- Predicted energy saving: 6.6 %



Taken from: https://neurohive.io/en/popular-networks/vgg16/

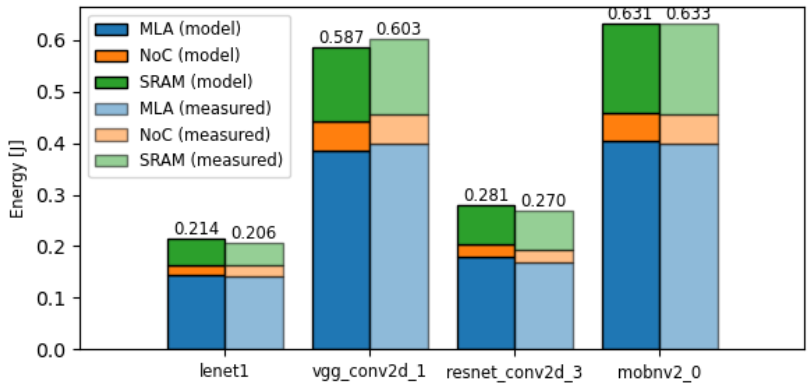


Summary and Outlook

Summary

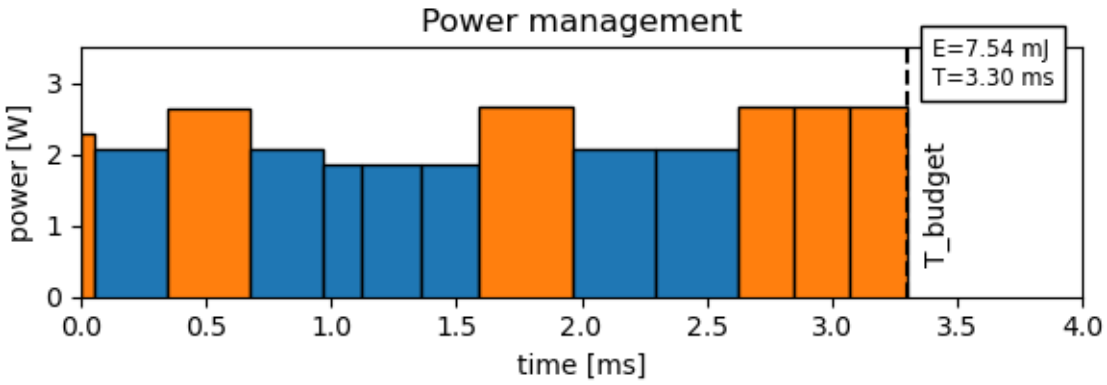
Performance Models for SpiNNaker2 MLA

- Simple grey box models for time and energy
- Time model has relative error < 9 %
- Energy model has relative error < 5 %

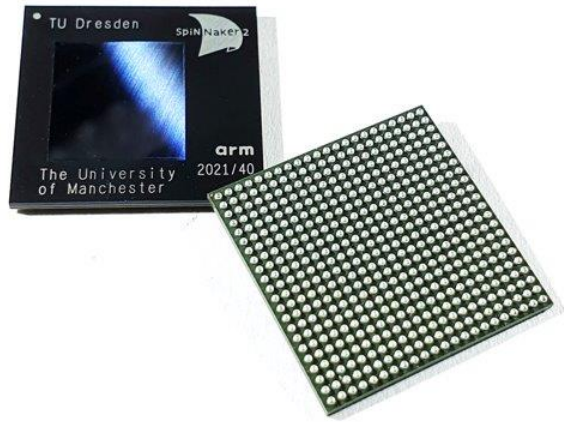


Power management concepts

- Concept for power management in many-core hardware with two DVFS settings for DNN inference
- Approach transferable to other hardare systems
- Predicted energy savings:
  Up to 12 % possible for full network



Outlook



Performance models

- Obtain performance models for **SpiNNaker2 chip**
- **Extend to other DNN layers** and operations, e.g.,
  - MaxPool in ARM Core
  - Data transfer and alignment

Power management concepts

- Validate concepts for full networks on **SpiNNaker2 chip**
- Try **more sophisticated** concepts
- Integrate performance models into **SpiNNaker2 DNN compiler** (in development, based on Apache TVM)
  → automatic selection of energy-optimal **mapping** AND **scheduling**

Scaling

- Scale models and power management to 48 node boards and 5 M Core SpiNNcloud (available 2nd half of 2023)

References

- **[Cordts 2016]** M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The Cityscapes Dataset for Semantic Urban Scene Understanding,” CVPR 2016.
- **[Hoeppner 2019]** S. Höppner, B. Vogginger, Y. Yan, A. Dixius, S. Scholze, J. Partzsch, F. Neumärker, S. Hartmann, S. Schiefer, G. Ellguth, L. Cederstroem, L. A. Plana, J. Garside, S. Furber, and C. Mayr, “Dynamic power management for neuromorphic many-core systems,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 8, pp. 2973–2986, 2019.
- **[Kelber & Wu et al 2020]** F. Kelber, B. Wu, B. Vogginger, J. Partzsch, C. Liu, M. Stolba, and C. Mayr, “Mapping deep neural networks on spinnaker2,” in Proceedings of the Neuro-inspired Computational Elements Workshop, 2020, pp. 1–3. [Online]. Available: https://easychair.org/publications/preprint/JPPG
- **[Wu et al. 2019]** Y. N. Wu, J. S. Emer, and V. Sze, “Accelergy: An architecture-level energy estimation methodology for accelerator designs,” in 2019 IEEE/ACM International Conference on Computer-Aided Design (IC-CAD). IEEE, 2019, pp. 1–8.