Fast Yet Accurate Timing and Power Prediction of Artificial Neural Networks Deployed on Clock-Gated Multi-Core Platforms

Quentin Dariol*
Sébastien Le Nours
Sébastien Pillement
Nantes Université - IETR UMR CNRS 6164, Nantes, France
quentin.dariol@univ-nantes.fr

Ralf Stemmer
Domenik Helms
Kim Grüttner
German Aerospace Center (DLR), Oldenburg, Germany

ABSTRACT

When deploying Artificial Neural Networks (ANNs) onto multi-core embedded platforms, an intensive evaluation flow is necessary to find implementations that optimize resource usage, timing and power. ANNs require indeed significant amounts of computational and memory resources to execute, while embedded execution platforms offer limited resources with strict power budget. Concurrent accesses from processors to shared resources on multi-core platforms can lead to bottlenecks with impact on performance and power. Existing approaches show limitations to deliver fast yet accurate evaluation ahead of ANN deployment on the targeted hardware. In this paper, we present a modeling flow for timing and power prediction in early design stage of fully-connected ANNs on multi-core platforms. Our flow offers fast yet accurate predictions with consideration of shared communication resources and scalability in regards of the number of cores used. The flow is evaluated on real measurements for 42 mappings of 3 fully-connected ANNs executed on a clock-gated multi-core platform featuring two different communication modes: polling or interrupt-based. Our modeling flow predicts timing with 97 % accuracy and power with 96 % accuracy on the tested mappings for an average simulation time of 0.23 s for 100 iterations. We then illustrate the application of our approach for efficient design space exploration of ANN implementations.

KEYWORDS

Power Model, Artificial Neural Networks, Multi-Core, System Level Simulation

ACM Reference Format:
Quentin Dariol, Sébastien Le Nours, Sébastien Pillement, Ralf Stemmer, Domenik Helms, and Kim Grüttner. 2023. Fast Yet Accurate Timing and Power Prediction of Artificial Neural Networks Deployed on Clock-Gated Multi-Core Platforms. In *Proceedings of the 2023 Workshop on System Engineering for constrained embedded systems (RAPIDO 2023), January 17–18, 2023, Toulouse, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3579170.3579263

*Also with DLR.

RAPIDO 2023, January 17–18, 2023, Toulouse, France

ACM ISBN 979-8-4007-0045-3/23/01…$15.00
https://doi.org/10.1145/3579170.3579263

1 INTRODUCTION

With the tremendous growth of the Internet-of-Things market, the need for smart applications requiring the execution of Artifical Intelligence algorithms on edge devices have become predominant. Commonly Artificial Neural Networks (ANNs) are the main adopted solution to solve classification or regression problems in that scope. The deployment of ANNs on edge devices is tough due to their high computational and memory needs, while resources are limited on embedded platforms with strict power budget. The optimization of ANN algorithms on resource limited multi-core platforms with shared communication resources requires intensive evaluation to meet timing and power constraints. To help developers finding optimized ANN deployment on edge devices, an early and efficient evaluation flow is required. Several approaches are already proposed in that field. Many focus on evaluation through systematic implementation and characterization of ANNs on a real platform [20] [17] [4]. Others propose analytical models that offer limited scalability for a wide range of platforms of varying complexity [13] [6]. Especially on multi-core platforms one difficulty comes from the accurate prediction of shared resource contention, which has non-negligible impact on timing and power consumption.

In this paper we propose a modeling flow for fast yet accurate timing and power prediction of ANN deployments on multi-core platforms. Our flow supports scalability regarding the partitioning of the ANN, the number of cores used and the communication workload. It can be used for platforms that rely on either polling-based or interrupt-based communications and which include power management of the platform resources through the use of clock gating. In this paper we show that the communication procedure that optimizes timing and power differs based on the ANN deployment at test and must therefore be established on a case-by-case basis. Our flow can thus help engineers identify the adequate communication procedure in order to optimize the ANN deployment for the considered applications. To validate the proposed modeling flow, we compared the average predicted power consumption against the real power consumption measured on an implementation platform. The modeling flow has demonstrated to predict timing with 97 % accuracy and power with 96 % accuracy for 42 mappings of 3 fully-connected ANNs on multi-core platforms containing up to 7 cores. We also show how the modeling flow can be used in a Design Space Exploration (DSE) process in order to identify deployments that optimize energy, power, latency and cost.

In Section 2 we present the contributions of our work compared to the state-of-the-art. We then explain our work hypothesis and how the power model was built and integrated into our system level modeling flow in Section 3. In Section 4 we present and discuss the

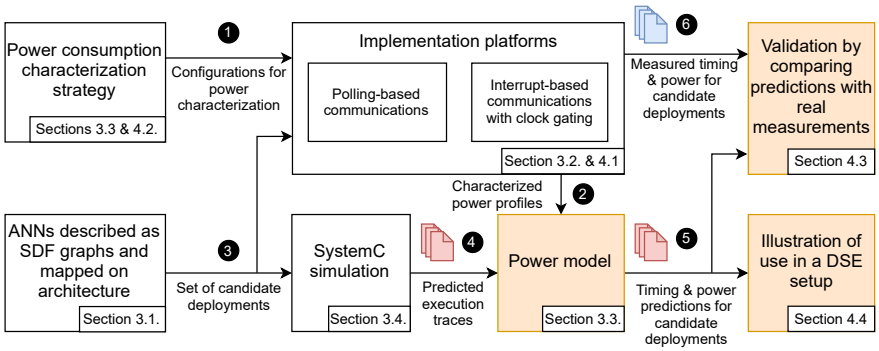

Figure 1: Overview of the proposed modeling flow. Contributions are highlighted in orange. Red files correspond to predicted time and energy files. Blue files correspond to measured time and power.

experimental setup used to perform the calibration of the power model and the validation of our approach. An example of the use of our modeling flow to perform DSE is also given. Section 5 provides our conclusion on this work and prospects for future work.

2 RELATED WORK

Several approaches have already been led to tackle the challenge of power/energy evaluation of ANNs on embedded systems. An important number of these approaches is focused on the monitoring of the ANN deployed on the targeted device in a Hardware-in-the-Loop setup [10] [2] [20] [17] [4]. These approaches rely on an automatized exploration workflow to generate and test numerous possible mappings through their deployment on the targeted edge device and evaluation through measurements. The effort to obtain evaluation results is however important as all mappings must be deployed and tested. The implementation technology is also fixed due to the need of having the real platform in the loop which can limit the possibilities in regards to architectural exploration.

To offer an evaluation of candidate mappings before their deployment, other flows proposed DSE setups relying on the use of a power model. NNest [6], Timeloop [13] and MAESTRO [7] are systematic approaches that offer an evaluation of time, power and cost of candidate ANN accelerators implementations using high level models. In these approaches, the emphasis is put on the exploration of algorithmic optimizations on the ANN and architectural alternatives simultaneously to offer a highly optimized accelerator. Communications in the accelerator are built based on the dataflow of the input ANN, which leads to highly optimized communications which can be modeled using elementary analytical models. These approaches are hence not easily transposable to tile-based multi-core architectures, on which overheads in latency and power consumption due to communications are not marginal and must be properly modeled. For these architectures, a system level simulation flow must be proposed in order to guarantee an accurate prediction of latency, power and cost in reasonable time.

The authors of [19], [11] and [1] propose a system level simulation flow for multi-core platforms with power models calibrated through measurement. [16] proposes a systematic methodology to automatically calibrate power models for simulation components based on execution traces. When targeting ANN implementation, the modeling can be further adapted to offer faster prediction with similar accuracy. In [3] we proposed our hybrid simulation flow based on analytical elementary models calibrated through measurements for timing prediction of fully-connected ANNs deployed on multi-core platforms. In this paper, we extend this work to the following contributions:

(1) A power modeling flow for ANNs deployed on multi-core platforms. We propose a power model calibrated by measurement for fully-connected ANNs deployed on multi-core platforms which implements a clock gating mechanism and either polling or interrupt-based communications.
(2) The validation and assessment of efficiency of the modeling flow by comparing time and power predictions against real measurements. This flow has been tested and validated for several mappings of fully-connected ANNs on platforms containing up to 7 cores, and is respectively 97 % and 96 % accurate for timing and power prediction.
(3) The demonstration of the application of the proposed modeling flow for DSE of ANN mappings under timing, power and cost (number of cores) constraints with prediction of the impact of using polling or interrupt-based communications.

3 PROPOSED MODELING FLOW

3.1 Flow presentation and work hypothesis

Figure 1 gives an overview of the workflow to build and validate the proposed power model. The power model expresses the dynamic and static power consumption of the multi-core platform executing ANNs. It is further presented in Section 3.3. This model is based on the activity executed by cores on the platform, and is characterized through measurement. The first phase aims at calibrating the power model. During the execution of ANNs on the platform, cores perform two main activities: neurons computation and communications. The power consumption of cores differs based on the activity being performed. We define therefore a power consumption characterization strategy ❶ in order to calibrate the power model in regards to these activities. Different mappings are considered independently of the applications in order to characterize the static power consumption of the platform, the evolution of the consumption based on the number of cores running and the consumption of

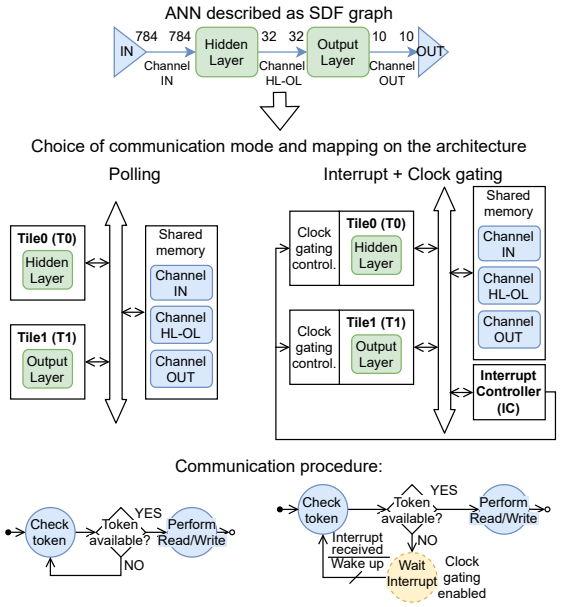

Figure 2: Example of mapping of an ANN described as SDF graph on the two considered architectures.

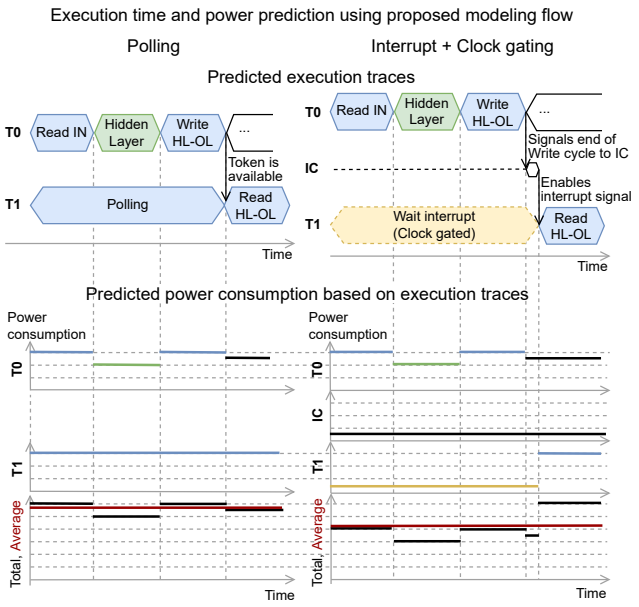

Figure 3: Observation of possible execution traces and predicted power for the two communication modes.

the bus and shared memory. We then perform a regression on the obtained power profiles to calibrate the power model ❷.

Once the power model calibrated, the next activity aims at the validation of the modeling flow for timing and power prediction of ANNs. To demonstrate the scalability of the flow, several mappings of different fully-connected ANNs are considered ❸. In this work we rely on the Synchronous Data Flow (SDF) [9] Model of Computation (MoC). It allows several representations of ANNs at various levels of granularity. We reuse the definition introduced in [3] for the partitioning of SDF graphs: each layer of the ANN is partitioned into actors, which are composed of a set of neurons. Actors issued from the same ANN layer have equitable workload in terms of neurons to compute. The communication channels of the SDF graph correspond to the communications between actors. Several SDF graphs of different complexity in terms of number of actors and communication channels can be generated from the same ANN. The Model of Architecture (MoA) is composed of a set of tiles, each composed of a single Processing Element (PE) and a private memory for instructions and data. A shared memory is accessible through a communication bus. Actors are mapped on tiles and communication channels are mapped on the shared memory. In this work, two versions of the MoA are considered with different communication procedures: one features polling-based communications, the other interrupt-based communications. Tiles also implement a clock gating mode in which their power consumption is reduced. An example of ANN mapped on the two different versions of the MoA is given in Figure 2.

The execution traces of candidate deployments are predicted using the SystemC simulation approach presented in [3] ❹. The power model uses the execution traces from the SystemC simulation to predict the power consumption of the system ❺. To validate the power modeling flow, we also measure the real power consumption of candidate deployments on the implementation platform ❻ and we evaluate the accuracy of the predictions against these measurements. Once the model validated, it can be used in a DSE setup to find ANNs deployment that meet latency, power, energy and cost constraints.

3.2 Power management strategy

When executing ANNs on multi-core platforms, dependencies between actors in the application can lead to important durations when tiles are blocked due to the unavailability of resources needed for their execution. An example is given in Figure 3: Tile1 (T1) needs to read data contained in the communication channel HL-OL to execute, but this data has first to be written by Tile0. Therefore Tile1 is blocked and must wait until the data is available. When waiting tiles are not contributing to the execution of the ANN. They can therefore be switched to a lower power consumption mode. To this end we choose to implement the clock gating power saving mechanism, as presented in [14]. This power management technique allows to dynamically reduce the power consumption of tiles by temporarily disabling their input clock signal. Clock gating does not reduce the static power consumption of circuits as they are still supplied with power. It however allows removing the dynamic part of power consumption as all activities are suspended when the clock is disabled. It also has the advantage of offering minimal overheads in latency due to the ease of enabling and disabling the clock signal and the low delay needed by the tile to resume its execution. In order to evaluate the impact of clock gating on ANN execution on tile-based multi-core platforms, we considered two versions of the MoA, as depicted in Figure 2:

(1) One version with polling-based communications and without the use of clock gating: when a tile needs unavailable data to execute, it polls the shared memory until the data is available.
(2) One version with interrupt-based communications and with the use of clock gating: when a tile needs unavailable data to execute, it enters low power mode until it is woken up by an interrupt signal, which indicates that the data is now available.

The two different communication modes are illustrated in Figure 2. The interrupt-based platform introduces an interrupt controller peripheral to manage the interrupt signal, and clock gating controllers to manage the activation and deactivation of clock gating mode. The clock gating mode is activated when a tile requires currently unavailable data. A clock-gated tile resumes its execution when the interrupt signal is enabled. The interrupt signal is enabled by the interrupt controller peripheral when requested by a tile. A tile requests the interrupt controller to generate a interrupt following a read/write transaction, in order to indicate to other tiles which are currently clock-gated that data is now available in the shared memory. Figure 3 shows how the communications are handled on the two different architectures, and the impact on power consumption. The additional circuits of the interrupt-based platforms add a contribution to power. However, they allow using the clock gating mode when waiting for data, instead of polling on the shared memory, which leads to power reduction. The best communication procedure when considering the impact on power consumption depends therefore on the studied deployment, and must be addressed on a case-by-case basis by the evaluation flow.

3.3 Power modeling

The power model is built by performing power characterization through measurements on the targeted platform, as depicted in Figure 1 with phases ❶ and ❷. During the execution of an ANN, tiles execute computation or communication activities. The dynamic power consumption of the platform executing ANNs can thus be described using two terms: $P^{\Sigma}_{comp}(t)$, the total power consumption of tiles in computation mode at time t and $P^{\Sigma}_{comm}(t)$, the total power consumption in communication mode at time t. To obtain the total power consumption of the system $P^{\Sigma}(t)$, the static power consumption of the circuit P_{static} must also be considered. Equation 1 gives the proposed model for the total instant power consumption of ANNs on tile-based multi-core platforms.

$$P^{\Sigma}(t) = P_{static} + P^{\Sigma}_{comp}(t) + P^{\Sigma}_{comm}(t) \tag{1}$$

When tiles are in the computation activity, they are processing actors (i.e. sets of neurons as presented in Section 3.1). They are therefore independent from one another as all necessary data is contained in the private memory of the tile. We introduce $P_{comp,i}$, the power consumption of tile i in computation mode. The platform is assumed to be homogeneous, hence $P_{comp,i} = P_{comp} \forall i$. We also introduce $\alpha_{comp,i}$, the computation activity factor of tile i. $\alpha_{comp,i}$ is equal to 1 at time t if tile i is currently in computation mode else, it is equal to 0. $\alpha_{comp,i}$ is obtained from the estimated execution traces obtained from the SystemC model (See Figure 1, phase ❹). The evolution of P^{Σ}_{comp} increases hence linearly with the number of tiles currently in this activity. Equation 2 gives the model for P^{Σ}_{comp}.

$$\begin{aligned} P^{\Sigma}_{comp}(t) &= \sum_{i=1}^{N_{tiles}} P_{comp,i}\alpha_{comp,i}(t) \\ &P_{comp,i} = P_{comp} \quad \text{(Homogeneous platform hypothesis)} \\ \text{with } &\alpha_{comp,i}(t) = \begin{cases} 1 & \text{if tile } i \text{ in computation mode at time } t \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

When tiles are in the communication activity, they either: 1. perform a read or write in the shared memory, 2. wait for the availability of data. As depicted in the automates on Figure 2, tiles operate differently when waiting based on the selected communication procedure: they either poll the shared memory or enter clock gating mode. Read, write and polling operations in the shared memory are intertwined and are thus modeled together as P^{Σ}_{RWP}. The contribution of clock gated tiles on the total power consumption is modeled as P^{Σ}_{CG}. Equation 3 gives the model for P^{Σ}_{comm}.

$$P^{\Sigma}_{comm}(t) = P^{\Sigma}_{RWP}(t) + P^{\Sigma}_{CG}(t) \tag{3}$$

We introduce P_{SM} the consumption of tiles when accessing the shared memory. The shared memory is a unique shared resource. When several tiles try to access the shared memory simultaneously, only one tile get access to it and the others are postponed until the shared memory is free. The shared memory can be accessed by only one tile at any given time t. The total power consumption in read, write and polling activities $P^{\Sigma}_{RWP}(t)$ is hence either equal to P_{SM} when at least one tile is in this activity, or 0 otherwise. We introduce the read, write and polling activity factor for tile i at time t: $\alpha_{RWP,i}(t)$. $\alpha_{RWP,i}(t)$ is equal to 1 if tile i is executing this activity at time t else 0 and is obtained from the estimated execution traces. The model for P^{Σ}_{RWP} is given in Equation 4.

$$\begin{aligned} P^{\Sigma}_{RWP}(t) &= P_{SM} \bigcup_{i=1}^{N_{tiles}} \alpha_{RWP,i}(t) \\ \text{with } &\alpha_{RWP,i}(t) = \begin{cases} 1 & \text{if tile } i \text{ uses the shared memory} \\ & \text{at time } t \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

$$\begin{aligned} P^{\Sigma}_{CG}(t) &= \sum_{i=1}^{N_{tiles}} P_{CG,i}\alpha_{CG,i}(t) \\ &P_{CG,i} = P_{CG} \\ \text{with } &\alpha_{CG,i}(t) = \begin{cases} 1 & \text{if tile } i \text{ is in clock gating (CG)} \\ & \text{mode at time } t \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

When tiles are clock gated, they are independent from one another. We introduce $P_{CG,i}$, the power consumption of tile i in computation mode. The platform is assumed to be homogeneous, hence $P_{CG,i} = P_{CG}$. The evolution of $P_{CG,i}$ increases hence linearly with the number of tiles currently in this activity. We introduce the clock gated activity factor for tile i at time t: $\alpha_{CG,i}(t)$. $\alpha_{CG,i}(t)$ is equal

to 1 if tile i is clock gated at time t else 0 and is obtained from the estimated execution traces. Equation 5 gives the model for P^{Σ}_{CG}.

$$\begin{aligned}(E_K) : P^{\Sigma}_{measured,K} =& P_{static} + \sum_{i=1}^{N_{tiles}} \alpha_{comp,i,K} P_{comp,i} \\ &+ P_{SM} \bigcup_{i=1}^{N_{tiles}} \alpha_{RWP,i,K} \\ &+ \sum_{i=1}^{N_{tiles}} \alpha_{CG,i,K} P_{CG,i}\end{aligned} \tag{6}$$

$\alpha_{comp,i}(t)$, $\alpha_{RWP,i}(t)$ and $\alpha_{CG,i}(t)$ are obtained through the estimated simulation traces (Figure 3). The base power consumptions P_{static}, P_{comp}, P_{RWP} and P_{CG} must be obtained through characterization. To obtain the base power consumptions, we measure power profiles on the platform associated to the identified activities, and apply multi-linear regression to solve a system of equations (E_K) as presented in Equation 6. In this equation K corresponds to the number of a measured power characterization profile. K is also the equation number in the system. The α_K terms are set for the configuration at test and $P^{\Sigma}_{measured,K}$ is obtained through measurement. At least N_{tiles} power characterization profiles must be obtained for each identified activity (computation and communication: read, write, polling and clock gating) to correctly perform the characterization. To obtain the power characterization profiles we measure the power consumption of the system for each activity in the following configurations:

(1) All tiles are disabled to characterize P_{static}.
(2) Each tile is set to execute the activity individually when others are disabled to verify that all tiles have the same contribution to power consumption and thus the validity of the homogeneous platform hypothesis,
(3) Tiles are progressively enabled and tasked to execute the activity to characterize the evolution of power when several tiles are enabled together.
(4) Complementary configurations are also considered to further constrain and improve the quality of the regression of the power profiles. Testing all possible configurations on platforms with numerous tiles would require an important characterization effort. Instead we generated randomly a tenth of additional configurations in which tiles are arbitrarily either set in activity or disabled.

3.4 Integration in SystemC simulation

To predict the latency of ANNs on tile-based multi-core platforms, we presented in [3] a high level simulation flow in SystemC. The simulation flow allows predicting the execution traces and deduce from them the latency. The flow relies on two separate base models calibrated through measurement: one for actor computation time and one for communication time. Compared to [3] we have updated the communication time model calibration for interrupt-based communications with clock gating. The execution traces delivered by the simulation flow contain time markers with current mode for each tile. A new marker is generated when a tile changes its state during the execution. The tile state corresponds to the activity being executing (actor computation, read/write transaction on shared memory, clock gating, polling). Figure 3 gives an example of traces output by the SystemC simulation. The power model presented in Section 3.3 is used as post processing of the execution traces output by the SystemC simulation. The model is used to predict the evolution of power consumption during the execution of the ANN.

4 EXPERIMENTAL SETUP AND RESULTS

4.1 Experimental setup

In order to calibrate the proposed modeling flow and evaluate the prediction accuracy, we implemented a tile-based multi-core platform which respects the MoA to obtain real time and power measurements. Two versions of the platform are implemented: one which relies on interrupt-based communications with clock gating and one which relies on polling-based communications without clock gating, as presented in Figure 2. The platforms are implemented on a Xilinx ZCU102 board, which features a UltraScale MPSoC+ FPGA. They are implemented on the programmable logic section of the FPGA. The processing core of the tiles is a MicroBlaze. The private memory of tiles and the shared memory are implemented as Block Rapid Access Memory (BRAM), which is internal to the FPGA SoC. The communication medium to access the shared memory is implemented as a shared interconnect Advanced eXtensible Interface (AXI). In the clock gating version, the interrupt controller and the clock gating controller are IPs provided by Xilinx. The implementation platform is composed of 7 tiles.

The targeted FPGA features several power supply levels. We probed *VCCINT* which corresponds to the supply voltage of the programmable logic of the FPGA and *VCCBRAM* which corresponds to the supply voltage for the BRAM. The BRAM supply voltage is already well optimized by the provider as discussed in [15], and we observed no variation of power consumption on *VCCBRAM* when performing our tests. In fact, the activity linked to memory usage is observed rather on the *VCCINT* power supply, as the memory controllers are implemented on the programmable logic part of the FPGA. The correctness of this observation is confirmed through the use of the Xilinx Power Estimator (XPE) [18] available in Vivado, which predicts marginal dynamic contribution to the system consumption on the *VCCBRAM* voltage supply. For this reason, we focus entirely on *VCCINT* power consumption in this study.

The measurement infrastructure for timing validation is presented in [3]. The power measurements are obtained using the R&S HMC8012 Digital Multimeter at a sampling rate of $\approx$ 100 samples per second. As the execution times measured are in the order of milliseconds or even hundreds of microseconds, we took a large number of records (4000) for each considered measurement in order to obtain a representative sample of the system consumption. We then checked that the deviation was marginal and used the average of the measurements.

4.2 Power model calibration

As presented in Section 3.3, we measured the power consumption profiles of the platform in order to calibrate the proposed model. Using multi-linear regression, we validated the hypothesis that the platform is homogeneous so the tiles share the same base power

consumption for each identified activity. We observed that the static consumption of the platform with interrupt-based communications is 4 % higher than its polling counterpart, with $P_{static,polling}$ measured equal to 1.227 W and $P_{static,interrupt} = 1.260$ W. This raise of the observed static consumption is due to the introduction of the interrupt and clock gating controllers. We also measured: $P_{comp} = 0.058$ W, $P_{SM} = 0.063$ W and $P_{CG} = -0.058$ W. P_{CG} is set negative by the multi-linear regression tool due to the use of P_{static} as reference, which is obtained when all tiles are disabled. Clock gated tiles consume less than disabled tiles, which leads thus to a negative value. We observed that when all tiles are clock gated on the interrupt-based platform, the power consumption is reduced of 32 % compared to the consumption of the polling-based platform when all tiles are disabled.

To provide a more thorough verification of our calibration, we compare our power model with the estimations provided by XPE [18]. XPE is a tool included in Vivado which allows predicting the power consumption of a FPGA design after the synthesis and implementation steps. The tool offers two default settings: *maximum* for high activity rates of all components of the system and *typical* for standard use activity rates. We used the *typical* setting. When comparing the power model built through characterization with the consumption predicted by this tool we observed that our measurements and the built power model were in the same scale as the predicted data from XPE. The static consumption of the circuit on VCCINT are estimated respectively as 1.227 W and 1.260 W for the two versions of the platform by XPE which corresponds precisely to the measured static consumption. XPE estimates that the AXI shared memory controller consumes 0.024 W and tiles consume 0.050 W on average which also fits our observations for P_{comp} and P_{SM}. The *typical* setting doesn't allow however verifying the consumption of tiles in clock gated mode (P_{CG}).

4.3 Validation of the modeling flow

Three fully-connected ANNs have been considered to test our power modeling flow. These different use-cases offer a variety of complexity as well as different communication workloads which leads to a comprehensive validation of our modeling flow. Two of these ANNs were trained on the MNIST [8] handwritten digit recognition dataset. Both have a input layer composed of 784 neurons and an output layer composed of 10 neurons. They differ in the number and size of hidden layers: the first network, referred to as *M1* in the rest of paper has one hidden layer of 10 neurons, while the second network, referred to as *M2*, is composed of two hidden layers of respectively 32 and 16 neurons. *M1* has 85 % classification accuracy and *M2* has 89 % classification accuracy. The third considered ANN, referred to as *G*, was trained on the German Traffic Sign Recognition Benchmark (GTSRB) [5] dataset. In order to enable the processing on the platform due to the complexity of the dataset, we had to remove neurons to reduce data size of the GTSRB ANN. For this reason *G* has only 20 % classification accuracy. *G* is composed of a input layer of 512 neurons, two hidden layers of 30 neurons each and an output layer of 43 neurons. The three considered ANNs were trained using the LibFANN open source deep learning framework [12]. They all use the ReLU activation function. We tested and stressed our flow in regards to the following aspects:

(1) Consideration of applications of variable complexity. We used SDF graphs with limited amount of actors and communication channels, respectively 2 and 3 as lowest, as well as complex SDF graph featuring up to 22 actors and 113 communication channels.
(2) Consideration of single-core mappings as well as multi-core mappings containing up to 7 tiles. The multi-core mappings leverage both ANN in-layer parallelism and intra-layer parallelism. This allows optimizing both inference time and throughput for ANN execution.
(3) Consideration of mappings with high and low communication rates. The lowest observed communication rate is 2 % on one mapping and the highest is 70 % on another mapping.
(4) Consideration of mappings with polling-based communications without the use of clock gating and mappings with interrupt-based communications in which cores are clock gated when waiting for the availability of data.

A total of 42 different mappings are tested, which corresponds to different partitionings and mappings of the 3 considered fully-connected ANNs. We predicted the end-to-end latency, execution traces and power of the considered mappings, and we measured the real end-to-end latency and power on the execution platform for validation. The power predictions are compared with the power measurements obtained through the same power measurement infrastructure that allowed performing the calibration.

The graphs in Figure 4 show the predicted end-to-end latency and power consumption and the prediction error against the measurements for each tested mapping. The presented power consumption values in the graphs contain both static and dynamic components. The graph is separated into two sub-graphs: the first one is dedicated to polling-based communications, the second to interrupt-based communications with clock gating. Each sub-graph is then separated into two categories: single-core and multi-core executions. Table 1 summarizes the results by providing the average and worst observed error of tested scenarios.

Table 1: Average and worst prediction error of the modeling flow against measurements on considered mappings.

Type of scenario	Polling				Clock gating + interrupt			
	Timing		Power		Timing		Power	
	avg	worst	avg	worst	avg	worst	avg	worst
Single core	0.36 %	0.75 %	1.97 %	2.28 %	0.39 %	0.95 %	0.63 %	2.08 %
Multi core	0.59 %	2.85 %	1.06 %	2.43 %	0.89 %	2.21 %	1.64 %	3.93 %

On the 42 tested mappings, the proposed modeling flow offers a prediction accuracy of 97 % on timing and 96 % on power. As shown in Table 1, the worst observed prediction error is 2.85 % on timing and 3.93 % on power. The timing prediction error is very low for both single-core and multi-core mappings with both communication procedures as shown in Table 1 and Figure 4. The highest observed errors are 2.85 %, reached with polling-based communications on *M1* partitioned in 7 actors and mapped on 7 cores with a communication rate of 70 %, and 2.21 % for the interrupt-based communications on *M1* with 15 actors and mapped on 7 tiles with a communication rate of 55 %. All other tested mappings have prediction errors lower than 2 %.

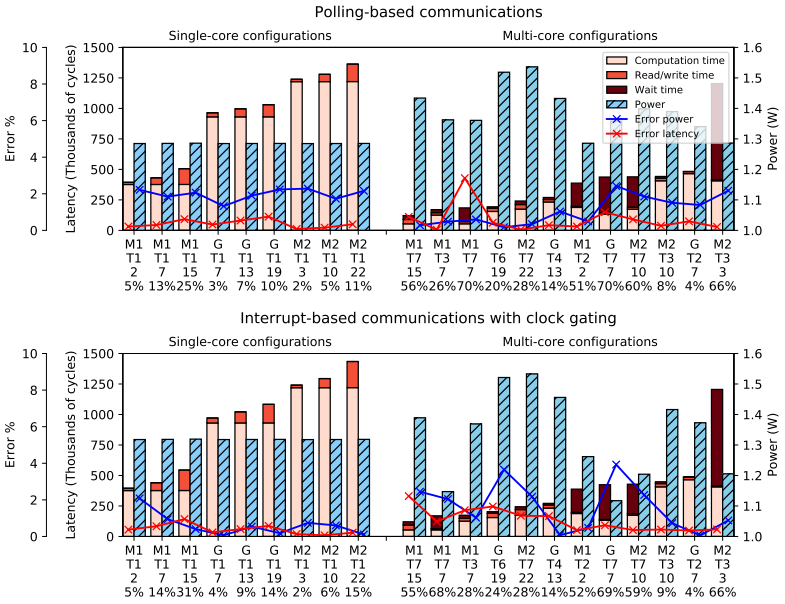

Figure 4: Predicted end-to-end latency (in processor cycles) and power consumption (in W) by the proposed modeling flow for 42 mappings of 3 different fully-connected ANNs with polling-based and interrupt-based communications. In X-axis the tested ANN *M1*, *M2* or *G*, followed by the number of tiles used T_N, the number of actors and the average communication rate (time spent in read, write and waiting) of tiles are provided.

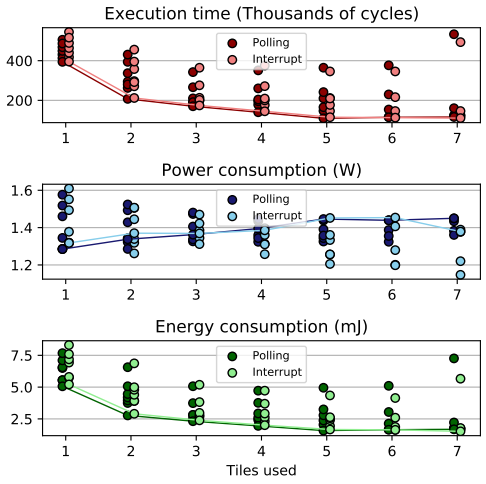

Figure 5: Use of the modeling flow to evaluate the execution time (in processor cycles), power (in W) and energy (in mJ) of 84 mappings of the ANN *M1*, with polling-based and interrupt-based communications. Mappings that optimize the energy consumption are connected with lines.

The power predictions are satisfying for both considered communication procedures as shown in Table 1 and Figure 4. The highest power prediction error is 3.93 % on *G* with 7 actors executed on 7 cores with interrupt-based communications, and 2.43 % for the same mapping with polling-based communications. This mapping presents an estimated communication rate (waiting + read/write) of approximately 70 % regardless of the communication procedure, and in both cases the power model underestimates the power consumption. On the polling-based platform, this can be explained by the high amount of simultaneous polling on the shared memory. On the interrupt-based platform, only one interrupt signal is implemented. Therefore all clock-gated tiles are simultaneously woken up and try to obtain data from the shared memory. The updated data is only available for one tile, the others re-enters the clock-gated mode again after checking the shared memory. However, due to all tiles simultaneously accessing the shared memory, overheads in power consumption which are not modeled can be observed. At the proposed level of abstraction, the observed error is acceptable. Overall on the 42 considered mapping, the average prediction error of the power model is less than 2 %.

A key advantage of the proposed modeling flow is the simulation time required to obtain time and power predictions. The high level of abstraction of the models combined with the characterization through measurements allows offering a simulation time of 0.23 s on average for 100 simulated iterations of the application on the platform on an Intel Core i5-8250U CPU @ 1.60 GHz. When also including the compilation time of the model, the time taken to evaluate a mapping is on average 20 s which is 2 times less than the compilation, programming and measurement time on the real platform using our configuration (approximately 40 s). XPE [18] could also be used to estimate the power consumption with a evaluation time of approximately 3 minutes (without considering the synthesis of the FPGA design and configuration of XPE). The low evaluation time of our modeling flow is an important feature as it can be used to perform a fast and intensive evaluation of ANN mappings on multi-core platforms and help engineers to quickly identify optimized solutions that meet their constraints.

The proposed modeling flow is built with the requirement to support scalability. Re-using the proposed flow on different tile-based multi-core architectures should require minimum effort. The timing modeling flow has been characterized on AXI bus communication, hence targeting different communication medium would require another characterization phase. When calibrating the power model, chip provider data can be used when available instead of a characterization through measurement to minimize development effort. Once done, this flow is able to predict accurately the execution time and power consumption of any fully-connected ANN deployed on tile-based multi-core platforms.

4.4 Use of the modeling flow to explore the design space

Once validated the modeling flow can be used to explore the design space in order to find mappings that optimize the deployment of a user defined ANN. An example of the use of the modeling flow evaluating 84 different mappings for *M1* in regards to latency,

power, energy, and number of cores is given in Figure 5. The energy consumption for one iteration is obtained by multiplying the execution time with the average power consumption. The Pareto front of mappings that optimize energy is depicted using lines on the energy consumption curve. We also connected these mappings on the plots of execution time and power consumption.

When comparing the mappings of the Pareto front when using up to 3 tiles, polling-based communications allows saving energy over interrupt-based communications. When using 6 and 7 tiles interrupt-based communications offer lower energy consumption. The difference between the two communication modes is marginal for 4 tiles. Polling-based communications allow however saving 6 % energy on 5 tiles and offer the second lowest energy consumption (1.58 mJ) identified during the evaluation. The lowest energy consumption (1.52 mJ) is obtained with interrupt-based communications on 7 tiles, and it allows saving 11 % energy compared to the best mapping on 7 tiles using polling. This shows that the comparison of candidate deployments using the modeling flow can lead to important energy savings. It can be noted that for the considered application, mappings that offer the most optimized energy consumption for a given number of tiles also offer the most optimized latency for that tile number.

The compilation and simulation of 100 iterations for each of the 84 candidate deployments required 28 minutes in total (approximately 20 s per candidate) on an Intel Core i5-8250U CPU @ 1.60 GHz. In comparison, we estimate that it would take approximately 1 hour when using our automatized deployment and measurement infrastructure on the real platform (40 s per candidate), and 4 hours using XPE (3 min per candidate, without considering the synthesis of the FPGA design and configuration time of XPE). The proposed approach can thus lead to important gains of time when evaluating numerous candidate deployments.

5 CONCLUSION

This paper presents a timing and power modeling flow for evaluation of fully-connected ANN deployment on multi-core platforms with clock gating capabilities. The power model is calibrated through measurement by characterizing the power consumption of activities led by cores on the platform when executing ANNs. The modeling flow allows predicting timing with more than 97 % accuracy and power with more than 96 % accuracy on 42 different mappings of three fully-connected ANNs. The models are fast to execute with an average simulation time of 0.23 s for 100 iterations. This approach allows comparing two different communication procedures: polling-based communications without the use of clock gating, and interrupt-based communications with the use of clock gating. The use of the flow to evaluate possible mappings of one ANN on up to 7 cores allows identifying the number of cores and communication procedure to use in order to optimize timing, power and energy consumption. The proposed flow can be calibrated to be used with other tile-based multi-core platforms either through the use of power profiling data given by the chip provider or through measurements. In future work, we will extend this flow to other classes of neural networks such as CNNs and we will automatize the use of the modeling flow in a DSE loop to find deployments that meet user defined constraints.

ACKNOWLEDGEMENTS

This work has been funded by the WISE consortium, France in the project pSSim4AI and by the Federal Ministry of Education and Research (BMBF, Germany) in the project Scale4Edge (16ME0465).

REFERENCES

[1] M. M. Ayub and F. Kreupl. 2020. A Modular and Distributed Setup for Power and Performance Analysis of Multi-Processor System-on-Chip at Electronic System Level. In *2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC)*. IEEE, Cancun, 1–8.

[2] Y.H. Chen, J. Emer, and V. Sze. 2017. Using Dataflow to Optimize Energy Efficiency of Deep Neural Network Accelerators. *IEEE Micro* 37 (2017), 12–21.

[3] Q. Dariol, S. Le Nours, S. Pillement, R. Stemmer, D. Helms, and K. Grüttner. 2022. A Hybrid Performance Prediction Approach for Fully-Connected Artificial Neural Networks on Multi-core Platforms. In *International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS 2022)*, Alex Orailoglu, Marc Reichenbach, and Matthias Jung (Eds.). Springer International Publishing, Samos, 250–263.

[4] L. Heim, A. Biri, Z. Qu, and L. Thiele. 2021. Measuring what Really Matters: Optimizing Neural Networks for TinyML. arXiv:2104.10645

[5] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. 2013. Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*. IEEE, Dallas, 1–8.

[6] L. Ke, X. He, and X. Zhang. 2018. NNest: Early-Stage Design Space Exploration Tool for Neural Network Inference Accelerators. In *Proceedings of the International Symposium on Low Power Electronics and Design* (Seattle) *(ISLPED '18)*. Association for Computing Machinery, New York, Article 4, 6 pages.

[7] H. Kwon, P. Chatarasi, V. Sarkar, T. Krishna, M. Pellauer, and A. Parashar. 2020. MAESTRO: A Data-Centric Approach to Understand Reuse, Performance, and Hardware Cost of DNN Mappings. *IEEE Micro* 40, 3 (2020), 20–29.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[9] E.A. Lee and D.G. Messerschmitt. 1987. Synchronous data flow. *Proc. IEEE* 75, 9 (1987), 1235–1245.

[10] M. Motamedi, D. Fong, and S. Ghiasi. 2016. Fast and Energy-Efficient CNN Inference on IoT Devices. arXiv:1611.07151

[11] M. Moy, C. Helmstetter, T. Bouhadiba, and F. Maraninchi. 2016. Modeling Power Consumption and Temperature in TLM Models. *Leibniz Transactions on Embedded Systems* 3, 1 (2016), 29 pages.

[12] S. Nissen. 2003. Implementation of a Fast Artificial Neural Network library (FANN). https://github.com/libfann/fann

[13] A. Parashar, P. Raina, Y. S. Shao, Y. H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, Brucek Khailany, Stephen W. Keckler, and Joel Emer. 2019. Timeloop: A Systematic Approach to DNN Accelerator Evaluation. In *2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, Madison, 304–315.

[14] J. Rabaey. 2009. *Low Power Design Essentials*. Springer, Boston.

[15] B. Salami, E. B. Onural, I. E. Yuksel, F. Koc, O. Ergin, A. Cristal Kestelman, O. Unsal, H. Sarbazi-Azad, and O. Mutlu. 2020. An Experimental Study of Reduced-Voltage Operation in Modern FPGAs for Neural Network Acceleration. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, Valencia, 138–149.

[16] S. Schürmans, D. Zhang, D. Auras, R. Leupers, G. Ascheid, X. Chen, and L. Wang. 2013. Creation of ESL Power Models for Communication Architectures Using Automatic Calibration. In *Proceedings of the 50th Annual Design Automation Conference* (Austin) *(DAC '13)*. Association for Computing Machinery, New York, Article 58, 58 pages.

[17] O. Spantidi, I. Galanis, and I. Anagnostopoulos. 2020. Frequency-based Power Efficiency Improvement of CNNs on Heterogeneous IoT Computing Systems. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*. IEEE, New Orleans, 1–6.

[18] A. K. Sultania, C. Zhang, D. K. Gandhi, and F. Zhang. 2017. *Designing with Xilinx® FPGAs: Using Vivado*. Springer International Publishing, Cham, Chapter Power Analysis and Optimization, 177–187.

[19] C. Trabelsi, R. Ben Atitallah, S. Meftali, J.-L. Dekeyser, and A. Jemai. 2011. A Model-Driven Approach for Hybrid Power Estimation in Embedded Systems Design. *EURASIP Journal on Embedded Systems* 2011-03-14, 1 (2011), 15 pages.

[20] F. Tsimpourlas, L. Papadopoulos, A. Bartsokas, and D. Soudris. 2018. A Design Space Exploration Framework for Convolutional Neural Networks Implemented on Edge Devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37 (2018), 2212–2221. Issue 11.