31.3 A Compute-Adaptive Elastic Clock-Chain Technique with Dynamic Timing Enhancement for 2D PE-Array-Based Accelerators

Tianyu Jia, Yuhao Ju, Jie Gu

Northwestern University, Evanston, IL

Dynamic timing error detection and correction techniques, e.g. razor flops, have been previously applied to microprocessors to exploit the dynamic timing margin within pipelines [1]. Adaptive clock techniques have also been adopted to enhance microprocessor performance, such as schemes to reduce the timing guardband for on-chip supply droops [2-3] or to exploit instruction-level dynamic timing slack [4]. Recently, 2D PE array-based accelerators have been developed for machine learning (ML) applications. Many efforts have been dedicated to improve the energy efficiency of such accelerators, e.g. DVFS management for the DNN under various bit precision [5]. A razor technique was also applied to a 1D 8-MAC pipelined accelerator to explore timing error tolerance [6]. Despite of the above efforts, a fine-grained dynamic-timing-based technique has not been implemented within a large 2D array based ML accelerator. One main challenge comes from the large amount of compute-timing bottlenecks within the 2D array, which will continuously trigger critical path adaptation or pipeline stalls, nullifying the benefits of previous dynamic-timing techniques [4, 6]. To deal with the difficulty, we propose the following solutions. A local in-situ compute-detection scheme was applied to anticipate upcoming timing variations within the PE unit and guide both instruction-based and operand-based adaptive clock management. To loosen the stringent timing requirements in a large 2D PE array, an "elastic" clock-chain technique using multiple loosely synchronized clock domains was developed enabling dynamic-timing enhancement through clusters of PE units.

Figure 31.3.1 shows the PE array design used in this work, based on a commonly used deep-neural-network (DNN) accelerator supporting dataflow of both 2D SIMD [7] and a tightly-coupled systolic array. Each PE is a configurable MAC unit supporting various dataflows and variable bit precision from 1b to 8b similar to that in [5]. The simulation results of the cycle-by-cycle timing variation of a single PE unit running the MINIST database are shown. A wide range of dynamic timing variation within each PE unit is observed. In addition, the longest critical paths are activated less than 5% of the time and are determined by the operands, e.g. at MSBs change, leading to operand-dependent dynamic timing margin. However, the dynamic timing margin diminishes with the size of PE array increasing, i.e. reducing from 40% to only 4% when the number of PEs increases from 1 to 128, because a critical timing path can be activated in any PE unit. Therefore, centralized adaptive techniques [4, 6] cannot exploit the dynamic timing margin effectively for a large 2D PE array.

Figure 31.3.2 shows the top-level chip architecture. An 8-column 16-row PE array is implemented supporting both 2D SIMD and systolic dataflow with variable bit precision. Each row of 8 local PE units with supporting image/weight SRAMs are clocked by a different clock domain. The critical timing paths inside a PE unit, based on the instructions (configurations) being used, have been analyzed. The longest paths are dominated by MAC operations at high precision (8b), while critical paths are observed at varied PE locations when low precisions (4b or 1b) are used. To discover the operand timing dependency, case-based static timing analysis method with a commercial EDA tool is used to find out the worst-case timing under certain transitioning conditions. A significant timing dependency on the number of transitioning bits and the transitioning bits' positions is observed. To exploit such a relationship, the summation of the transitioning bits with programmable significance are calculated to guide the dynamic clock management.

Figure 31.3.3 shows details of the adaptive clock management technique. A root PLL feeds the clock to a global DLL, which generates 28 equally delayed phases of clock edges. The 28 phases are sent into 16 clock domains through a global clock bus, travelling a total of about 1.5mm distance. Each phase is generated from one delay stage of the DLL, with a delay step of about 50ps. The clocks for each clock domain are dynamically chosen from the 28 phases of the clock bus in a rotary manner, with a maximum phase offset constrained between neighboring clock domains. The 28 wires of clock bus are carefully matched at layout with dummy clocks at the boundaries, showing up to 4ps mismatch among the nearby phases and 18ps static mismatch from end to end across the long clock routing. The static mismatch across the long clock trace is not critical, as only neighboring clock domains need to be carefully synchronized. To exploit the instruction-based timing variation, tunable clock buffers are implemented for the PE units to rebalance the pipeline timing under different instructions. At each clock domain, a data detection and timing controller (DDTC) module is implemented to dynamically select one clock phase through a glitch-free phase-selection mux based on the compute operands to exploit runtime dynamic timing margin.

Figure 31.3.4 shows the synchronization policy between neighboring clock domains which form an interlocked clock chain with constrained maximum phase offset. Depending on runtime instructions/configurations, a programmable maximum phase offset of up to 0.3ns (or 25% of clock period) among neighboring clocks is set inside the DDTC. Data fetched from the image memory is first passed through a single-stage data buffer. A transition detector, built from XOR circuits, detects the transitioning bits, with their significance summed up and sent to the following selection logic. The significance of each transitioning data bit can be programmed to accommodate the timing margin difference and PVT variations. A small lookup table for the summed significance value is used to determine the target dynamic clock period settings . The phase-selection logic utilizes the target clock period setting, as well as the phase offset information from two neighbors, i.e. North and South neighbors, to decide which phase to use for the next clock cycle. The overall data buffering and DDTC introduces a negligible one clock cycle of latency in the accelerator's execution. For 2D SIMD dataflow, single transition detection is used for the entire row of PE units. For systolic dataflow, because the data travels horizontally, the history of transition results are kept locally and the worst case is selected across 8 previous clock cycles. As a result of the chained operation, if one domain is too fast, it will be locked by a neighboring domain until the neighbors catch up, leading to a wave-like phase propagation. The data signals and the synchronization signals passing across clock domains are carefully managed during timing closure to satisfy the setup or hold requirements at PE boundaries.

A 65nm test chip was built to demonstrate the proposed clock chain scheme. Up to four high-speed phases can be captured in a real-time oscilloscope simultaneously during testing and the phase offsets in measurement ports are calibrated. Clock phases across all the clock domains were repetitively measured to reconstruct the clock propagation map. Fig. 31.3.5 shows the measured color map representing the phase selection at each clock domain along execution cycles. The locking conditions can be observed with a large phase offset between neighboring clock domains. Different neural-network layers in MINIST and CIFAR-10 database have been measured under various bit precisions with up to 19% performance gain or equivalent 34% energy savings using reduced supply voltage. At lower precision, as the timing is more limited by various control paths, the operand-based adaptive operation offers less benefits. 2D SIMD dataflow shows more benefits than the systolic dataflow which needs to consider the worst-case timing within the past eight clock cycles. Fig. 31.3.6 shows the measured performance gain with voltage scaling down to 0.5V and a comparison table with previous adaptive- techniques. This work extends the dynamic-timing-detection scheme to a large 128 PE array accelerator, which is difficult to handle using previous adaptive schemes. A 3.3% area overhead is observed due to the use of DDTC module and additional clock routing. Fig. 31.3.7 shows the die photo with area and power breakdown of sub-modules.

Acknowledgements:
This work was supported in part by the National Science Foundation under grant numbers CCF-1618065. We thank Integrand Software, Inc. for the support of EM simulation tool.

References:

[1] Y. Zhang et al., "iRazor: 3-Transistor Current-Based Error Detection and Correction in an ARM Cortex-R4 Processor," *ISSCC*, pp. 160-161, 2016.
[2] K. Bowman et al., "A 16 nm All-Digital Auto-Calibrating Adaptive Clock Distribution for Supply voltage droop tolerance across a wide operating range," *IEEE JSSC*, vol. 51, no. 1, pp. 8-17, Jan. 2016.
[3] M. Floyd et al., "Adaptive Clocking in the POWER9™ Processor for Voltage Droop Protection," *ISSCC*, pp. 444-445, 2017.
[4] T. Jia et al., "An Adaptive Clock Management Scheme Exploiting Instruction-Based Dynamic Timing Slack for a General-Purpose Graphics Processor Unit with Deep Pipeline and Out-of-Order Execution," *ISSCC*, pp. 318-319, 2019.
[5] B. Moons et al., "ENVISION: A 0.26-to-10TOPS/W Subword-Parallel Dynamic-Voltage-Accuracy-Frequency-Scalable Convolutional Neural Network Processor in 28nm FDSOI", *ISSCC*, pp. 246-247, 2017.
[6] P. Whatmough et al., "A 28nm SoC with a 1.2GHz 568nJ/Prediction Sparse Deep-Neural-Network Engine with >0.1 Timing Error Rate Tolerance for IoT Applications," *ISSCC*, pp. 242-243, 2017.
[7] K. Ueyoshi et al., "QUEST: A 7.49TOPS Multi-Purpose Log-Quantized DNN Inference Engine Stacked on 96MB 3D SRAM Using Inductive-Coupling Technology in 40nm CMOS," *ISSCC*, pp. 216-217, 2018.

 978-1-7281-3205-1/20/$31.00

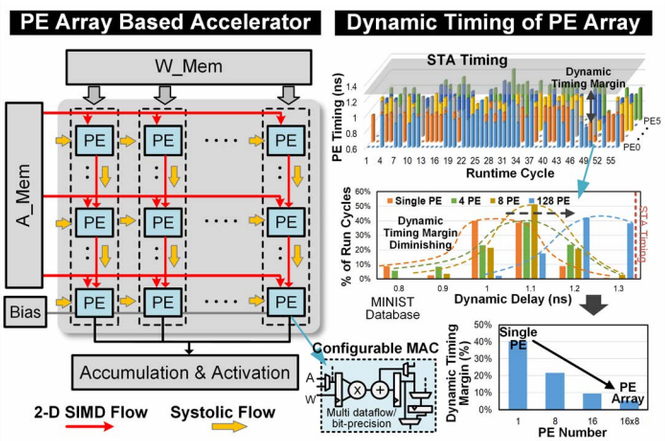


Figure 31.3.1: Commonly adopted PE array-based deep neural network accelerator architecture and the observation of diminishing dynamic timing margin from a single PE unit to a large PE array.

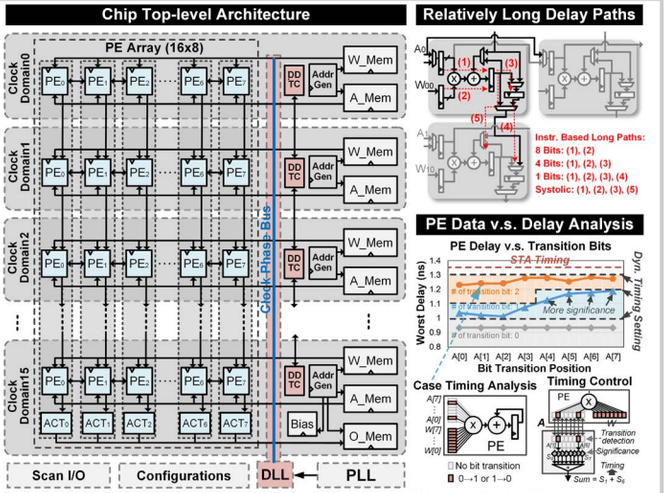


Figure 31.3.2: Chip architecture of this work; the critical paths inside and among PE units; timing dependency on the number of transitioning bits and transitioning bits' positions of PE input values.

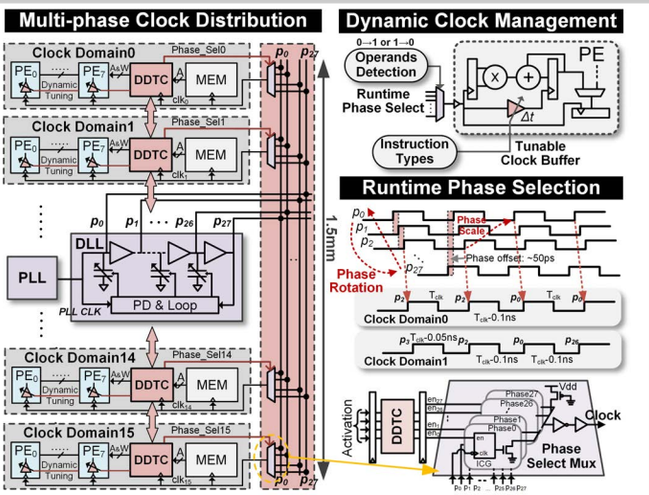


Figure 31.3.3: Multi-phase clock bus design to distribute the clock sources to all the clock domains; the dynamic clock-management scheme for each clock domain.

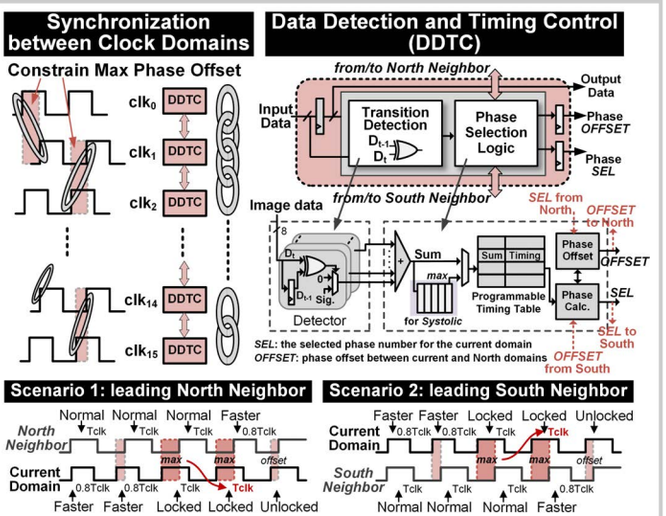


Figure 31.3.4: The clock-chain synchronization policy between the neighboring clock domains and the data detection and timing control module design.

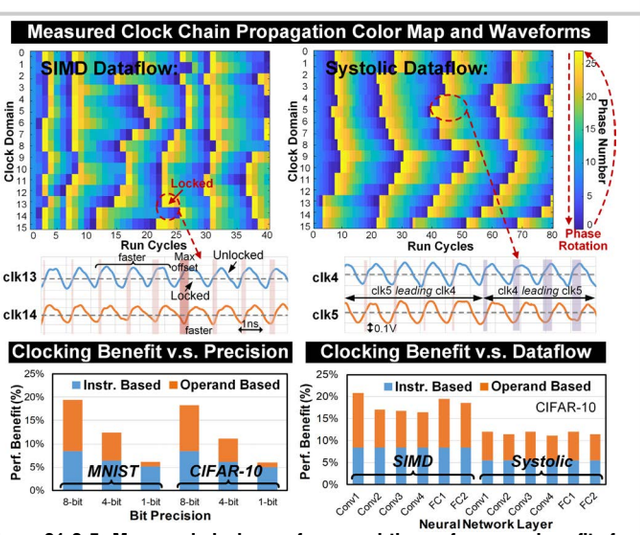
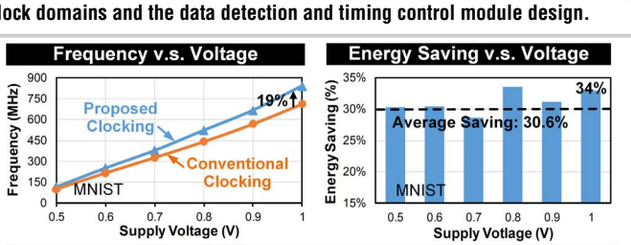


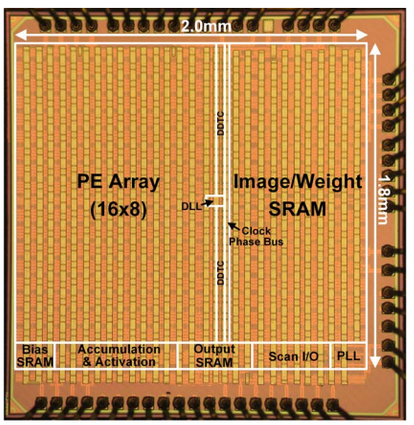
Figure 31.3.5: Measured clock waveforms and the performance benefits for benchmarks.



Comparison with Prior Adaptive Techiques

	JSSC'16 [2]	ISSCC'17 [3]	ISSCC'19 [4]	ISSCC'17 [6]	This work
Process	16nm	14nm	65nm	28nm	**65nm**
Architecture	MAC Unit	CPU cores	GPU cores	DNN Accelerator	**DNN Accelerator**
Adaptive Tech.	Droop Detection	Droop Detection	Instruction Based	Razor Detection	**Instruction and Operand Based**
Datapath	64bit	64bit	32bit	8/16bit	**1/4/8bit**
PE Number	1	4	2	8	**128**
Frequency	2500MHz	4000MHz	417MHz	667MHz	**700MHz**
Voltage	0.9V	not reported	1.0V	0.9V	**1V**
Power	250mW	not reported	468mW	33.7mW	**288mW**
ML Efficiency	N/A	N/A	N/A	1.2TOPs/W	**310GOPs/W (8bit) 2.8TOPs/W (1bit)**
Perf. Improve	13%	3.5%	18.2%	50%	**19%**
Energy Saving	5%	8%	30.4%	30%	**34%**

Figure 31.3.6: Benefits with scaled voltage and comparison table.



Technology	65nm CMOS
Active Area	3.6mm^2
Total Power	288mW
PE Array Power	165mW
SRAM Power	92mW
PLL Power	11.5mW
DLL Power	5.1mW
Clk Bus Power	3.6mW
Base Freq.	700MHz
Boosted Freq.	820MHz
Supply Vdd	0.5-1V
SRAM	84KB
DLL Area	1.6%
Clk Bus Area	1.1%
DDTC Area	2.2%

Figure 31.3.7: Die micrograph.

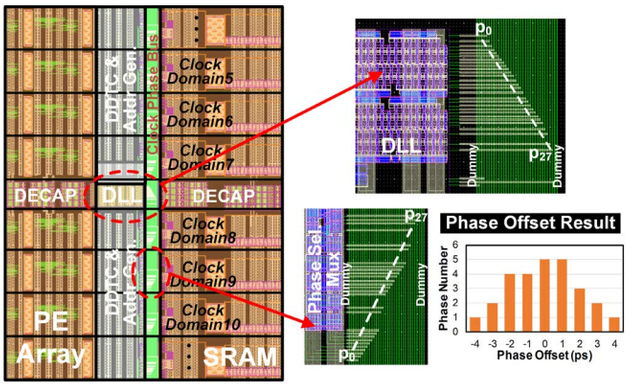


Figure 31.3.S1: The multi-phase clock distribution bus has been carefully designed at layout to balance the total travel length of each clock phase. Dummy phases (extra phases p-1 and p28 generated from DLL) are used to match the boundary conditions for p0 and p27. Both RC extraction and electromagnetic (EM) simulations are utilized to simulate the clock skews between the neighbor phases showing up to 4ps phase mismatch after physical design optimizations.

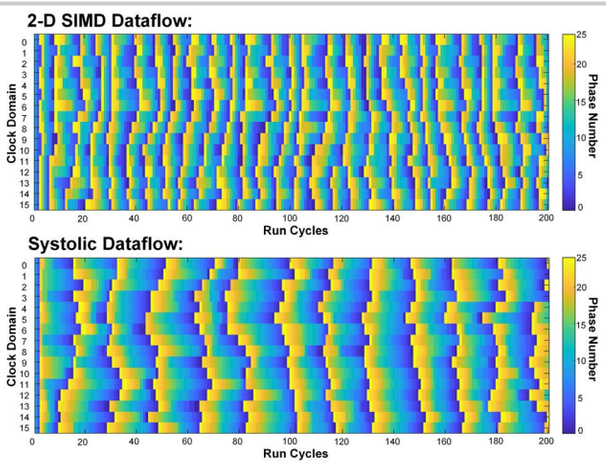


Figure 31.3.S2: The clock phase propagation for different dataflows within 200 cycles. The systolic dataflow has slower phase propagation due to consideration of input operand transitions in the past 8 clock cycles, leading to less performance benefit.

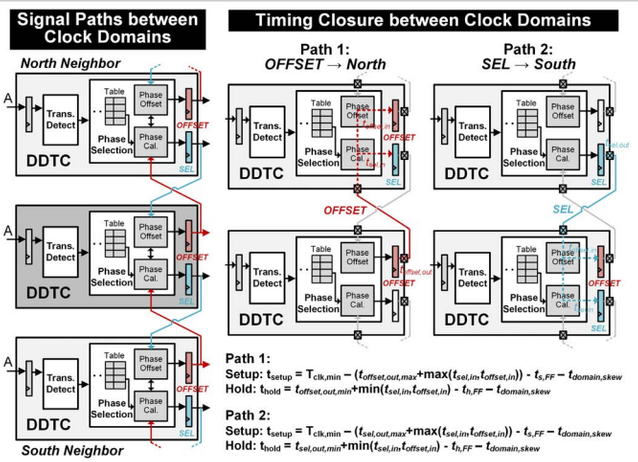


Figure 31.3.S3: The synchronization signals between the neighboring clock domains need to be very carefully managed using STA during timing closure. For example, the setup and hold-time margin for the Offset/Sel signals between DDTCs are shown in the figure.

978-1-7281-3205-1/20/$31.00 ©2020 IEEE