A Mixed-signal Time-Domain Generative Adversarial Network Accelerator with Efficient Subthreshold Time Multiplier and Mixed-signal On-chip Training for Low Power Edge Devices

Zhengyu Chen[1], Sihua Fu[2], Qiankai Cao[1], Jie Gu[1]
[1]Northwestern University, Evanston, IL, USA; [2]Google Inc., Mountain View, CA, USA

Abstract - This work presents a low-cost mixed-signal time-domain accelerator for generative adversarial network (GAN). A significant reduction in hardware cost was achieved through delicate architecture optimization for 8-bit GAN training on edge devices. An area efficient subthreshold time-domain multiplier was designed to eliminate excessive data conversion for mixed-signal computing enabling high throughput mixed-signal online training demonstrated in a 65nm CMOS test chip.

Introduction

GAN is rendered as one of the most interesting and challenging applications in deep learning space. As shown in Fig. 1, GAN contains two deep neural networks (DNN), i.e. a generator and a discriminator, contesting and evolving with each other [1]. Despite its broad real-time applications in gaming, authentication, VR, there is a lack of dedicated low power GAN accelerator due to the tremendous challenges on resource-limited edge devices. From the algorithm aspect, GAN is extremely difficult to train due to model collapses from unbalanced models and high sensitivity to hyper-parameters. From the hardware aspect, GAN involves two DNNs with complex training sequences, e.g. 41 different training stages as in this work. Moreover, the typical floating-point training and complex calculation, e.g. batch normalization and optimizers, are very expensive for a resource-limited edge device [1]. This work, through significant architecture improvement and hardware adaptation, presents a mixed-signal GAN accelerator with 8-bit resolution for cost-effective implementation on edge device. The contributions include: (1) for the first time, a complete GAN training core was implemented on an 8-bit low-power ASIC chip consuming only 39mW; (2) An efficient subthreshold time-domain (TD) multiplier was designed with significant area saving compared to digital design; (3) On-chip training was performed in mixed-signal TD for the first time. The design eliminated 94% overhead from domain conversion, leading to the state-of-art throughput for a mixed-signal based accelerator which normally suffers from slow operation speed.

GAN Accelerator Design with Time-domain Circuits

Fig. 1 shows the implemented GAN architecture with model compression and hardware adaptation techniques used in this work. For fitting with a small chip budget on edge device, we targeted a low-budget architecture implementation of DCGAN [1] using greyscale image with a size of 28 x 28 pixels. The following techniques were specially developed: (1) model balancing and adaptive training were utilized to enable 8-bit training versus conventional floating-point training, leading to a 5X reduction in hardware cost; (2) The challenging and memory consuming operations of batch normalization were simplified by disabling low-impact runtime operations, rendering a 77% removal of the associated operations; (3) The expensive ADAM optimizer was replaced by a succinct momentum stochastic gradient descent optimizer suitable for integer implementation with an 11X reduction of the optimizer's computation; (4) The number of layers and channels were further minimized to reduce the computation load by 6X to 9X. Overall, a 6X reduction of training complexity, a 6.5X hardware cost reduction, and an 11X reduction of on-chip memory were achieved through the algorithm simplification with about a 3% loss of accuracy.

Fig. 2 shows the training sequence. Each training iteration consists of 7 unique phases (e.g. forward prop., loss cal.) with 5 phases for the generator and 4 phases for the discriminator. Each phase also contains 4 to 6 sub-tasks (e.g. Conv, FC, pooling, etc.). To avoid model collapsing, an adaptive training and model strength control scheme was implemented which ceases the training of discriminator if its strength is too high and adaptively increases the magnitude of the gradients during backpropagation. The training sequence is managed by an ASIC training management unit (TMU) shown in Fig. 2. A total of 41 training stages were implemented in the TMU as a finite state machine. Special operations such as pooling, sigmoid, data transpose etc. were handled by the dedicated hardware modules inside the TMU. Register files were used to store temporary weights and feature map outputs, bridging the throughput mismatch between SRAM and MAC arrays.

Fig. 3 shows the test chip architecture diagram including the TMU, a 10x10 time-domain (TD) MAC matrix, SRAM modules and supporting blocks. All the MAC operations of CNN and Transpose-CNN are performed by a TD MAC matrix to improve area and energy efficiency. The time pulses generated from digital-to-time converters (DTC) are processed by the subsequent multiplication, accumulation and activation all in time domain and are finally converted back into digital domain using time-to-digital converters (TDC). A special 16b time-pulse based time-domain accumulator (TD-ACC) is designed using four 4-b ring-based time accumulators [6] with carry propagation to realize accumulation efficiently. With the special TD-ACC, the TDC is only activated once every 25 MAC operations, removing 94% of the time and power overhead from the expensive TDC operations. Pushing all operations in time domain significantly reduces the cross-domain data conversion, rendering a 160X speed-up in MAC operation compared with previous counter-based TD designs [3]. The 8-b TD multiplication is partitioned into four 4-bit multiplications to improve the computation accuracy and speed.

Fig. 4 shows the detailed circuit design featuring a subthreshold (sub-vth) TD multiplier (TD-MUL) and a DTC-based linearization technique. The TD-MUL takes input time pulses and generates output pulses of the multiplication results. As in Fig. 4, the current starving PMOS transistor is pre-biased at subthreshold region and generates a delay equals to the multiplication results through charge accumulation at the gate with logarithmic addition, i.e. a multiplication is addition in log domain. Compared to the digital implementation, the implemented sub-vth multiplier renders a 4.3X reduction of area. However, as shown in simulation, significant nonlinearity is observed in sub-vth multiplication. The nonlinearity is compensated by a logarithmic encoding of DTC. As shown in both equation and the simulated waveforms in Fig. 4, the linearization technique elegantly removes nonlinearity with negligible overhead. After the multiplication, the resulting time pulses are sent into TD-ACC for accumulation of 25 cycles avoiding time-consuming digitalization as [3, 4, 5].

Simple TD ReLU function is also implemented at each CNN layer except the final layer which uses digital sigmoid function.

Measurement

Fig.5 shows the measured linearity from both the TD-MUL and TD-ACC. For the multiplier, although up to 4% error is seen in the result, most of the error is just a small scaling factor shift. Less than 1b error is observed in the TD-ACC design. We trained the GAN with 3 databases, i.e. a digit-MNIST, a fashion, and an emoji database [8-9]. The accuracy of the generated images with conditional GAN from 3 databases shows less than 1% error compared to the ideal integer 8-bit training on CPU and 3% compared with ideal floating-point training (1.6% comes from quantization loss and the rest from process variation of TD circuit). The chip is verified with supply voltages down to 0.7V with up to 5% degradation of accuracy compared with ideal GAN operation. Interestingly, a "self-healing" feature of GAN is observed, recovering most of the error loss from on-chip variations compared with no on-chip training. This intrinsic resiliency presents a merit for training empowered design using mixed-signal circuits. The chip consumes 39mW power with TD-MAC at 90MHz. The total training time of MNIST database takes 4.5 minutes which is 82X less than a high-performance CPU (2.6GHz Intel i7 Quad-core with a power of 197W). The die photo and comparison table with prior analog mixed-signal (AMS) designs are shown in Fig. 6. As most of existing AMS designs suffer from low throughput, this work achieves the highest throughput of 18~5400X [2-5, 7] with similar efficiency. In addition, a low-cost 8-bit on-chip training was realized for AMS design on the very challenging GAN operation.

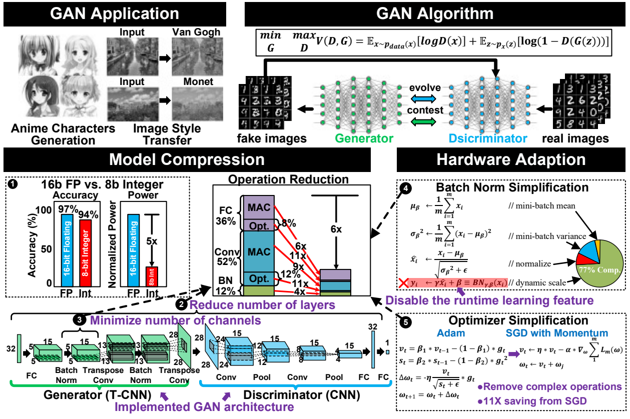


Fig. 1 GAN algorithm and hardware adaption in this work.

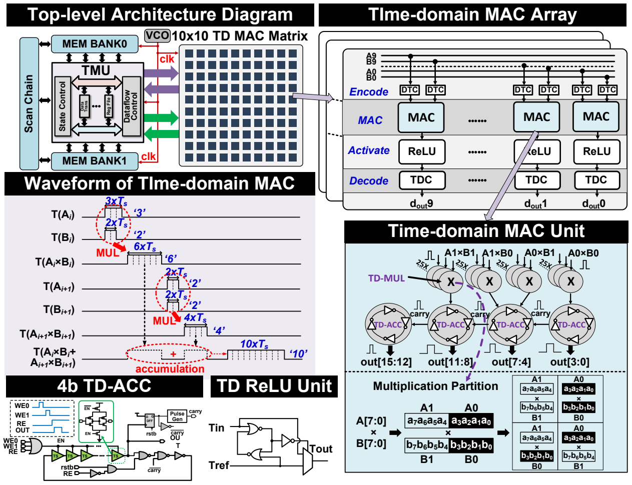


Fig. 3 Top-level architecture diagram with MAC array and TD-Accumulator, TD ReLU circuit, and TD MAC unit.

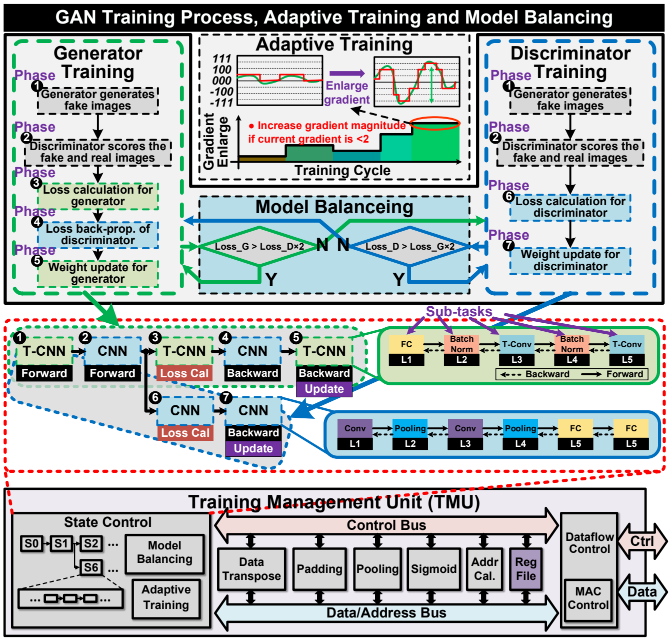

GAN Training Process, Adaptive Training and Model Balancing
Generator Training
Phase 1 Generator generates fake images
Phase 2 Discriminator scores the fake and real images
Phase 3 Loss calculation for generator
Phase 4 Loss back-prop. of discriminator
Phase 5 Weight update for generator
Adaptive Training
Enlarge gradient
Increase gradient magnitude if current gradient is <2
Gradient Enlarge
Training Cycle
Model Balanceing
Loss_G > Loss_D×2
Loss_D > Loss_G×2
N
Y
Discriminator Training
Phase 1 Generator generates fake images
Phase 2 Discriminator scores the fake and real images
Phase 6 Loss calculation for discriminator
Phase 7 Weight update for discriminator
Sub-tasks
T-CNN Forward
CNN Forward
T-CNN Loss Cal
CNN Backward
T-CNN Backward Update
CNN Loss Cal
CNN Backward Update
FC
Batch Norm
T-Conv
Backward
Forward
Conv
Pooling
Training Management Unit (TMU)
State Control
S0 S1 S2 S6
Model Balancing
Adaptive Training
Control Bus
Data Transpose
Padding
Pooling
Sigmoid
Addr Cal.
Reg File
Data/Address Bus
Dataflow Control
MAC Control
Ctrl
Data

Fig. 2 GAN training sequence and ASIC TMU core design.

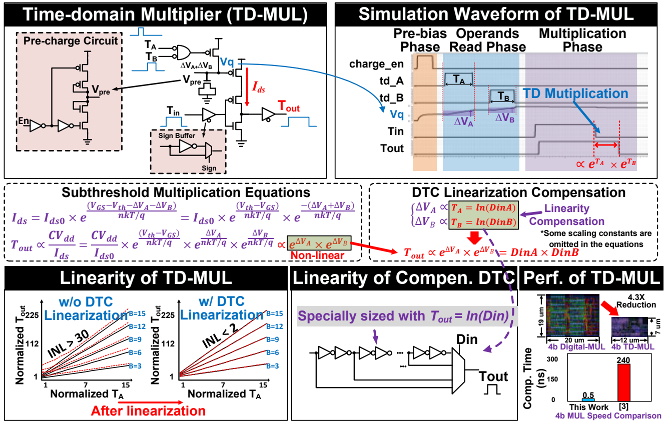


Fig. 4 TD sub-vth multiplier and linearization technique.

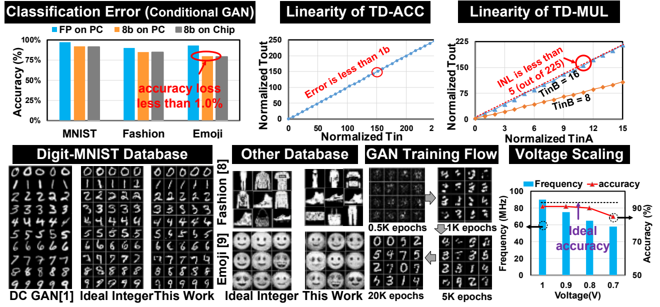
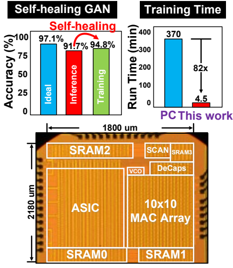


Fig. 5 Measurement results.



	[5] ISSCC 2016	[7] VLSI 2018	[2] CICC 2017	[3] ISSCC 2019	[4] ISSCC 2019	This work
Architecture	Switch Capacitor	TD ASIC	TD ASIC	TD ASIC	TD ASIC	TD ASIC
Application	Gradient Descent	DNN Inference	Image Recog.	Reinforcement Learning	CNN Inference	GAN
Process (nm)	40	28	65	65	40	65
Area (mm^2)	1.44	0.02	0.24	2.0	0.12	3.94
Input/Weight Resolution (bit)	6/3	8/8	1/3	8/8	8/1	8/8
Learning	Offline	Offline	Offline	Online	Offline	Online
Freq. (MHz)	2500	780	99	1.5	25	90
Power (mW)	0.65	0.15	77	0.003	0.03	8 (MAC) 31 (ASIC)
Throughput (GOPS)	1	0.8	0.75	0.0033	0.365	18*
MAC Efficiency (TOPS/W*Bit)	16	112	0.004	1.1	12	18*
On-die SRAM	-	-	0.1 KB	16 KB	No	52 KB

* Accumulation and multiplication are counted as 2 operations.

Fig. 6 Die photo and comparison table.

Acknowledgements: This work is supported by NSF (CCF-1846424).

References [1] A. Radford, et al., *arXiv*, 2015. [2] M. Liu, et. al. *CICC'*17. [3] N. Cao, et al., *ISSCC'*19. [4] A. Sayal, et al., *ISSCC'*19. [5] E. H. Lee, et al., *ISSCC'*16. [6] Z. Chen, et al., *ISSCC'*19. [7] K. Yoshioka, et al., *VLSI'*18. [8] FASHION database, https://www.kaggle.com/zalando-research/fashionmnist. [9] EMOJI database, https://getemoji.com/.