M.S. THESIS

Synthesis of Clock Gating Based on Accurate and Learning Driven Power Analyses

정확하고 학습 기반 전력 분석을 기반으로 하는 클록 게이팅의 합성

BY

Park Sora

FEBURARY 2023

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

Synthesis of Clock Gating Based on Accurate and Learning Driven Power Analyses

정확하고 학습 기반 전력 분석을 기반으로 하는 클록 게이팅의 합성

BY

Park Sora

FEBURARY 2023

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Synthesis of Clock Gating Based on Accurate and Learning Driven Power Analyses

정확하고 학습 기반 전력 분석을 기반으로 하는 클록 게이팅의 합성

지도교수 김 태 환

이 논문을 공학석사 학위논문으로 제출함

2023년 2월

서울대학교 대학원

전기·정보 공학부

박 소 라

박소라의 공학석사 학위 논문을 인준함

2023년 2월

위 원 장: _____

부위원장: _____

위 원: _____

Abstract

In this paper, we introduce two techniques to efficiently apply clock gating in the synthesis stage.

First, We propose a new clock gating methodology based on a precise power saving analysis to overcome the ineffectiveness of the conventional logic structure based clock gating. Two new features exploited in our proposed clock gating are (i) the multiplexer selection signal probability that a flip-flop with multiplexer feedback loop receives a new input and (ii) the joint probability of selection signals that two flip-flops with different multiplexor selection signals both receive new inputs at the same clock cycle. In summary, our method reduces the total power consumption by 2.46% on average (up to 5.00%) over the conventional clock gating method.

In the second work, we address a new problem of transforming the long toggling/untoggling sequences of flip-flops' cycle-accurate activities into short embedding vectors, so that the flip-flop grouping for clock gating is practically feasible in terms of the memory usage and run time for checking activity similarity among flip-flops. To this end, we propose a machine learning based generation of embedding vectors which are accurate enough to predict the original flip-flop toggling sequences. Precisely, we develop a neural network model of LSTM (long short-term memory) based AE(autoencoder) model combined with SDAE (stacked denoising autoencoder) to take into account the time-series (i.e., clock cycle) similarity feature among the toggling sequences, which is essential to determine which flip-flops should be grouped together for clock gating. By integrating (1) our LSTM based embedding vector generation model, we propose two additional ML models for clock gating: (2) joint state probability predictor (JSP) model for generating 0-state probability of two embedding vectors, and (3) joint feature predictor (JFP) model for generating a new embedding vector that combines two embedding vectors. Through experiments, it is confirmed

that our proposed LSTM combined with AutoEnc improves the toggling sequence prediction accuracy up to 0.88 while an LSTM (long short-term memory) based AE model produces accuracy to 0.72, thereby enabling our ML based clock gating framework to save the dynamic power consumption further over that by the state-of-the-art commercial clock gating tool, which relies on the flip-flops' toggling probability for grouping flip-flops. Through experiments with benchmark circuits in IWLS, it is shown that our method is able to reduce the dynamic power by 14.0% on average over that by the conventional toggling-driven clock gating.

keywords: clock gating, flip-flop grouping, low-power design

student number: 2021-25316

Contents

List of Tables

List of Figures

Chapter 1

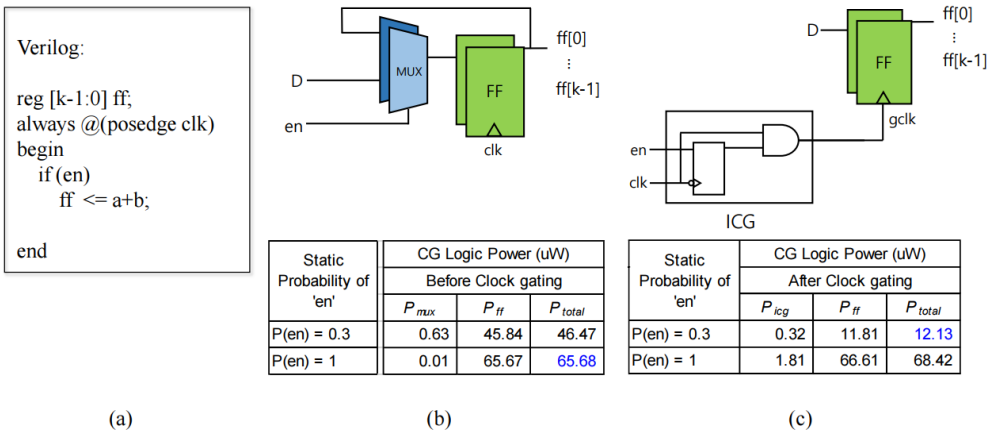
Selective Clock Gating Based on Comprehensive Power Saving Analysis

1.1 Introduction

Clock gating saves dynamic power by shutting off a subtree of clock network during the idle state of the driven logic blocks. This paper proposes a new clock gating methodology based on a precise power saving analysis to overcome the ineffectiveness of the conventional logic structure based clock gating. Two new features exploited in our proposed clock gating are (i) the multiplexer selection signal probability that a flip-flop with multiplexer feedback loop receives a new input and (ii) the joint probability of selection signals that two flip-flops with different multiplexor selection signals both receive new inputs at the same clock cycle.

1.2 Preliminary and Motivation

Figs. 1.1(a) and (b) show a part of Verilog RTL code that commonly appears in the description of design behavior and its synthesized structure, respectively [3], from which it is shown that each of the k flip-flops contains combinational multiplexer logic at its input side. A flop-flop that has a multiplexer-feedback loop at its input side is



Static Probability of 'en'	CG Logic Power (uW) Before Clock gating		
	P_{mux}	P_{ff}	P_{total}
P(en) = 0.3	0.63	45.84	46.47
P(en) = 1	0.01	65.67	65.68

Static Probability of 'en'	CG Logic Power (uW) After Clock gating		
	P_{icg}	P_{ff}	P_{total}
P(en) = 0.3	0.32	11.81	12.13
P(en) = 1	1.81	66.61	68.42

(a) (b) (c)

Figure 1.1: (a) A section of RTL code in Verilog; (b) A synthesized logic for (a) without clock gating; (c) Transformed clock gating structure for (a)

called a self-loop flip-flop. Fig. 1.1(c) shows the transformed logic structure for the circuit in Fig. 1.1(b), produced by applying a conventional clock gating tool where the k multiplexers are completely removed at the cost of allocating a single ICG (integrated clock gating) to enable or disable the clock signal according to the state of the 'en' signal that was used, in Fig. 1.1(a), as a select input to the k multiplexers. The numbers shown in the tables of the initial logic and the transformed logic structures are the amounts of power consumption before and after the application of clock gating to Fig. 1.1(b) with k = 8. Comparing the power consumptions with and without clock gating (CG) when p(en) = 0.3 and p(en) = 1.0 indicates that as p(en) goes to 0, the power saving by CG increases. This means that there may be cases where there is no benefit in terms of power saving even if the clock gating logic is pre-defined in the RTL code by the designer. We focus on the fact that the conventional clock gating of the current commercial tool unconditionally applies clock gating without accurate power analysis based on the switching activity of p(en), and we propose a method of applying selective clock gating. We also propose a method to reduce the dynamic power of the clock tree while ensuring the same logic function by formulating the merge condition

of two different clock enable signals based on accurate power analysis.

1.3 Selective Clock Gating

1.3.1 Concept of Selective Clock Gating

Table 1.1 summarizes the number of self-loop flip-flops in the circuits synthesized from IWLS benchmark code [1, 3]. It is shown that the portion of self-loop flip-flops is 49%∼95% of the total number of flip-flops in circuits, which clearly indicates that applying clock gating to such logic structures may lead to potentially a considerable saving on dynamic power consumption as the value of k increases. Nevertheless, one critically impacting factor that has not been taken into account by the conventional clock gating tools is the behavior of 'en' signal. For a group of self-loop flip-flops like that in Fig. 1.1(b), we define

$$p(en) = the\ probability\ that\ en = True. \tag{1.1}$$

Then, it is obvious that the higher the p(en) value is, the less effective the power saving is since ICG enables clock signal for most time of clock cycles.

Table 1.1: The portion of flip-flops with self-loops in benchmark circuits [1].

Circuit	# of FFs	# of self-loops	% of self-loops
SPI	239	181	75.73%
WB_DMA	587	369	62.86%
AES_CORE	535	263	49.16%
WB_CONMAX	842	656	77.91%
MEM_CTRL	1181	869	73.58%
AC97_CTRL	2326	1691	72.70%
VGA_LCD	17762	16942	95.38%

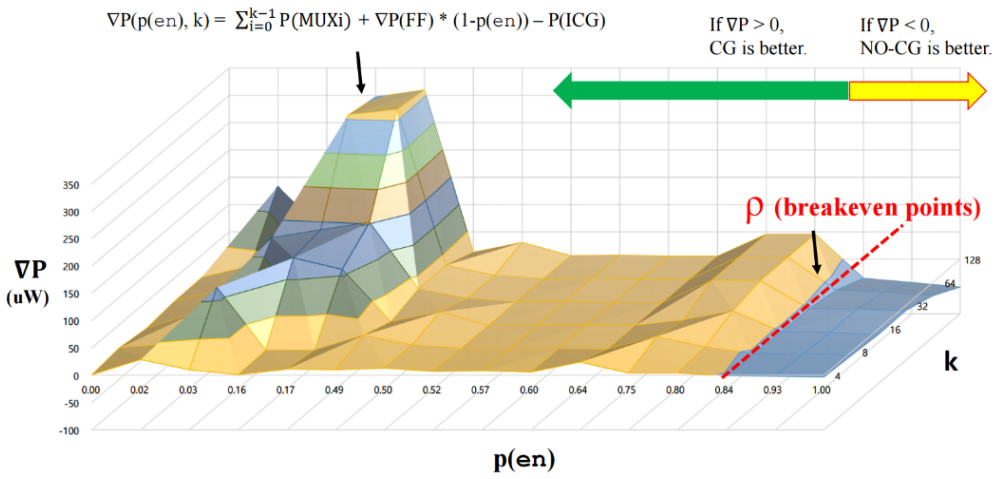


Figure 1.2: The changes of power saving as the k and p(en) values change.

Consequently, a clock gating tool should selectively determine whether implementing a clock gating is beneficial or not according to how much p(en) affects power saving, as illustrated in the power tables in Figs. 1.1(b) and (c). Fig. 1.2 shows the changes of power consumption for the transformed clock gated circuits as the p(en) and k values change for the circuits in Fig. 1.1(b) where the curve marked as ρ corresponds to the collection of the breakeven points of power saving by clock gating. The shape indicates $\nabla P(p(en), k)$ values calculated by Eq.1.2 for pairs of p(en) and k values, in which the breakeven points correspond to the p(en) and k values such that ∇P = 0.

Our objective is to accurately predict the amount of power saving by clock gating for a group of k self-loop flip-flops with p(en) value. Precisely, for a circuit with k self-loop flip-flops and p(en), by constructing a clock gating structure for the circuit, we want to analytically compute the amount of power saving ∇P:

$$\nabla P(p(en), k) = \Sigma_{i=0}^{k-1} P(MUXi) + \nabla P(FF) * (1 - p(en)) - P(ICG) \qquad (1.2)$$

where $P(MUXi)$ and $P(ICG)$ are the amounts of power saved by the removal of

multiplexor at the i th flip-flop and power consumed by the ICG block, respectively, and $\nabla P(FF) * (1 - p(en))$ is the amount of flip-flop power saved by disabling clock signal. Thus, the first two terms in Eq.1.2 indicate the amount of power saved by clock gating while the last term indicates the power overhead induced by clock gating. Based on Eq. 1.2, the condition to implement clock gating is

$$\nabla P(p(en), k) > 0. \tag{1.3}$$

Since the value of k is given from RTL code and the value of p(en) can be estimated through simulation of the code with typical input patterns, we can easily compute the quantity of $\nabla P(p(en), k)$ in Eq.1.2.

1.3.2 Joint probability of selection signals

Furthermore, for multiple groups of flip-flops with self-loops such that each group has different multiplexer selection logic, for example, en1 and en2 as shown in Fig. 1.3, it is possible to merge some of the groups for clock gating if it could result in saving more power. For example, in Fig. 1.3, a set of conditions to implement clock gating for all flip-flops together is

$$\begin{gathered}\nabla P(p(en1 \| en2), k1 + k2) > \nabla P(p(en1), k1) + \nabla P(p(en2), k2), \\ \nabla P(p(en1 \| en2), k1 + k2) > max(\nabla P(p(en1), k1), \nabla P(p(en2), k2)), \\ \nabla P(p(en1 \| en2), k1 + k2) > 0 \;\; if |d_1 - d_2| < D_{th.}\end{gathered} \tag{1.4}$$

where d_1 and d_2 are the center coordinates of the two groups of flip-flops, and D_{th} is the threshold (local) distance given by designer. We have implemented our idea for the clock gating conditions in Eq.1.3 and Eq.1.4 by examining the k and P(en) values, and selectively implemented clock gating structures to maximize the total power saving.

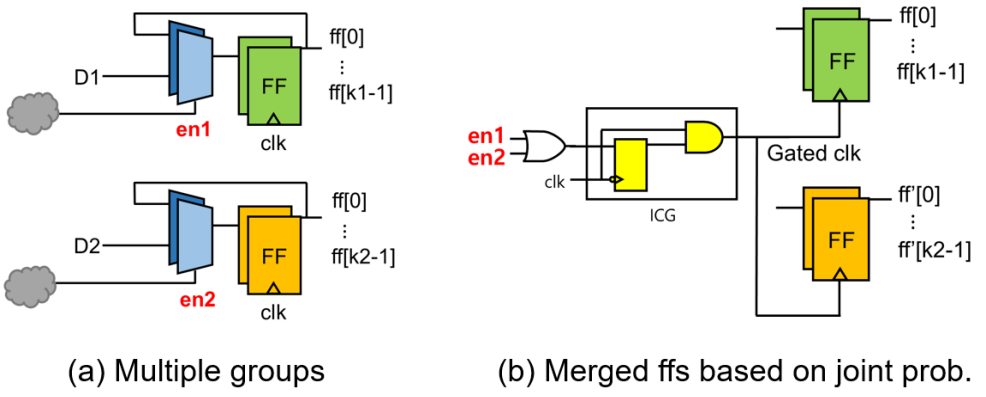


Figure 1.3: Example of merging flip-flops for clock gating based on joint probability of en1 and en2.

1.4 Experimental Results

1.4.1 Experimental Setup

We tested our method and the conventional clock gating method for circuits taken from IWLS benchmarks [1]. The benchmarks were synthesized and physically implemented by using Synopsys Design Compiler. The operating clock frequency was set to 200 MHz for all circuits and we set the initial layout utilization to 70%. We used Nangate 15nm Generic library [4] and a slow PVT corner to guarantee the worst case performance. In addition, for power analysis we performed RTL simulations to get the switching activity information of the benchmark circuits and used PrimeTime PX for power estimation on MUX, ICG, and flip-flops.

We compared our selective clock gating called Selective CG with the existing clock gating method (Conv. CG) as described by the two clock gating flows shown in Fig.1.4. (We used Synopsys Design Complier with option $-clock\_gating$ [2] for Conv. CG.) Our selective clock gating method was implemented as a 1-Pass flow in the process of executing a commercial tool. Conventional CG flow invariably implements clock gating logics, one for each enable signal while our proposed selective CG flow considers the switching activities of the enable signals. In the course of logic synthesis, our CG

flow estimates power saving by computing Eqs.1.2,1.3, and 1.4 with switching activity information. In this step, the amount of power saving is calculated using Eq.1.2 and joint prob Eq.1.4 using the required p(en) obtained from the *Switching Activity Interchange Format* (SAIF) file from RTL simulation. However, since the clock gating logic is already inserted into the netlist, the power value of k multiplexors in Eq.1.2 and the power value of the newly created OR cell in Eq.1.4 for cost calculation are insufficient. In order to calculate the power of cells that do not exist in this way, a virtual cell is created to propagate the toggle rate and 1-Pass flow is introduced to calculate the exact amount of power saving.

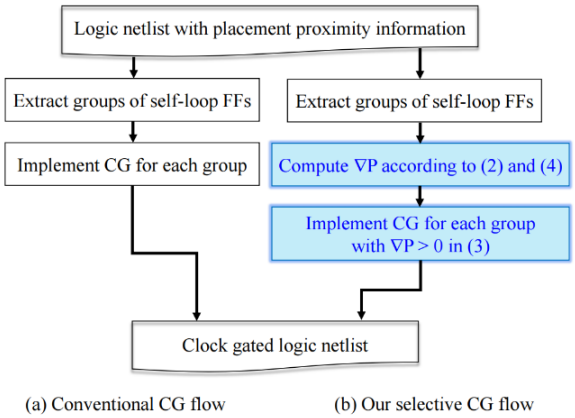


Figure 1.4: Comparison of (a) conventional CG flow [2] and (b) our selective CG flow.

1.4.2 Experimental Result

Tables 1.2 and 1.3 summarize the results produced by the conventional clock gating method provided by a commercial tool and our method. We compare the clock gating ratio (i.e., the ratio of the number of gated flip-flops to the total number of flip-flops) and the number of clock gating cells in Table 1.2 as well as the power consumptions of clock tree (P_{clk}), flip-flops (P_{ff}), combinational logics (P_{combi}), and the total power consumption (P_{total}) in Table 1.3.

Compared with the conventional clock gating, the number of ICG cells is reduced

because unnecessary ICG cells disappeared or merged while applying the selective clock gating method. The following table shows that the total power consumption is reduced even though the clock gating ratio is reduced accordingly.

Table 1.2: Comparison of the ratio of flip-flops for CG and the number of ICGs used by Conv. CG [2] and ours.

Circuit	Conv. CG [2]		Selective CG	
	CG ratio	# of ICGs	CG ratio	# of ICGs
SPI	75.98%	10	69.33%	9
WB_DMA	56.39%	16	54.85%	13
AES_CORE	24.91%	5	24.15%	4
WB_CONMAX	46.94%	24	38.18%	20
MEM_CTRL	74.51%	51	63.19%	43
AC97_CTRL	70.55%	68	70.11%	66
VGA_LCD	98.69%	704	98.46%	701

Compared with the conventional clock gating, the clock power P_{clk} is reduced consistently and effectively by our method for all test cases while there are fluctuations in the flip-flop power P_{ff} due to the load changes on the multiplexers to the corresponding flip-flops. In addition, the combinational logic power P_{combi} increases because more flip-flops are selectively gated by our method, which retains some multiplexers as they are. However, the flip-flop power P_{ff} and clock power P_{clk} dominate the overall power, causing to decrease the total power consumption over the conventional clock gating method. In summary, our method reduces the total power consumption by 2.46% on average (up to 5.00%) over the conventional clock gating method.

Table 1.3: Comparison of power saving for the implementations by our proposed input logic behavior driven (selective) clock gating and the conventional logic structure driven clock gating [2].

Circuit	Conv. CG [2]				Selective CG			
	Power (uW)							
	P_{clk}	P_{ff}	P_{combi}	P_{total}	P_{clk}	P_{ff}	P_{combi}	P_{total} Red.
SPI	47.97	278.44	0.39	326.80	41.21	273.39	1.57	316.17 3.25%
WB_DMA	81.64	583.24	0.06	664.93	56.33	579.61	2.04	637.99 4.05%
AES_CORE	20.40	509.12	17.46	546.98	14.56	509.23	17.48	541.27 1.04%
WB_CONMAX	108.89	1339.66	0.09	1448.64	79.58	1341.61	0.25	1421.43 1.88%
MEM_CTRL	227.47	763.47	0.36	991.30	169.72	771.49	0.52	941.73 5.00%
AC97_CTRL	223.99	386.59	0.30	610.89	212.39	387.67	0.56	600.62 1.68%
VGA_LCD	2525.87	550.71	1877.34	4953.91	2505.21	554.70	1877.37	4937.27 0.34%
Avg. reduction	2.46%							

1.5 Conclusion

We propose a new clock gating methodology based on a precise power saving analysis to overcome the ineffectiveness of the conventional logic structure based clock gating. In the existing conventional clock gating method, clock gating logic was inserted for all flip-flops specified by RTL designer. However, in our selective clock gating method, clock gating was selectively applied by analyzing the power based on the switching activity of the clock gating enable signal. In addition, based on accurate power analysis, a combination of enable signals that can reduce power consumption was selected. Two new features exploited in our proposed clock gating are (i) the multiplexer selection signal probability that a flip-flop with multiplexer feedback loop receives a new input and (ii) the joint probability of selection signals that two flip-flops with different multiplexor selection signals both receive new inputs at the same clock cycle. Experimental results have demonstrated that our approach of selective clock gating offered benefits on reducing the power consumption of designs.

Chapter 2

Machine Learning Based Flip-Flop Grouping for Toggling Driven Clock Gating

2.1 Introduction

For synchronous digital systems, a considerable amount of dynamic power is consumed by the signal on the clock network, up to consuming 70% of total dynamic power consumption in the whole systems [5, 6]. This clock induced power dissipation can be classified into two groups: (*group 1*) the power consumed by the clock delivery network including the storage elements (i.e., flip-flops/latches) and (*group 2*) the power consumed by the combinational logic synchronized by the sequential elements. As a means to reduce the power consumption in group-2, clock gating has been known to be one of the most effective techniques.

Clock gating reduces power by on-and-off a part of clock tree during the idle state of its driven logic blocks or by blocking the clock signal to a flip-flop group during the untoggling state of all flip-flops in that group [7]. This work belongs to the toggling based clock gating.

The basic concept of toggling driven clock gating is to compare the current state of the flip-flop with the data state of the next cycle, and then block the clock signal

supply of the subsystem of the clock network if the data is not toggled (ii.e, the state being maintained as $0 \rightarrow 0$ or $1 \rightarrow 1$.) The structure of the toggling driven clock gating consisting of k flip-flops is shown in Fig. 2.1 where the newly inserted XOR, OR, and ICG (integrated clock gating) [3] cells are marked in blue. Assuming that k flip-flops exist, the Boolean equation of generating clock disable signal *EN* is expressed as k XOR gates and up to $k-1$ 2-input OR gates as follows.

$$EN = (D_1 \oplus Q_1) + (D_2 \oplus Q_2) + ... + (D_k \oplus Q_k) \tag{2.1}$$

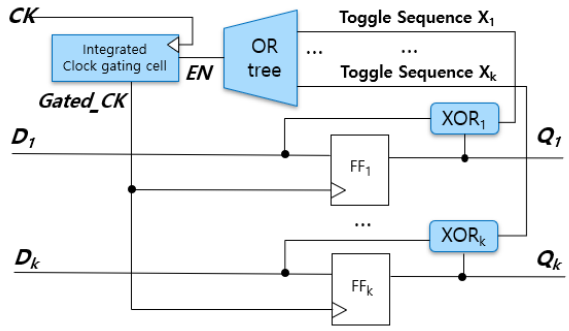


Figure 2.1: Block-level structure of the toggling driven clock gating.

One main issue in the conventional toggling based clock gating method is how the flip-flops in the target design should be grouped so that the clock signal to the flip-flops in each group is collectively enabled and disabled. Here, the key metric to measure the quality of the flip-flop grouping is the amount of occurences of clock cycles at which all the flip-flops in the group are in untoggling state. To accurately measure the metric, a long length of toggling/untoggling simulation sequence of each flip-flop is required. However, the sequence length is practically unacceptable due to the very expensive memory usage and computational time to compare the toggling sequences and produce the union (i.e., bitwise-OR) of the toggling sequences of various combinations of flip-flop groupings. Consequently, in practice, the conventional methods sample the toggling sequences so that the sequence length be in a managable size at the expense of

losing the quality of flip-flop grouping. In the work, we overcome this limitation with the help of machine learning based embedding vector (EV) generation to be used as the representative of the original long toggling sequence of the flip-flops. The key consideration in this work is that the EVs should be fully accurate to predict the original long toggling sequences as well as the length of EVs is short enough to be computationally and memory usage-wise economical.

2.2 Preliminaries and Prior Works

2.2.1 Preliminary and Motivation

Flip-flop's toggling sequence can be obtained through simulation, producing so called *Value Change Dump* (VCD) file, which is mostly so large reaching tens of giga bytes. Thus, it is inefficient to load the VCD file directly a commercial tool provides. Instead, in industry, a simplified one of VCD, called *Switching Activity Interchange Format* (SAIF) is commonly used. Contrary to VCD which records the toggle information of all signals in circuits on every clock cycle, SAIF includes only the percentage of time at which each signal maintains 1-state as well as the total number of toggles of the signal.

The state-of-the-art clock gating tools group flip-flops according to the information in SAIF. However, since SAIF dose not have time-related information, it may miss grouping flip-flops with high similarity on toggling sequence each other. As mentioned previously, the key factor for maximizing power saving by clock gating is to find a group of flip-flops with the close similarity on the toggling patterns among the flip-flops in the group. To this end, we propose a machine learning (ML) based flip-flop grouping which is practically efficient and effective. Specifically, we compress the cycle-accurate toggle sequences extracted from VCD into small-sized *embedding vectors* (EVs), which are convenient to use for making a decision on flip-flop grouping. Our proposed ML based clock gating framework recognizes flip-flop grouping as

a clustering problem in the deep learning field, compresses high-dimensional information into low-dimensional information, and then performs flip-flop grouping based on the similarity between those embedded vectors.

2.2.2 Prior Works

The conventional methods of toggling driven gating are usually based on static probability and toggle rate of individual flip-flops for grouping flip-flops. Although the work in [3] completed the vector-based grouping equation considering physical distance, it is practically infeasible to perform flip-flop grouping due to a vast amount of simulation cycles. The work in [8] proposed a method of flip-flop grouping along with logic optimization using Binary Decision Diagrams (BDDs) [9], but required a large runtime-overhead. The work in [10] combined toggling driven clock gating and multi-bit flip-flop to find the optimal bit according to the toggle rate, but did not focus on the similarity of the toggle sequences. In addition, [11] made part of the clock gating logic into a standard cell, and [12] reduced dynamic power by optimizing the clock gating logic, so flip-flop grouping was not mentioned. On the other hand, the work in [13] attempted grouping flip-flops by generating a correlation matrix using a long toggle sequence, but it requires a lot of memory to support the matrix. The work in [14] also calculated binary pattern based similarity, but did not consider realistic sequence length. Finally, in [15] and [16], machine learning techniques have been applied in the field of early power prediction, but the concept of compressing time-related information was not introduced.

2.3 Machine Learning Based Clock Gating Framework

2.3.1 Primary Model: Embedding Vector Generation

A toggle sequence of tens of thousands of cycles can be thought of as a high-dimensional binary vector. To represent the high-dimensional toggling sequence as a low-dimensional

embedding vector, we applied a stacked autoencoder (SAE) [17]. The stacked autoencoder, illustrated in Fig. 2.2, is a deep neural network that compresses the input sequence (X) and reconstructs the compressed data back to the original input sequence. Each hidden layer is implemented with a denoising autoencoder by applying random corruption to the input data to efficiently restore the outcome of the previous layer. (Refer to [18, 19] for implementing denoising autoencoder with one hidden layer.)

For the learning process of stacked denoising autoencoder (SDAE), the parameters of each hidden layer are learned through layer-wise training that reconstructs the outcome of the previous layer. Subsequently, MSE (mean square error) loss value between the reconstructed input sequence and the original input sequence is propagated to all hidden layers to tune the parameter values.

Once training is done, the decoder is discarded and only the encoder is used to generate the low-dimensional embedding vector. We constructed a hidden layer in [3000, 500, 200, 10, 200, 500, 3000] to reduce the input toggle data of 5000 dimensions to 10 dimensions. The 5000 dimension is the length of the input toggle sequence obtained through various experiments, which is the value that can reduce the dimension at the maximum as long as memory is available in our experimental environment.

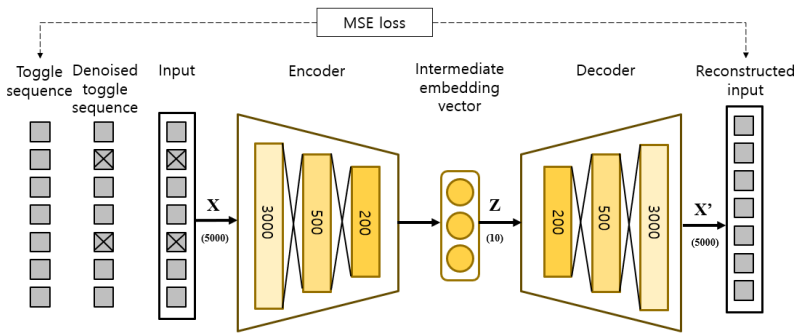


Figure 2.2: Network structure of stacked denoising autoencoder.

Since the actual entire simulation cycle reaches tens to hundreds of thousands, compressing toggling sequence using only SDAE is not sufficient. We want to ob-

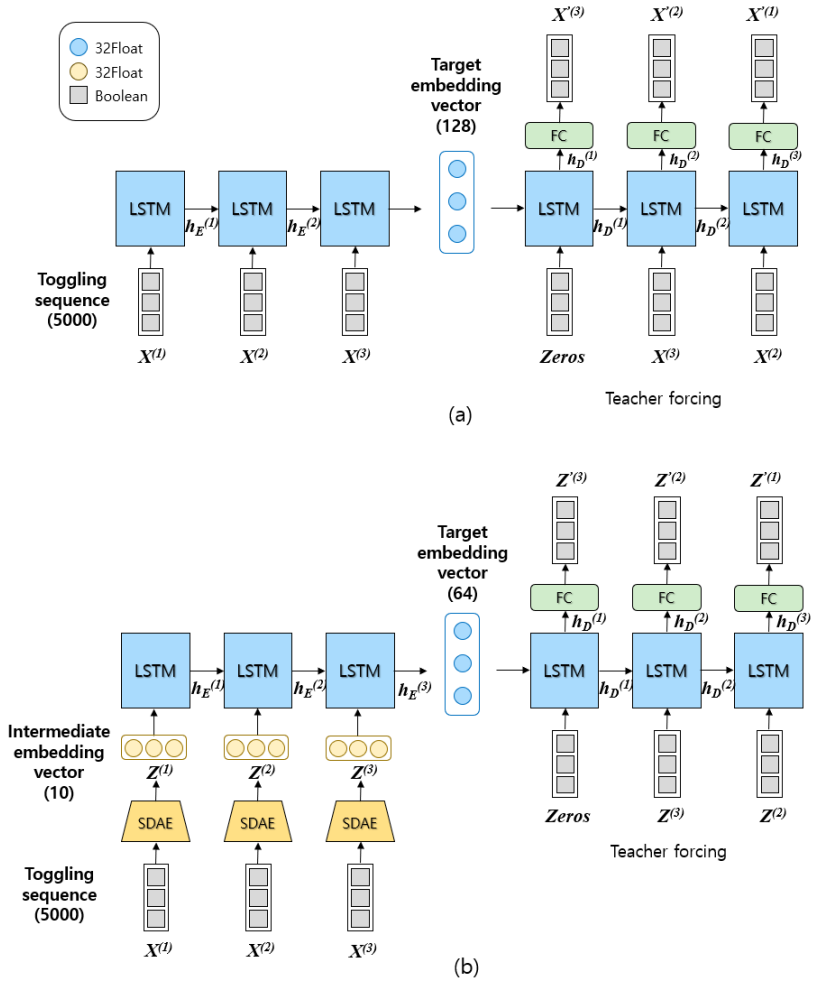


Figure 2.3: (a) The network structure of LSTM-AE. (b) The network structure of our proposed SDAE based LSTM-AE.

tain an embedding vector representing a huge input toggle sequence by introducing an LSTM-based autoencoder (LSTM-AE) suitable for time-serise data expression. The LSTM-AE that can learn sequence consists of a model that directly inputs the toggling sequence, as shown in Fig. 2.3(a), and a 2-step model that inputs through SDAE, as shown in Fig. 2.3(b). The LSTM-AE receives a 5000-dimensional toggle sequence and sequentially learns m sequences to generate a target EV. For the SDAE based LSTM-AE in Fig. 2.3(b), first, 5000-dimensional input sequence is compressed into 10-dimensional through SDAE to generate an *intermediate*-EV. Then, $m/5000$ intermediate EVs pass through the LSTM to learn the reconstruction loss. If the model generates sequence data at the desired level through training, the decoder part is removed and only the encoder part left out for inference.

The LSTM model can be composed of Encoder-Decoder LSTM. It can support variable-length input and output sequences via zero-padding, which makes it suitable to be applied to various simulation cycles.

2.3.2 Secondary Models: Joint State Probability and Joint Feature Prediction

According to the data-driven structure, the input toggling sequences of flip-flops grouped together passes through the OR gate-tree to form a clock EN (Enable) toggle sequence. As the number of zeros in EN toggle sequence increases, all flip-flops in the group can be in inactive state, saving dynamic power on the flip-flops. We tried to predict 0-state probability by implementing with 2-input OR gate as a supervised learning model. First, as shown on the left side in Fig. 2.4, the original signal pattern (D_1, D_2) is converted into an input toggling sequence (X_1, X_2) using custom script. X_1 and X_2 are then passed through the trained SDAE to generate the embedding vectors Z_1 and Z_2.

We devised two additional models to increase the accuracy of flip-flop grouping. The first model, called *Joint State Probability Predictor* (JSP) shown on the upper-

right side in Fig. 2.4, is used for flip-flop grouping, which is a 3-layer fully connected neural network receiving a concatenated value of embedding vectors Z_1 and Z_2, and producing 0-state probability of toggling sequence of X_1 and X_2 as label. If Z_2 with minimal state probability with Z_1 is found through the JSP-based greedy search, actual OR gate insertion will be performed. Then, Z_{12}, which is a newly combined embedding vector, will be created to be used as an input to find the next flip-flop for grouping in a greedy manner.

The second model, called *Joint Feature Predictor* (JFP) shown in the lower-right side in Fig. 2.4, creates an embedding vector corresponding to the bit-wise ORing of X_1 and X_2 when Z_1 and Z_2 enter the model as input.

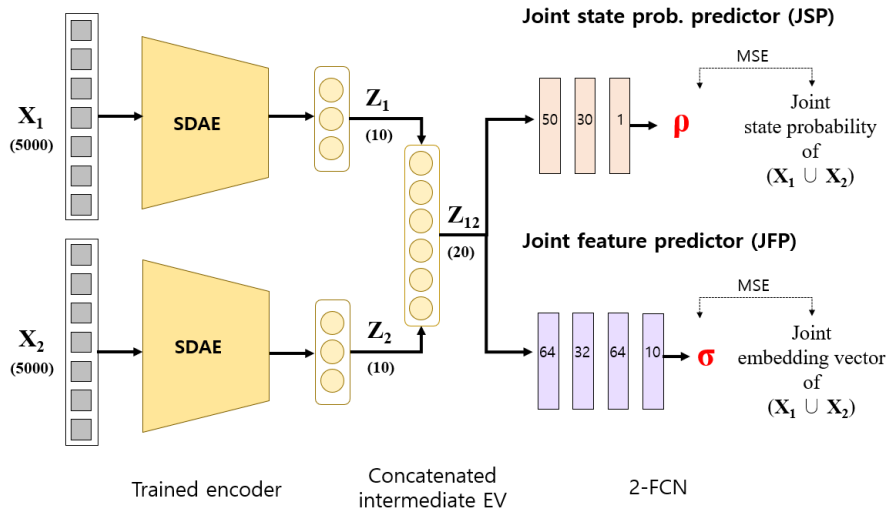


Figure 2.4: Network structure of JSP (joint state probability predictor) and JFP (joint feature predictor).

2.3.3 Distance Analysis Between Embedding Vectors

Euclidean distance between two embedding vectors can be another indicator of similarity. In fact, many image machine learning fields are conducting image classification based on the distance of low-dimensional embedding. We introduced the concept of Euclidean distance during flip-flop grouping, confirming an improvement over that

using the supervised learning model only.

2.3.4 Power Analysis Model

Whenever one additional flip-flop is selected to be included in a group during itertaion process of flip-flop grouping, a stopping condition is required. The stopping condition is (1) the number of flip-flops in the group reaches the pre-determined limit *k*, (2) timing violation occurs, or (3) there is no power saving. For a circuit with k flip-flops, we want to analytically compute the amount of power saving, ∇P, as

$$\nabla P = \sum_{i=1}^{k} P_{ff_i} \cdot p(EN = 0) - \sum_{i=1}^{k} P_{xor_i} - \sum_{j=1}^{k-1} P_{or_j} - P_{icg} \tag{2.2}$$

where P_{xor}, P_{or}, and P_{icg} represent the amounts of power consumed by the XOR-gate, OR-gate and ICG cells, respectively, and $P_{ff_i} \cdot p(EN = 0)$ indicates the amount of flip-flop power saved by disabling clock signal. Thus, the first term in *Eq.2.2* indicates the amount of power saved by clock gating while the last three terms indicate the power overhead induced by clock gating. Based on *Eq.2.2*, the condition to implement clock gating for the current flip-flop grouping is $\nabla P > 0$.

2.3.5 Overall Flow of Flip-flop Grouping

Measuring similarity by comparing toggling sequences of three or more flip-flops at the same time does not guarantee optimal flip-flop grouping. We use a greedy heuristic to select flip-flops sequentially, one by one. Fig. 2.5 shows the flow of our flip-flop grouping, to which a set of deep learning based models at the essential parts are applied. All machine learning models used in the flow perform inference only. Thus, an optimal flip-flop is selected very quickly at each iteration of our greedy heuristic.

2.4 Experimental Results

Our ML based clock gating framework was implemented with PyTorch and was verified using IWLS benchmarks[1], Synopsys *Design compiler* and *Primetime-PX*. We

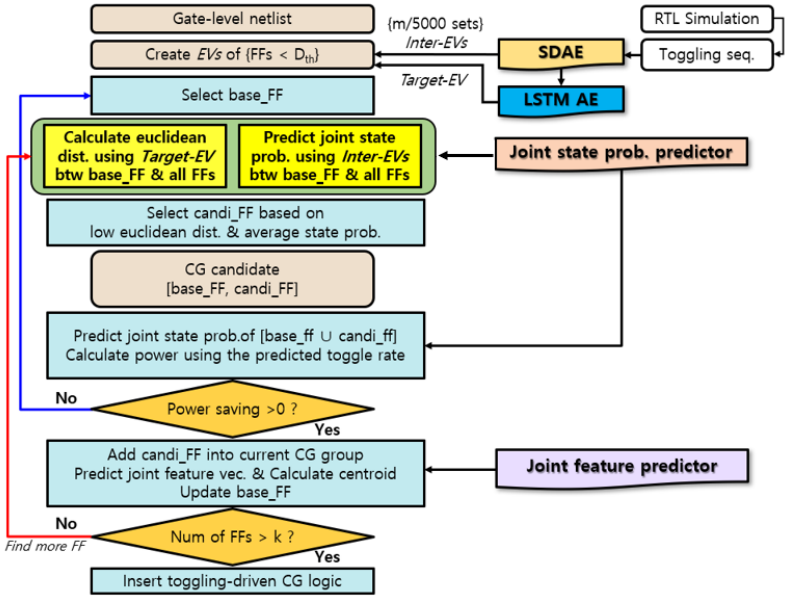


Figure 2.5: Overall flow of our proposed ML based flip-flop grouping for clock gating.

used Nangate 15nm Generic library[20] and a slow PVT corner to guarantee the worst case performance.

Dataset consists of six designs where two designs are used for training and three designs are used for both training and inference. Although the simulation takes a long sequence length, the actual toggling sequence is not much diverse, so, the dataset tends to be sparse. Thus, for sufficient parameter learning, we generate random numbers and used them as additional dataset.

2.4.1 Comparison of Dynamic Power Saving

The dymaic power saved by our proposed ML based clock gating is compared with that by the clock gating of commercial tool (Synopsys Design compiler), and the result is summarized in Table 2.1. Power consumption was measured from gate-level simulation on clock gating logic (i.e., XORs, OR-trees, and ICGs) as well as flip-flops. In comparison with the saving by the commercial tool, P_{ff} (power by flip-flops) is reduced consistently and effectively by our method for all testcases except for WB_DMA

while there is fluctuation on P_{cg} (power by clock gating logic) due to variance on the number of ICGs. However, P_{ff} dominates the overall power, causing to decrease the total power consumption by 14.0% on average over that by the conventional clock gating.

Table 2.1: Comparison of CG logic and flip-flop power of conventional toggling-driven CG and our ML-based toggling-driven CG

Circuit	# of FFs	Datasets bandwidth			Conventional Toggling-driven CG (uW)					ML-based Toggling-driven CG (uW)					Power saving (%)
		Total	Train	Test	CG ratio	# of ICGs	P_{cg}	P_{ff}	P_{total}	CG ratio	# of ICGs	P_{cg}	P_{ff}	P_{total}	
SPI	229	5k	all	-	84.7%	13	75.0	202.6	277.7	90.8%	12	48.5	182.1	230.7	16.9%
AES_CORE	530	5k	all	-	15.1%	5	22.5	1625.5	1647.9	21.1%	7	31.4	1462.8	1494.2	9.3%
WB_CONMAX	770	100k	-	all	70.6%	41	180.8	1238.1	1418.9	81.0%	39	198.1	775.3	973.5	31.4%
MEM_CTRL	1065	100k	0∼50k	50k∼100k	66.4%	53	670.8	1030.7	1701.5	89.3%	61	266.8	1061.2	1327.9	22.0%
AC97_CTRL	2199	100k	0∼50k	50k∼100k	92.1%	256	972.8	1838.9	2811.6	95.6%	269	989.3	1683.9	2673.2	4.9%
WB_DMA	3009	100k	0∼50k	50k∼100k	99.1%	193	1526.9	2691.9	4218.8	98.7%	194	1599.8	2713.6	4313.4	-2.2%
Avg. saving in validation (except for SPI and AES_CORE)															14.0%

2.4.2 Performance of Auto-encoder Reconstruction Model

To compare the reconstruction errors of LSTM-AE and SDAE-based LSTM-AE, cosine similarity was introduced. As indicated in Table 2.2, SDAE-based LSTM AE shows a higher cosine similarity (= 0.88) over that (= 0.72) of LSTM based AE, in which the input dimension of SDAE is determined by sweeping from 1000 to 5000 cycles and checking if the change in loss is acceptable and memory usage is at a feasible level.

2.5 Conclusion

This paper addressed a new problem of transforming the long toggling sequences of flip-flops' cycle-accurate activities into short embedding vectors, so that the flip-flop grouping for clock gating was practically feasible. To this end, we proposed a set of ML model ingredients for clock gating: (1) LSTM based AE model combined with

Table 2.2: Cosine similarity for assessing our LSTM combined with SDAE.

Model	SDAE dim.		LSTM dim.		Cosine
	In	Out	In	Out	Similarity
LSTM AE	-	-	5000	128	0.72
SDAE based LSTM AE	5000	10	10	64	0.88

SDAE for embedding vector generation, (2) joint state probability predictor (JSP) model for generating 0-state probability of two embedding vectors, and (3) joint feature predictor (JFP) model for generating a new embedding vector that combined two embedding vectors.

Bibliography

[1] C. Albrecht, "Iwls 2005 benchmarks," in *International Workshop for Logic Synthesis (IWLS): http://www. iwls. org*, 2005.

[2] Synopsys, "Dc."

[3] S. Wimer and I. Koren, "Design flow for flip-flop grouping in data-driven clock gating," *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 22, no. 4, pp. 771–778, 2013.

[4] M. Martins, J. M. Matos, R. P. Ribas, A. Reis, G. Schlinker, L. Rech, and J. Michelsen, "Open cell library in 15nm freepdk technology," in *Proceedings of the 2015 Symposium on International Symposium on Physical Design*, 2015, pp. 171–178.

[5] D. M. M. Vojin G. Oklobdzija, Vladimir M. Stojanovic and N. M. Nedovic, "Digital system clocking: High-performance and low-power aspects," *IEEE Press, ISBN: 047127447X*, 2003.

[6] S. Jairam, M. Rao, J. Srinivas, P. Vishwanath, H. Udayakumar, and J. C. Rao, "Clock gating for power optimization in asic design cycle theory & practice," in *ACM International Symposium on Low Power Electronic Design*, 2008, pp. 307–308.

[7] M. Müller, S. Simon, H. Gryska, A. Wortmann, and S. Buch, "Low power synthesizable register files for processor and ip cores," *Integration, -The VLSI Journal*, vol. 39, no. 2, pp. 131–155, 2006.

[8] E. Arbel, C. Eisner, and O. Rokhlenko, "Resurrecting infeasible clock-gating functions," in *ACM/IEEE Design Automation Conference*, 2009, pp. 160–165.

[9] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Transactions on Computers*, vol. 100, no. 8, pp. 677–691, 1986.

[10] D. Gluzer and S. Wimer, "Probability-driven multibit flip-flop integration with clock gating," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 3, pp. 1173–1177, 2016.

[11] G. Yang and T. Kim, "Design and algorithm for clock gating and flip-flop co-optimization," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. ACM, 2018, pp. 1–6.

[12] G. Hyun and T. Kim, "Flip-flop state driven clock gating: concept, design, and methodology," in *ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, 2019, pp. 1–6.

[13] Q. Tong and K. Choi, "Activity correlation-based clustering clock-gating technique for digital filters," *International Journal of Electronics*, vol. 104, no. 7, pp. 1095–1106, 2017.

[14] B. Le, D. Maksimovic, D. Sengupta, E. Ergin, R. Berryhill, and A. Veneris, "Constructing stability-based clock gating with hierarchical clustering," in *IEEE International Workshop on Power and Timing Modeling, Optimization and Simulation*, 2015, pp. 97–102.

[15] Y. Zhou, H. Ren, Y. Zhang, B. Keller, B. Khailany, and Z. Zhang, "Primal: Power inference using machine learning," in *ACM/IEEE Design Automation Conference*, 2019, pp. 1–6.

[16] Y. Zhang, H. Ren, and B. Khailany, "Grannite: graph neural network inference for transferable power estimation," in *ACM/IEEE Design Automation Conference*, 2020, pp. 1–6.

[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. 12, 2010.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[20] Nangate, "Nangate freepdk15 open cell library." in *http://www.nangate.com/?page id=2328.*, 2014.

초 록

본 논문에서는 합성 단계에서 클록 게이팅을 효율적으로 적용하기 위한 두 가지 기법을 소개한다.

첫째로, 클록 게이팅 기반의 기존 로직 구조의 비효율성을 극복하기 위해 정밀한 절전 분석을 기반으로 한 새로운 클록 게이팅 방법론을 제안한다. 제안된 클록 게이팅 방법에서 활용되는 두 가지 새로운 기능은 (i) 피드백 루프가 있는 플립플롭의 멀티플렉서 선택 신호 확률 및 (ii) 서로 다른 멀티플렉서 선택 신호를 갖는 두 플립플롭의 멀티플렉서 선택 신호 결합 확률이다. 전력 이득이 있는 경우에만 클록 게이팅을 적용하고 서로 다른 클록 게이팅 그룹을 통합함으로서 전체 동적 전력를 줄이고자 하였다. 실험을 통해 기존의 클록 게이팅 방법에 비해 평균 2.46%(최대 5.00%)의 총 전력 소비를 줄이는 것을 확인하였다.

두 번째로 플립플롭의 클록 주기별 상태를 나타내는 긴 토글링/언토글링 시퀀스를 짧은 임베딩 벡터로 변환하는 문제를 해결하였다. 이를 토글링 기반 클록 게이팅을 위한 플립플롭 그룹화에 적용하여 플립플롭 간의 상태 유사성 확인이 메모리 사용량 및 실행 시간 측면에서 실질적으로 실현 가능하게 하였다. 이를 위해 기계학습 기반으로 원래의 플립플롭 토글 시퀀스를 예측하기에 충분히 정확한 저차원의 임베딩 벡터의 생성을 제안한다. 우리는 토글링 시퀀스 간의 시계열 유사성을 고려하기 위해 디노이즈 오토인코더를 이용하여 5000 클록 사이클의 토글링 시퀀스를 10차원으로 압축하고 이를 장단기 메모리 오토인코더에 입력하여 전체 시퀀스를 대변하는 저차원 임베딩 벡터를 생성하는 신경망 모델을 개발하였다. 또한 우리는 클록 게이팅을 위한 두 가지 부가적인 신경망 모델인 (1) 2개의 임베딩 벡터의 0-상태 확률 생성을 위한 결합 확률 예측 모델과 (2) 두 개의 임베딩 벡터를 결합하여

새로운 임베딩 벡터를 예측하는 결합 특징 예측 모델을 제안한다. IWLS 벤치마크 회로를 이용한 실험을 통해, 디노이즈 오토인코더만 사용했을때보다 장단기 메모리 기반의 오토인코더를 결합했을 때 입력 데이터를 복원 정확도가 더 우수한 것을 확인하였다. 또한 우리의 방법이 기존의 토글링 기반 클록 게이팅에 비해 평균 14.0% 의 동적 전력을 줄일 수 있음을 확인하였다.

주요어: 클록게이팅, 플립플롭 그룹핑, 논리합성
학번: 2021-25316