A Dynamic Timing Error Avoidance Technique Using Prediction Logic in High-Performance Designs

Mehrnaz Ahmadi, Sahand Salamat, and Bijan Alizadeh

***Abstract*— Time borrowing techniques have been widely used to mitigate the timing errors in high-performance designs. A new dynamic flip-flop conversion technique is introduced by Ahmadi *et al.* (2015) which dynamically converts flip-flops into transparent latches to grant the time borrowing from the next stage and prevent setup time violation. However, it is not able to prevent the timing violation in the successive critical path (SCP) and critical feedback path (CFP) structures. In this brief, we introduce a novel idea of using the output of fast prediction logic of the critical path along with dynamic clock stretching in SCP and CFP structures. The results show that our technique, on average, is able to improve the performance by 20.2% and 14.8% during the prelayout and postlayout simulations, respectively. Furthermore, the proposed technique is almost 7.7% more effective in terms of the performance improvement with only 0.1% area overhead in comparison with the best existing technique.**

***Index Terms*— Dynamic clock stretching, high-performance design, prediction logic, setup time violation, time borrowing.**

I. Introduction

The demand for high-performance design has been significantly increasing over the past few years. Traditionally, the maximum tolerable frequency, in which a circuit works correctly, is computed based on the delay of the longest paths (called critical paths) in the circuit. In recent years, several methods have been investigated to increase the performance of the design. In [3], data are captured by a shadow latch with a delayed clock signal, as well as by the main flip-flop. In the case of timing violation, the system corrects the propagated error by halting the next stages for a cycle. In variable latency (V.L.) designs, the period of the clock is set to T. Then, in the case of activating a path with the latency of $T + \Delta T$, an extra clock cycle is required [4].

Another solution is time borrowing technique in which the clock period of the circuit, i.e., T_1, is set to a smaller value than the maximum delay of the critical path, i.e., $T_1 + \Delta T$. Whenever a path having a larger delay than T_1 is activated, the extra ΔT time is borrowed from its successive stages. In [5], latches were used to perform time borrowing during the high phase of the clock cycle. In [6], soft-edge flip-flops (SEFFs), having a small window of transparency instead of a hard edge, have been used. In [7], a pulsed-latch augmented with an additional circuit to delay the clock signal over multiple cycles in case of time borrowing.

In [1] and [8], a dynamic flip-flop conversion (DFFC) technique is introduced. In this technique, the timing violation is predicted by a timing violation predictor (TVP) block which detects a transition at the midpoint (the node that cuts the path into half in terms of delay) during the second half of the clock period. In the case of detecting a late data at the midpoint and predicting error, the Err signal is issued by the TVP block. Then, a data arrival detector (DAD) block toggles the Conv signal, and the flip-flop operates as a transparent latch. When the late data arrive at the input of the critical flip-flop, it is captured. Then, the Conv signal toggles again and the critical flip-flop goes back to its normal operation [see Fig. 1(a)–(c)].

In [1], it is discussed that DFFC is not able to prevent the timing errors in problematic path structures. These paths include critical paths with short sequential depth, critical feedback path (CFP)—a feedback path which is also a critical path—and successive critical path (SCP)—a critical path followed by another critical path in successive sequential stages. In [2], a hybrid technique is proposed which uses the DFFC of [1] in critical paths. Whenever a problematic path is activated, the clock is, dynamically, stretched for half of the clock cycle using the clock shifter of Fig. 1(d). This technique, however, suffers from the large number of clock stretching.

In order to alleviate this problem, in this brief, we propose a dynamic timing error avoidance (DTEA) technique which first tries to prevent timing violation using DFFC; if it is not possible, the output of a faster logic (prediction logic) along with dynamic clock stretching and time borrowing is used to prevent timing violation. The rest of this brief is organized as follows. Section II presents our proposed DTEA technique. In Section III, the experimental results are reported. Finally, Section IV concludes this brief.

II. Proposed Dynamic Timing Error Avoidance

The DFFC technique [1] is not able to prevent timing violation in problematic path structures. In our proposed DTEA technique, first, the static timing analysis is performed, and the paths are classified into the problematic (SCPs and CFPs) and nonproblematic structures. Since DFFC [1] is able to improve the performance of the nonproblematic structures efficiently, in our DTEA technique, these structures are equipped with DFFC blocks of Fig. 1. Nevertheless, in problematic structures, cone of the critical paths (exact cones) are predicted by a fast prediction logic. Then, these logics, along with the exact cones and some additional blocks are inserted into the circuit to guarantee the fault-free high-performance operation. The prediction logic has a smaller delay than that of the cone of the critical path and its output is equal to that of the critical path for most of its input vectors (hit-predict). In this case, the circuit uses the early output of the prediction logic instead of the late output of the exact cone and generates its outputs at a high speed. However, for C_t number of inputs, the prediction logic produces the output values which are not equal to those of the exact cone (miss-predict). In this case, the clock is stretched and the output of the exact critical path overrides the invalid data from the prediction logic. Fig. 2 depicts a circuit that is augmented with the DTEA technique along with the necessary blocks used for clock stretching mechanism. The clock shifter block of Fig. 1 ("clock shifter") is used to stretch the system clock (i.e., CLK). All the flag signals from problematic structures are gathered into "flag collector" which is a NAND gate tree. If any flag falls down due to

Manuscript received July 31, 2018; revised October 6, 2018; accepted November 7, 2018. *(Corresponding author: Bijan Alizadeh.)*

M. Ahmadi and S. Salamat are with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 1417614418, Iran (e-mail: m.ahmady@ut.ac.ir; s.salamat@ut.ac.ir).

B. Alizadeh is with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 1417614418, Iran, and also with the School of Computer Science, Institute for Research in Fundamental Sciences, Tehran 19538-33511, Iran (e-mail: b.alizadeh@ut.ac.ir).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2018.2881173

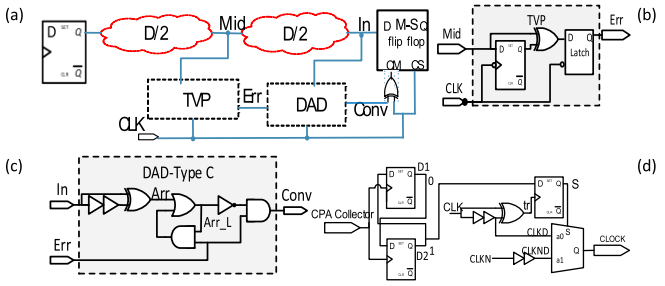

Fig. 1. (a) DFFC microarchitecture. (b) TVP block. (c) DAD block [1]. (d) Clock shifter block [2].

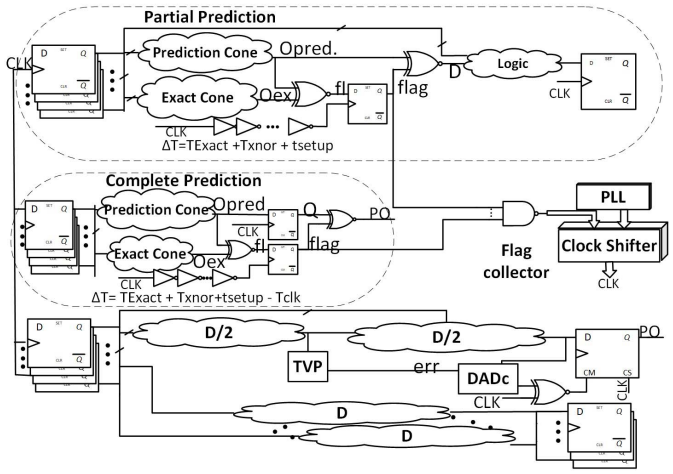

Fig. 2. Demonstration of applying our technique to a complete circuit.

miss-predict, the output of this block rises and the "clock shifter" block stretches the clock of the system for half of a cycle.

Definition-Error Rate of jth Path (ER_j): The error rate of the jth prediction cone (ER_j) which has #coneinput inputs and $C_{\mathrm{t}j}$ number of miss-predict outputs equals $C_{\mathrm{t}j}/2^{\#\mathrm{coneinput}}$.

A. Circuit Prediction

To generate the prediction cone, we first compare its output to zero or one logic as it has zero overhead. If this leads to smaller ER than that of the $\mathrm{ER}_{\mathrm{maxlimit}}$ (the maximum allowed ER of the logic) and then it is selected. Otherwise, the approximation technique of [9] which is based on do not caring the minterms to achieve the best approximate circuit with the shortest delay is used. First, C_{t} is set to 1 so that the ER equals its minimum possible value. If the delay of the approximate cone using the approximation technique of [9] ($\mathrm{Delay}_{\mathrm{ac}}$) is less than that of the expected delay of the approximate path, it is selected as prediction cone. Otherwise, C_{t} and thus, ER is increased, and the above steps are repeated until ER exceeds the $\mathrm{ER}_{\mathrm{maxlimit}}$. In this case, no logic is found under ER and delay constraints.

B. Complete Critical Cone Prediction

In the case of complete cone prediction, the behavior of the critical path cone (exact cone) is predicted by the prediction cone. As illustrated in Fig. 2, in high-performance operation of the circuit (hit-predict case), the prediction cone generates the correct output (O_{pred}) before the rising edge of the clock and its value is stored in the critical flip-flop (Q). An fl signal is generated by comparing the output of the prediction cone to that of the exact cone (Oex signal) using an XNOR gate.

To compare the outputs of exact and prediction cones and prevent glitches in the design, it is necessary to store the fl value as soon as Oex is ready (in the second cycle). Therefore, the CLK signal is delayed for $\Delta T = T_{\mathrm{Exact}} + T\,\mathrm{XNOR} + t_{\mathrm{setup}} - \mathrm{Tclk}$ (in which T_{Exact} is the worst case delay of exact cone and Tclk is the period of CLK) to store fl at the rising edge of delayed clock signal (CLKd). At Tclk $+\Delta T$ not only the outputs of both prediction and exact cones are ready but they are also compared and the result of comparison is ready. When O_{pred} and Oex are equal (hit-predict), the flag remains one and PO (the input of the next stage) is equal to Q. In the case of a miss-predict, the flag falls down and the output of the flip-flop (Q) is toggled using an XNOR gate. The value of primary output PO is valid in $\mathrm{Delay}_{\mathrm{PO}} = T_{\mathrm{Exact}} + t_{\mathrm{setup}} + t_{\mathrm{cq}} + 2 \times T_{\mathrm{Xnor}}$. The flag signal is used to stretch the system clock in the second cycle and compensate for the extra time ($\mathrm{Delay}_{\mathrm{PO}} - \mathrm{Tclk}$) required for the valid value on PO.

C. Partial Critical Cone Prediction

Large area overhead is obviously the major impediment of complete critical cone prediction, especially when the size of the critical cone is relatively large in comparison with the complete circuit. In addition, in fan-out path structures, the output of joint parts of the branches must be predicted several times which exacerbates the issue. Partial prediction of the critical cone is our solution for this problem. As depicted in Fig. 2, the output of prediction cone (O_{pred}) is XNORed with the output of exact cone (Oex) to generate an fl signal. To compare the exact and prediction cones and prevent glitches in the design, it is necessary to store the fl value after Oex is ready. Therefore, to save its value the clock of the register must be delayed for $\Delta T = T_{\mathrm{Exact}} + T_{\mathrm{Xnor}} + t_{\mathrm{setup}}$. The fl signal is stored in flag signal which will be used later to stretch the clock signal whenever $O_{\mathrm{pred}} \neq O$ex. In normal operation of the circuit, hit-predict cases, $O_{\mathrm{pred}} = O$ex. The output of XNOR gate (fl) and flip-flop (flag) are one and D signal (the signal fed to the subsequent logics instead of the output of the exact cone) equals O_{pred}. Consequently, the correct value of O_{pred} is valid on D at $T_{\mathrm{Ap}}+T_{\mathrm{Xnor}}$, where T_{Ap} is the delay of the prediction logic path. In case of miss-predict, $\overline{Oap} = O$ex and the flag signal falls down at $\Delta T = T_{\mathrm{Exact}} + T_{\mathrm{Xnor}} + t_{\mathrm{setup}} + t_{\mathrm{cq}}$ which is then, used by the clock shifter to stretch the clock at the very clock cycle. After the D signal is toggled and its valid data is ready, the late data arrive at the flip-flop before the sampling edge of the clock and stores in critical flip-flop.

D. DTEA Utilization Flow

At the beginning, the designer sets a value for the total tolerable ER of the circuit ($\mathrm{ER}_{\mathrm{circuit}}$) according to the required design performance. This value stands for the total maximum allowable number of clock stretching during the complete gate-level simulation and is caused by occurring the mismatches between the prediction cones and the exact critical cones. Next, a list of critical paths with a larger delay than the desired clock period (Tclk) is generated and their cones are extracted and stored. If these paths are problematic they are determined. Next, the maximum ER for each problematic critical path ($\mathrm{ER}_{\mathrm{maxlimit}}$) is set by dividing $\mathrm{ER}_{\mathrm{circuit}}$ by a total number of problematic paths. Also, $\mathrm{ER}_{\mathrm{maxlimit}}$ (maximum allowable error rate of the prediction cone) equal to $(1/\mathrm{problematic\_path}) \times \mathrm{ER}_{\mathrm{circuit}}$ (line 6).

In the case of problematic structure, first, the whole critical cone is predicted completely. If no prediction cone is founded under $\mathrm{ER}_{\mathrm{maxlimit}}$ and expected delay (determined by clock frequency) constraints, the partial prediction is applied. Then, the corresponding blocks to guarantee the fault-free operation of the high-performance circuit are inserted. Otherwise, the DFFC blocks of Fig. 1 are applied

TABLE I

PRELAYOUT PERFORMANCE COMPARISON BETWEEN DFFC [1], SEFF [12], V.L. [4], HYBRID [2], AND OUR DTEA TECHNIQUES

	DUT	DFFC [1]	SEFF [12]	V.L. [4]	Hybrid [2]	DTEA
Performance Improvement %	s15850	9.2	15.0	19.5	25.8	25.3
	s38417	9.5	7.3	0.0	14.0	14.0
	s38584	8.3	3.4	18.5	0.0	18.0
	b12	8.8	3.1	0.0	17.6	17.6
	b15	5.1	0.0	15.2	5.1	26.1
Average %		9.3	5.8	10.6	12.5	20.2
Area Overhead%	s15850	2.5	2.1	6.4	4.5	5.0
	s38417	5.8	5.3	N.A	4.9	5.0
	s38584	5.5	4.3	4.5	5.5	4.8
	b12	24.4	9.5	N.A	26.4	25.5
	b15	5.6	N.A	6.2	6.1	7.4
Average %		8.8	5.3	5.7	9.4	9.5

TABLE II

POSTLAYOUT PERFORMANCE COMPARISON BETWEEN HYBRID [2] AND OUR DTEA TECHNIQUE

circuit	O.F. (GHz)	% Freq.Imp		#stretch		%Improve		CPU(s)		Avg. E.R.
		Hybrid	DTEA	Hybrid	DTEA	Hybrid	DTEA	Hybrid	DTEA	
b03	1.66	36.7	46.4	400	100	0.28	28.31	12	57	0.28
b05	1.19	7.5	10.1	36	6	0.05	8.92	29	65	0.05
b11	1.16	22.4	22.4	109	52	0.08	17.41	25	68	0.08
b12	1.25	12.0	17.6	6	10	0.06	14.53	33	76	0.06
b15	0.93	7.5	11.8	82	25	0.03	9.28	36	78	0.03
s5378	1.90	5.2	10.5	69	50	0.11	7.17	40	125	0.11
s38584	0.33	56.0	43.7	445	147	0.02	22.23	55	141	0.02
s38417	1.33	12.7	12.7	23	18	0.01	10.16	49	89	0.01
Avg.	1.25	14.6	21.4	146	51	0.08	14.80	34.9	87.4	0.08

TABLE III

POSTLAYOUT POWER AND AREA OVERHEAD COMPARISON BETWEEN HYBRID [2] AND DTEA TECHNIQUES

circuit	% Power increase			%Area Overhead		#Criticals			#HTV buffer
	Hybrid	DTEA Slow	DTEA Fast	Hybrid	DTEA	Total	Partial	Comp.	
b11	63.4	64.3	66.1	11.5	12.3	4	1	0	5
b12	57.0	38.7	57.1	24.7	23.8	9	0	1	3
b15	15.4	15.3	19.9	2.9	2.9	3	0	1	5
s5378	25.5	27.2	30.5	12.5	16.1	5	2	0	11
s38584	33.6	14.3	41.7	4.7	6.0	7	0	2	11
s38417	23.5	23.9	23.9	4.0	3.9	5	0	1	14
Avg.	36.4	30.6	39.8	10.0	10.8	5.5	0.5	0.8	8.2

to the critical path. Next, in the case of satisfying the area and power constraints that are set by the designer, the above stages are applied to the next longest critical path in LOC. Otherwise, the desired clock period is not achievable.

III. EXPERIMENTAL RESULTS

In this section, we evaluate our DTEA technique by applying it to the complete gate-level benchmark circuits from ISCAS'89 and ITC'99 [10]. The benchmark circuits are timing optimized during the synthesis. Usually, a large number of critical and near-critical paths exist after optimization, which is referred to as "timing wall" and causes large area overhead in the high-performance techniques. To ameliorate the effect of timing wall issue, the "critical_range" command in the synopsys design compiler [11] tool which reduces the number of near-critical paths is used during the gate-level synthesis of the baseline circuits. This command imposes some overhead which is included in the reported area and power overhead.

A. Performance Analysis

In this section, we evaluate and compare the performance improvement and the area overhead of our DTEA technique as well as the state-of-the-art techniques of DFFC [1], SEFF [12], hybrid [2], and V.L. [4]. To do so, five of the largest benchmark circuits from ISCAS'89 and ITC'99 were synthesized into gate-level netlist using the delay information of lib2.genlib library so that the timing characteristics resemble the real implementation timing. The performance of a technique is calculated based on the time required to finish a task which is affected by both operational frequency (MAF) and the number of clock stretching. The performance improvement and the area overhead of five techniques are reported in Table I. Considering all benchmarks, our proposed DTEA technique, on average, shows 7.7% and 9.6% performance improvement at the expense of only 0.1% and 3.8% area overhead in comparison with hybrid and V.L. [4], respectively.

In another experiment, we evaluate the performance of benchmark circuits using the hybrid and DTEA techniques. The RTL codes of the benchmark circuits were synthesized using the Nangate 45-nm library [13]. To consider the effects of interconnects, clock tree, and so forth we performed the postlayout simulation. Table II compares the performance improvement of DTEA and hybrid techniques [2]. In Table II, the original frequency of the circuits is shown in the second column. The frequency improvement, number of clock stretching in 1000 cycles on average, performance improvement and the CPU time of applying high-performance techniques are reported in % Freq.Imp, #stretch, %Improve, and CPU(s) major columns for both techniques. The average E.R column presents the average error rate of the prediction cones in each benchmark. The results show that after inserting the clock tree, our DTEA technique has 5.4% performance improvement over the hybrid technique. Considering b03, b15, and s38584 benchmarks in which the number of clock stretching is reduced dramatically using DTEA, the performance of benchmark circuits is 11.46%, and 19.93% higher than that of the circuits adopting hybrid technique, and original circuits with no time borrowing on average, respectively.

B. Overhead Analysis

In the third experiment, we evaluate the power consumption and area overhead of the circuit considering the clock network and inserted logics. In Table III, the second column shows the power consumption increase in the circuits which is caused by both frequency increment and our proposed architectures. The hybrid, DTEA slow, and DTEA fast subcolumns of the second column represent the power increase in the circuits plus their clock shifter equipped with hybrid (working at hybrid MAF), DTEA (working at hybrid MAF), and DTEA (working at DTEA MAF) techniques. The next column of Table III reports the percentage of area overhead for hybrid and DTEA techniques. The column #HTV buffer represents the number of added buffers in the design to prevent hold time violation. The total number of critical paths, the paths equipped with complete prediction, and the path with partial prediction using the DTEA technique are presented in #Criticals column. As illustrated in Table III, the six benchmarks have 39.8% higher power consumption using the DTEA technique on average. Considering 19.8% increase in frequency, our proposed architectures only accounts for 20.0% of the power overhead. At isofrequency condition, the benchmark circuits consume 5.6% less power using our DTEA than hybrid technique [2] on

TABLE IV
PERCENTAGE OF VALID OUTPUTS IN PRESENCE OF VARIATION

technique	Standard deviation						
	0.05	0.2	0.35	0.5	0.6	0.8	1
valid(Exact)%	100	100	96.1	92.2	73	52	44
valid(DTEA)%	100	100	99.6	97.0	80.9	70.3	62

average. Note that, by increasing the frequency of DTEA (from the frequency of DTEA slow to DTEA fast), more critical paths are introduced which increase the power consumption. Also, the average area of DTEA has increased by 12.2% and 1.1% compared to that of the original circuits and hybrid technique, respectively.

C. Variation Analysis

To evaluate our proposed architectures in the presence of variation, the gate-level netlists of prediction and exact cone of five structures—both for partial and complete prediction—are extracted, and a normal distribution of gate delay is applied to each gate. The mean value of each gate delay was extracted from NanGate 45-nm library and seven different values of standard deviation for all gates were considered. Table IV presents the percentage of valid outputs in the complete gate-level simulation in the exact cone and the proposed architectures invalid (exact)% and valid (DTEA)% rows, respectively. According to Table IV, for a small amount of variations with deviation below 0.2, the error is not introduced in the exact cone. However, at the deviation of 1, 62% of outputs of our proposed architectures (i.e., *D* and PO signals of Fig. 2) are valid while only 44% of the outputs of the exact cone (i.e., *O*pred signals of Fig. 2) are valid (i.e., 18% of invalid outputs are corrected).

IV. CONCLUSION

In this brief, a new DTEA technique was presented which first tries to prevent timing errors using the time borrowing technique of [1]. If the structure of the critical paths does not allow time borrowing, our technique utilizes prediction logic of the critical paths—having a smaller delay than that of the exact one—along with dynamic clock stretching to achieve the high-performance operation of the circuit.

REFERENCES

[1] M. Ahmadi, B. Alizadeh, and B. Forouzandeh, "A timing error mitigation technique for high performance designs," in *Proc. IEEE Comput. Soc. Annu. Symp. (VLSI)*, Jul. 2015, pp. 428–433.
[2] M. Ahmadi, B. Alizadeh, and B. Forouzandeh, "A hybrid time borrowing technique to improve the performance of digital circuits in the presence of variations," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 1, pp. 100–110, Jan. 2017.
[3] D. Ernst, *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. 36th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2003, pp. 7–18.
[4] D. J. Banñeres and J. Cortadella, M. Kishinevsky, "Variable-latency design by function speculation," in *Proc. Conf. Design Automat. Test Eur.*, Apr. 2009, pp. 1704–1709.
[5] J. S. Wang, "Dynamic voltage scaling system having time borrowing and local boosting capability," U.S. Patent 8 933 726, Aug. 26, 2014.
[6] M. Wieckowski *et al.*, "Timing yield enhancement through soft edge flip-flop based design," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2008, pp. 543–546.
[7] K. Chae and S. Mukhopadhyay, "Resilient pipeline under supply noise with programmable time borrowing and delayed clock gating," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 3, pp. 173–177, Mar. 2014.
[8] M. Nejat, B. Alizadeh, and A. Afzali-Kusha, "Dynamic flip-flop conversion: A time-borrowing method for performance improvement of low-power digital circuits prone to variations," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, pp. 2724–2727, Nov. 2015.
[9] S. Salamat, M. Ahmadi, B. Alizadeh, and M. Fujita, "Systematic approximate logic optimization using don't care conditions," in *Proc. 18th Int. Symp. Qual. Electron. Design (ISQED)*, Mar. 2017, pp. 419–425.
[10] Accessed: May 2018. [Online]. Available: http://www.cad.polito.it
[11] Accessed: May 2018. [Online]. Available: https://www.synopsys.com
[12] K. Chae and C.-H. Lee, "Timing error prevention using elastic clocking," in *Proc. IEEE Int. Conf. IC Design Technol.*, May 2011, pp. 1–4.
[13] Accessed: May 17, 2018. [Online]. Available: http://projects.si2.org/openeda.si2.org/projects/nangatelib