Power-Efficient Code Converters Using Sub-Threshold Adiabatic Logic Ultra-Low-Power Applications

Prashant Gaurav, Prashant, and Sangeeta Singh

Abstract Power dissipation becomes a decisive parameter in VLSI design in modern-day ultra-low-power applications. Sub-threshold has shown its potential as more efficient logic for ultra-low energy-consuming circuits. Circuits using sub-threshold logic have more timing delay comparable to conventional CMOS logic. Here, code converter circuits are realized using sub-threshold adiabatic logic (SAL) by deploying Cadence 45 nm technology. An extensive simulation study has been carried out, and our study validates the improved circuit performance using sub-threshold adiabatic logic. The present work will facilitate researchers for circuit realization for energy-efficient code converter circuit applications.

Keywords Sub-threshold adiabatic logic · Binary code · Gray code · Excess-3 code · Code converters

1 Introduction

Adiabatic logic is a concept which reduces the power dissipation excessively as compared to conventional CMOS logic. Low power consumption will be achieved by providing a supply that has a gradually varying voltage. Sub-threshold adiabatic logic has very less total power dissipation compared to conventional CMOS logic. The word adiabatic is derived from Greek which is referred to as a thermodynamic activity in which there is no exchange of energy with the surroundings and therefore concluded no power dissipation loss. The transistors count will be nearly half in adiabatic logic as compared to conventional CMOS logic design. Area required and delays are also comparatively lower in adiabatic logic as compared to conventional CMOS logic. Energy recovery loss is another name for adiabatic logic. Low power dissipation can be achieved by the adiabatic technique through charging and discharging the nodes using adiabatic nature. Energy stored in the load capacitor is reused in adiabatic logic circuits [1–3].

P. Gaurav · Prashant · S. Singh (✉)
Microelectronics and VLSI Design, National Institute of Technology Patna, Patna, Bihar, India
e-mail: sangeeta.singh@nitp.ac.in

V. Goyal et al. (eds.), *Proceedings of International Conference on Communication and Artificial Intelligence*, Lecture Notes in Networks and Systems 192,
https://doi.org/10.1007/978-981-33-6546-9_12

1.1 Adiabatic Logic

Time-varying voltage source or constant current source as shown in Fig. 1a is used for charging the capacitor in adiabatic switching. ON resistance of the PMOS network is represented by *R*. Some fraction of the total energy gets stored in the capacitor which can later be claimed back by reversing the current source direction, and thus, the charge gets switch from the capacitance back into the supply.

Adiabatic switching during discharging phase is shown in Fig. 1b. Therefore, adiabatic logic circuit requires time-varying voltage source as a supply voltage. For charging the load capacitance, a constant voltage source is used in conventional CMOS logic, whereas it gets charged by constant current source in adiabatic logic.

Supply voltage applied to the adiabatic logic changes gradually (e.g., ramped waveform). The potential drop across the resistor becomes very less due to the ramp waveform voltage. As a result, the energy dissipation across the resistance during the charge–discharge operation decreases. Power is represented as $P = E_{\text{Total}}/T = C_L {V_{DD}}^2 f$ if the supplied voltage is a ramped waveform with period T (i.e., frequency $f = 1/T$). Total energy consumption for both operations in the above case is given by

$$E_{\text{adiabatic}} = k(PT) = kI^2RT = k(C_L V_{DD}/T)^2 RT \tag{1}$$

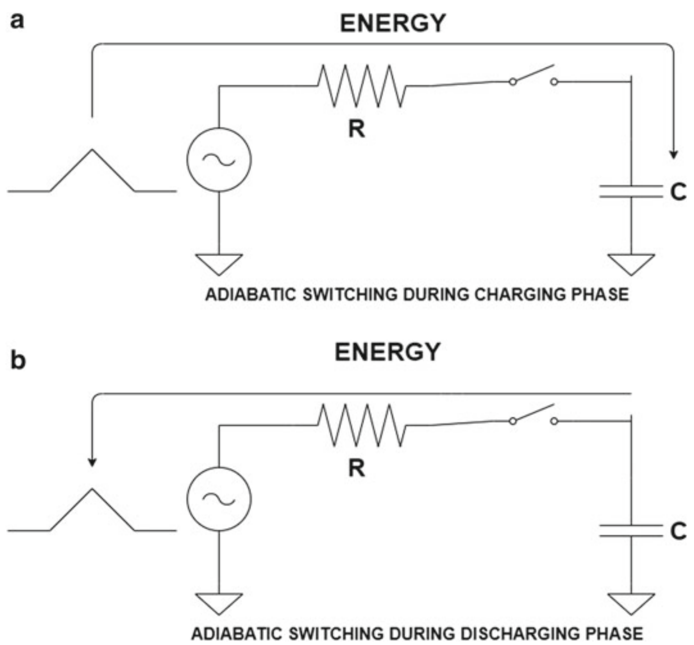

Fig. 1 Adiabatic switching **a** during charging and **b** during discharging

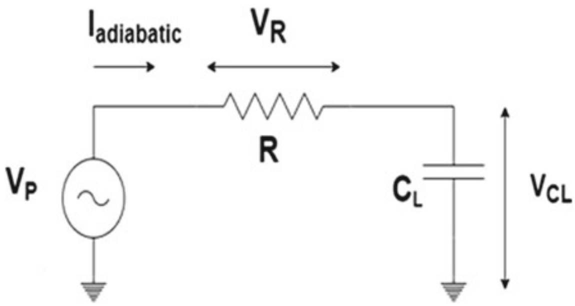

Fig. 2 Circuit diagram with ramp voltage source (used in adiabatic logic)

where k is the shape factor, and shape of the clock edges decide the value of k. It can be concluded here that when the period of signal T is adequately large, the energy consumption in adiabatic logic is considerably less as compared to CMOS logic. Both of these transistors can be modeled as ideal switches in series with a resistor and load capacitance C_L, as shown in Fig. 2. The channel resistance of each transistor is equal to the resistor.

Sub-threshold Adiabatic Logic The analysis of $I_D - V_{GS}$ characteristics of an NMOS transistor (W/L ratio:45 nm/45 nm) suggests that at $V_{GS} = V_T$ (where V_T is the threshold voltage of MOSFET), drain current is not equal to zero because MOS already conducts at $V_{GS} < V_T$. This region is known as the "sub-threshold" or "weak-inversion" conduction region. Leakage current in sub-threshold logic flows between drain and source regions in MOSFET and is expressed as

$$I_o = \mu C_{OX}(W/L)(n-1)V_T^2 \quad (2)$$

$$I_{ds} = I_{oe}(V_{GS} - V_{TH})/nV_T \quad (3)$$

where μ is the mobility, C_{ox} is gate oxide film capacity, V_T is the thermal voltage which is equal to 26 mV at 300 K, W&L is width and length of channel, respectively, n is sub-threshold slope parameter [4–9]. The main component of leakage in sub-threshold devices is sub-threshold leakage current, and many other leakage components which are nearly equal in magnitudes are dependent on the device design parameters. The delay of the circuit increases quickly since the driving current reduces exponentially. Therefore, SAL logic can only be applied to confined areas where performance is not of primary importance. Sub-threshold conduction is very small for long-channel devices in OFF state [10–16]. It is a considerable factor when transistor size, as well as supply voltage, is scaled down. Delay in SAL logic is comparable to conventional CMOS logic. SAL logic can be used where performance is not the key.

2 Code and Code Converter

2.1 Gray Code

Gray code is an ordering of the binary number system in such a manner that each incremental value can only differ from the previous value by only one bit. It is also known as cyclic code as each successive code word differs from the preceding one in only one bit position. It is also a popular example of reflective codes. It is widely used in digital communication for error correction. Gray codes are used in linear and rotary position encoders instead of weighted binary encoding. This means that while using gray code in rotator shaft encoder, only a single bit differs in successive bits; so if multiple bit differs, it will be easy to detect errors.

2.1.1 Binary-to-Gray Code Converter and Gray-to-Binary Code Converter

This converter is a combinational circuit that converts binary code to equivalent gray code. The leftmost bit of gray code is equivalent to the leftmost bit of the given binary code. The second leftmost bit of the gray code is the EX-OR of the leftmost and the second leftmost bit of given binary number. The third leftmost bit of the gray code is the EX-OR of the second leftmost and third leftmost bit of given binary number. And in this manner, binary code to gray code conversion goes on. The gate-level circuit implementation for binary-to-gray code converter is shown in Fig. 3.

This converter is a logical circuit that converts gray code to equivalent binary code. The leftmost bit of binary code is equivalent to the leftmost bit of given gray code. The second leftmost bit of the binary code is the XOR of the leftmost and second leftmost bit of the given gray code. The third leftmost bit of the binary code is the XOR of the second leftmost bit of gray code and third leftmost bit of given binary code. Hence, in this manner, gray code to binary code conversion goes on.

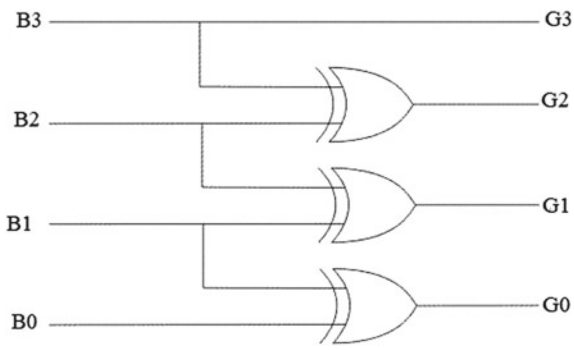

Fig. 3 Gate-level circuit of binary-to-gray code converter

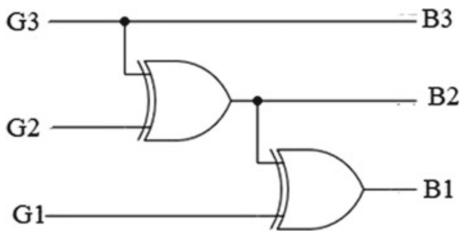

Fig. 4 Gate-level circuit of gray-to-binary code converter

The gate-level circuit implementation for gray-to-binary code converter is shown in Fig. 4.

2.1.2 Excess-3 Code and Binary-to-Excess-3 Code Converter

Excess-3 code is a non-weighted code, where each digit binary code word is the combination of corresponding 8421 code word and 0011. Non-weighted are codes that are not assigned fixed values. It is a biased representation. It is also a self-complementary code. It overcomes the difficulties faced during arithmetic operation in 8421 BCD code. Another major advantage of this representation is that the 0000 and 1111 codes are not used for representation of any digit. It is a logical circuit which converts binary-coded decimal to excess-3 code converter. Binary-coded decimal can be converted to excess-3 code by adding 0011 to the given code. Logical expression for this conversion is given below.

$$Y_3 = B_3 + B_2 \cdot B_1 + B_2 \cdot B_0 \tag{4}$$

$$Y_2 = B_2' B_1 + B_2' B_0 + B_2 B_1' B_0' \tag{5}$$

$$Y_1 = B_1 B_0 + B_1' B_0' \tag{6}$$

$$Y_0 = B_0' \tag{7}$$

where binary and excess-3 codes are given by $B_3 B_2 B_1 B_0$ and $Y_3 Y_2 Y_1 Y_0$, respectively.

3 Simulation Result and Discussion

The simulation analysis results presented are obtained on 45-nm Cadence Virtuoso using SAL logic [17, 18]. Figure 5 demonstrates the output waveform of SAL binary-to-gray code converter. In this figure, the first plot is of supply voltage; the second, third, fourth and fifth plots are of B_3, B_2, B_1 and B_0, respectively, where $B_3B_2B_1B_0$ is

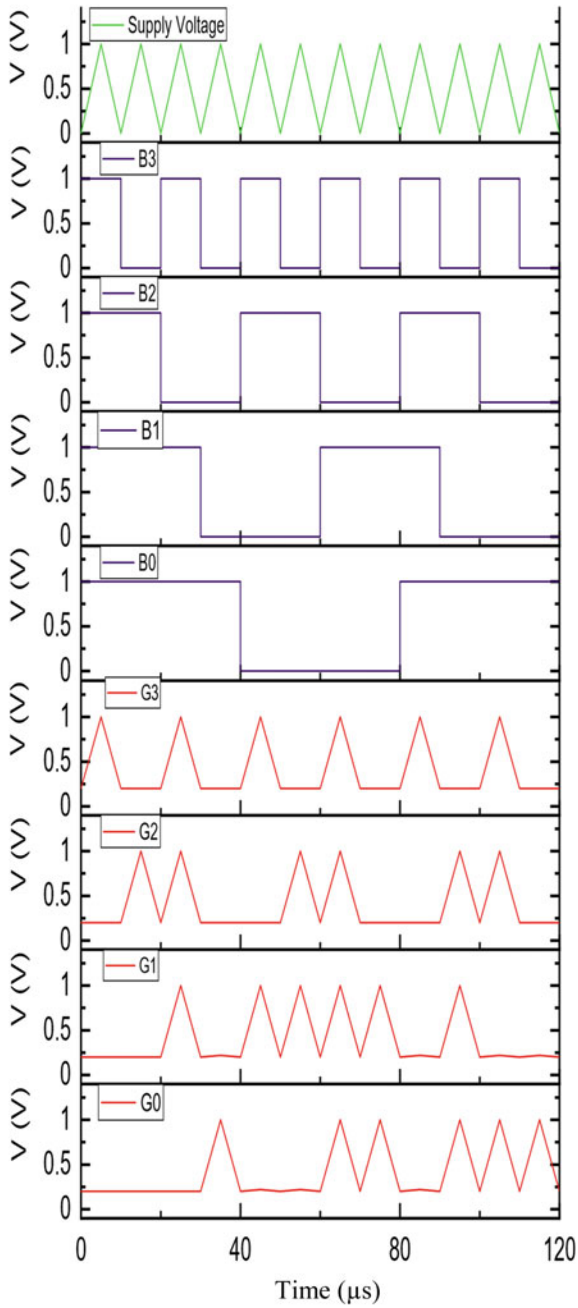

Fig. 5 Output waveform of SAL binary-to-gray code converter

the binary code data. The fifth, sixth, seventh and eighth plot are of G_3, G_2, G_1 and G_0, respectively, where, $G_3G_2G_1G_0$ is the gray code data. Figure 6 shows the output waveform of SAL gray-to-binary code converter. Here, in this figure, the first plot is of supply voltage; the second, third, fourth and fifth plots are of G_3, G_2, G_1 and G_0, respectively, where $G_3G_2G_1G_0$ is the gray code data. The fifth, sixth, seventh and eighth plot are of B_3, B_2, B_1 and B_0, respectively, where $B_3B_2B_1B_0$ is the binary code data. Figure 7 shows the output waveform of SAL binary-to-excess-3 code converter. Here, in this figure, the first plot is of supply voltage; the second, third, fourth and fifth plots are of B_3, B_2, B_1 and B_0, respectively, where $B_3B_2B_1B_0$ is the binary code data. The fifth, sixth, seventh and eighth plot are of Y_3, Y_2, Y_1 and Y_0, respectively, where $Y_3Y_2Y_1Y_0$ is the excess-3 code data.

From Tables 1, 2 and 3, it can easily be observed that power dissipation in SAL logic is approximately 10^{-3} times lower that of power dissipation in CMOS logic.

For binary-to-gray code converter bit G_3, the adiabatic circuit has a considerably low power dissipation of 2.71 pW against CMOS logic, 3.495 nW. For gray-to-binary code converter bit B_0, the adiabatic circuit has a considerably low power dissipation of 27.52 pW as compared to conventional CMOS logic, 8.11 nW. For binary-to excess-3-code converter bit Y_3, the adiabatic circuit has a considerably low power dissipation of 129.06 pW against conventional CMOS logic, 3.86 nW. It can easily be observed that power dissipation in SAL logic is approximately 10^{-3} times lower that of power dissipation in CMOS logic. The detailed performance comparison of SAL and CMOS-based converter circuit is shown in Tables 1, 2 and 3. For CMOS circuit and SAL circuit, the peak DC voltage connected and peak ramp voltage applied are 1 V.

4 Conclusion

Binary code to gray code converter, gray code to binary code converter and binary code to excess-3 code converter are realized using sub-threshold adiabatic logic (SAL) and compared with conventional CMOS logic in this paper. From the above simulation results, we conclude that SAL reduces appreciable amount of energy as in parallel with conventional CMOS logic. Scaling down of power dissipation in adiabatic circuits is mainly because of recycling of energy stored in the capacitive loads. SAL is preferred for application which requires low frequency. This proposed sub-threshold adiabatic logic can be used in energy-efficient converter circuit.

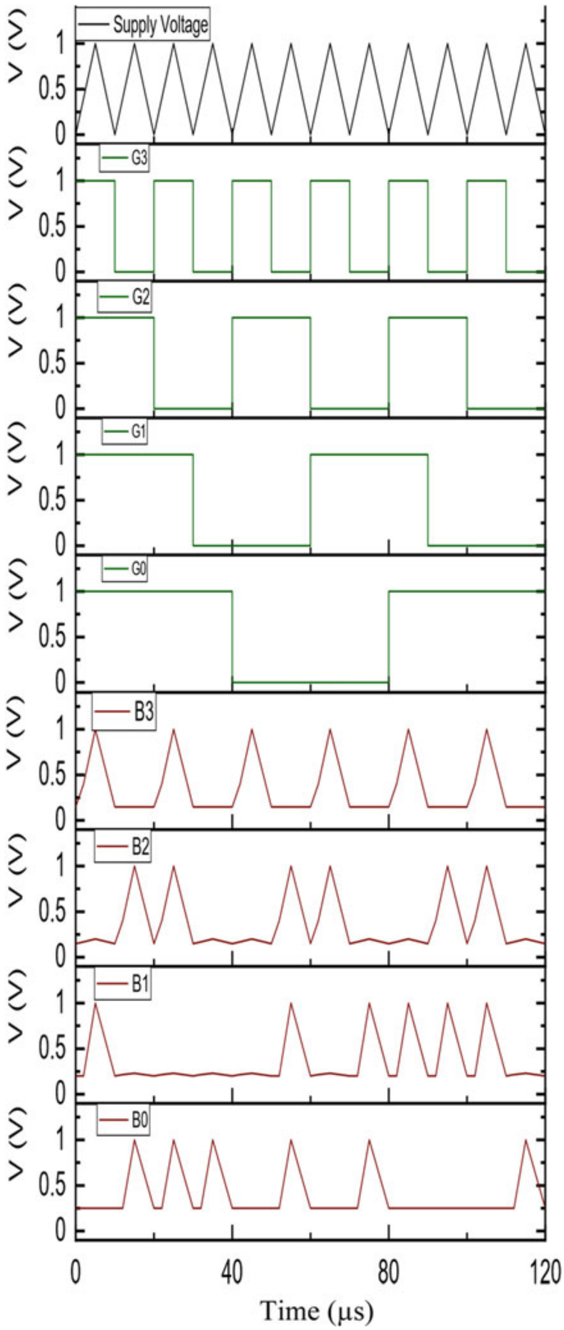

Fig. 6 Output waveform of SAL gray-to-binary code converter

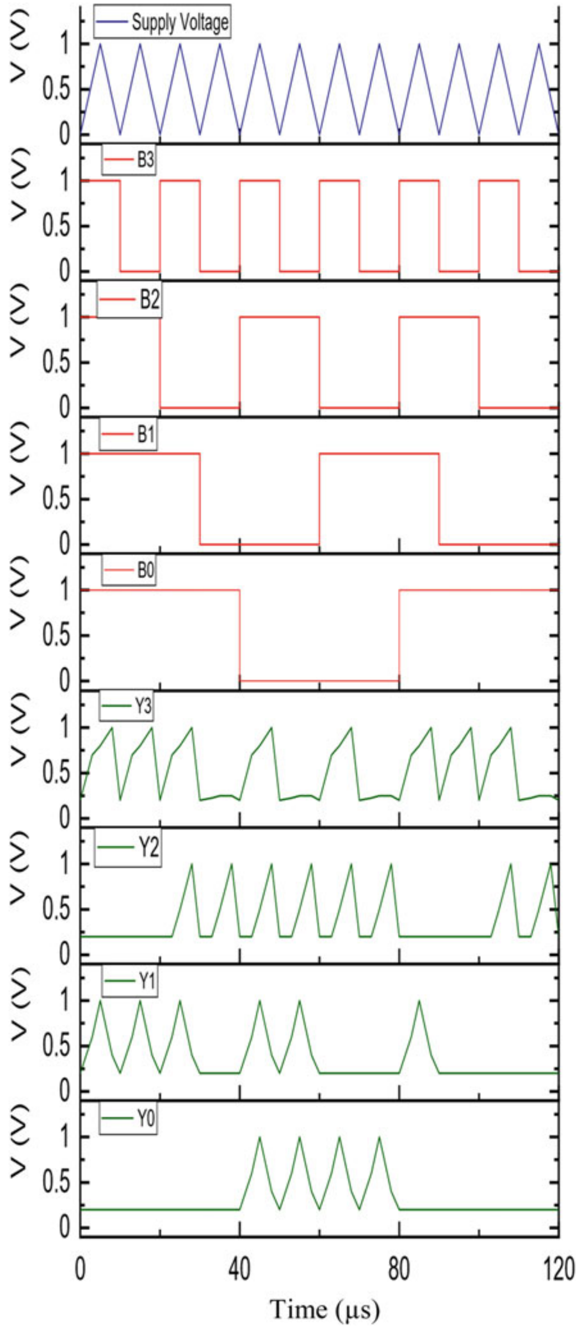

Fig. 7 Output waveform of SAL binary-to-excess-3 code converter

Table 1 Comparison of total power dissipation of binary-to-gray code converter using SAL logic and CMOS logic

BITS	SAL logic (pw)	CMOS logic (nw)
G_3	2.71	3.495
G_2	2.714	1.84
G_1	2.452	2.08
G_0	2.306	2.07

Table 2 Comparison of total power dissipation of gray-to-binary code converter using SAL logic and CMOS logic

BITS	SAL logic (pw)	CMOS logic (nw)
B_3	2.716	3.495
B_2	2.714	1.86
B_1	13.52	5.117
B_0	27.52	8.11

Table 3 Comparison of total power dissipation of binary-to-excess-3 code converter using SAL logic and CMOS logic

BITS	SAL logic (pw)	CMOS logic (nw)
Y_3	129.06	3.86
Y_2	87.12	2.418
Y_1	2.316	2.96
Y_0	1.68	2.89

References

1. Chaudhuri A, Saha M, Bhowmik M, Pradhan SN, Das S (2015) Implementation of circuit in different adiabatic logic. In: 2015 2nd IEEE international conference on electronics and communication systems (ICECS), pp 353–359
2. Chanda M, Jain S, De S, Sarkar CK (2015) Implementation of subthreshold adiabatic logic for ultralow-power application. IEEE Trans Very Large Scale Integr (VLSI) Syst 23(12):2782–2790
3. Yadav RK, Rana AK, Chauhan S, Ranka D, Yadav K (2011) Adiabatic technique for energy efficient logic circuits design. In: 2011 IEEE international conference on emerging trends in electrical and computer technology, pp 776–780
4. Pindoo IA, Dhariwal S, Sharma R, Lata S (2018) Speed enhancement in the performance of two phase clocked adiabatic static CMOS logic circuits. In: International conference on intelligent circuits and systems (ICICS), Phagwara, India, pp 149–154
5. Grover V, Gosain V, Pandey N, Gupta K (2018) Arithmetic logic unit using diode free adiabatic logic and selection unit for adiabatic logic family. In: 5th international conference on signal processing and integrated networks (SPIN), Noida, India, pp 777–781
6. Wang A, Calhoun BH, Chandrakasan AP (2006) Sub-threshold design for ultra low-power systems, vol 95. Springer, New York
7. Takahashi Y, Sekine T, Nayan NA, Yokoyama M (2012) Power saving analysis of adiabatic logic in subthreshold region. In: 2012 IEEE international symposium on intelligent signal processing and communications systems, pp 590–594
8. Khatir M, Mohammadi HG, Ejlali A (2010) Sub-threshold charge recovery circuits. In: IEEE international conference on computer design, pp 138–144
9. Moon Y, Jeong DK (1996) An efficient charge recovery logic circuit. IEICE Trans Electr 79(7):925–933

10. Goyal S, Singh G, Sharma P (2015) Variation of power dissipation for adiabatic CMOS and conventional CMOS digital circuits. In: 2015 2nd international conference on electronics and communication systems (ICECS), pp 162–166
11. Kim S, Papaefthymiou MC (2001) True single-phase adiabatic circuitry. IEEE Trans Very Large Scale Integr (VLSI) Syst 9(1):52–63
12. Yadav RK, Rana AK, Chauhan S, Ranka D, Yadav K (2011) Four phase clocking rule for energy efficient digital circuits—an adiabatic concept. In: 2011 2nd IEEE international conference on computer and communication technology (ICCCT-2011), pp 209–214
13. Samanta S (2009) Adiabatic computing: a contemporary review. In: 2009 4th international conference on computers and devices for communication (CODEC), pp 1–4
14. Chanda M, Dutta R, Rahaman A, Sarkar CK (2016) Analysis of NAND/NOR gates using subthreshold adiabatic logic (SAL) for ultra low power applications. In: 2016 international conference on microelectronics, computing and communications (MicroCom), pp 1–5
15. Maheshwari S, Bartlett VA, Kale I (2018) VHDL-based modelling approach for the digital simulation of 4-phase adiabatic logic design. In: 2018 28th IEEE international symposium on power and timing modeling, optimization and simulation (PATMOS), pp 111–117
16. Maheshwari S, Bartlett VA, Kale I (2019) Modelling, simulation and verification of 4-phase adiabatic logic design: a VHDL-Based approach. Integration 67:144–154
17. Chanda M, Mal S, Mondal A, Sarkar CK (2018) Design and analysis of a logic model for ultra-low power near threshold adiabatic computing. IET Circ Dev Syst 12(4):439–446
18. Chanda M, Ganguli T, Mal S, Podder A, Sarkar CK (2017) Energy efficient adiabatic logic styles in sub-threshold region for ultra low power application. J Low Power Electr 13(3):472481; Eason G, Noble B, Sneddon IN (1955) On certain integrals of Lipschitz-Hankel type involving products of Bessel functions. Phil. Trans. Roy. Soc. Lond. A247:529–551