Design and analysis of gray code generator as test pattern generator

Palla Narasimha
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India
Email: pallanarasimha111@gmail.com

Geethu Remadevi Somanathan
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India
Email: geethurs@am.amrita.edu

Maddala Vamsi Krishna
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India
Email: vamsikrishnamaddala123@gmail.com

Ramesh Bhakthavatchalu
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India

***Abstract*—Gray code is a process of encoding integers as a sequence of binary bits. These binary bits are arranged in such a way that two adjacent integers differ by a single binary digit. There are different types of grey code such as reflected binary, uniform balanced, antipodal, and so on. This paper mainly focuses on the reflected binary grey code. The reflected binary code is light weighted. The primary function of counters is to produce a specified output sequence and they are also called pattern generators. This paper contains the comparison of different bits of grey counters. The parameters of comparison are static power, dynamic power, utilization, and on-chip temperature.**

***Keywords*— Gray code, Reflected binary code, Vivado, Verilog, Static power, Dynamic power**

I. INTRODUCTION

It is similar to the binary system in which two successive numbers vary in only one bit [1], [2]. If the first half values of gray code are compared with the other half, they are exactly opposite, so it's called Reflected binary code [3]. Because there is only a one-bit change, the amount of switching is much less than in binary and switching systems are more reliable. It is also called a cyclic variable code. It is also a light-weighted code as it won't depend on the value of the digit specified by the position. It is used for error correction in digital communications like cable TV systems, Karnaugh-map function reduction. It is also used in the process of a chip design that works with various clock frequencies in digital logic design, FIFO, which is mainly used in the representation of some state machines and some converters [4]. It is mainly used in applications where the usage of binary sequence might produce an error during their transition from one number to the other [5].

Counters are the basic building blocks of many digital systems and have many applications and they are supposed to be fast and less power conservative. These are sequential circuits, which are mainly used to generate a particular sequence [6]. A counter that generates the output of the gray sequence is called the n-bit gray counter. We can generate gray code of binary sequence by applying exclusive-or (XOR) gates [7].

II. RELATED WORK

Vivado is a software developed by Xilinx mainly for the analysis and synthesis of HDL (Hardware Description languages) [6]. It helps us in verifying and implementing our design. Through this, we can examine and check the accuracy of the design and it also allows us to change the design at any step in this process [8]. In Vivado, there also exist some processes like Simulation, RTL analysis, power consumption, and utilization to improve the circuit performance [9]. Vivado simulator is a feature-rich, mixed language simulator as it supports simulation

models in both Verilog and VHDL languages. It also supports TCL scripts and encrypted IP [8]. It exists in all Vivado editions. It does not have any design size or instances, so we can use it to run more instances of a single language and mixed-language simulations.
Leakage current that is abstracted from the power source leads to static power. Static power dissipation takes place when the device is in on state and switching activity is not involved. When the discharging and the charging takes place at the load capacitor, that leads to the dynamic power [10]. It occurs mainly during switching activity. Total on-chip power is the sum of static power and dynamic power. Junction temperature is the temperature of the device when the device is in operating condition.

III. METHODOLOGY

In Gray code, the MSB remains the same as that of the MSB of the binary number. For calculating the 2nd bit from the MSB side of the gray code, we have to perform XOR of the MSB bit and the bit next to it of the binary number [3]. It results in 1 when both bits are different otherwise it will result in 0. To find the 3rd bit from the MSB side, we have to perform XOR for the 2nd and 3rd bit from the MSB side of the binary number. The same process should be repeated till the LSB bit of Gray code. The general formula of an n-bit Gray code is [4]:

$$G(n-1) = B(n-1) \quad (1)$$

$$G(k-1) = B(k) \oplus B(k-1) \quad (2)$$

Where G denotes Gray, B denotes Binary, $0 < k < n$ and $\oplus$ represents XOR operation [11]

TABLE I shows us the difference between the gray and binary representation of decimal numbers from 0 to 7.

TABLE I: 3 bit gray code

Binary	Gray	Decimal
000	000	0
001	001	1
010	011	2
011	010	3
100	110	4
101	111	5
110	101	6
111	100	7

IV. PROPOSED WORK

Here we are executing the entire process in Vivado HLS (High-Level Synthesis) 2018.3. The design of gray code starts with the design of the Verilog files. In order to design an n-bit gray counter, we need an n number of flip-flops. Firstly, we write the Verilog code to the T-flipflop and then to the design file, which will generate the required output [7], [12]. The output of the flip-flops is the binary output, so by applying the XOR we will get the sequence of the gray counter.

The next step in this process is Simulation [4], [7], [13]. We executed this process in the Vivado simulator. For Simulation, we provided the values for the input parameters. So after running the simulation, we will get the sequence of the gray counter as output.

The next step in this process is RTL (Register transfer level) analysis, through which we will get the schematic of an n-bit Gray code. It shows the circuit representation of an n-bit Gray code.

The next step in this process is Synthesis [4], [7]. It gives information about the parameters of utilization LUT (Lookup Table), IO (Input-Output), FF (Flip-Flops), BUFG (Global Buffer).

The next step in this process is design implementation. It gives information about on-chip power, which contains static and dynamic power, and junction temperature [14].

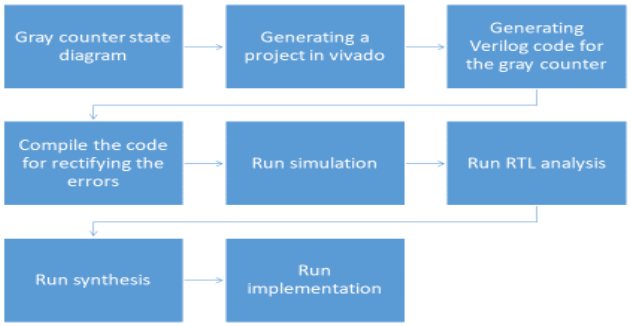

Fig.1. Implementation flow

V. RESULT ANALYSIS

TABLE II: Static power, Dynamic power, Total on-chip power

Bit size	Static power (in W)	Dynamic power (in W)	Total on-chip power in(W)
2 bit	0.099	0.656	0.775
3 bit	0.1	0.693	0.793
4 bit	0.146	2.859	3.005
5 bit	0.153	3.082	3.235
6 bit	0.157	3.202	3.36
7 bit	0.16	3.262	3.421
8 bit	0.161	3.307	3.469

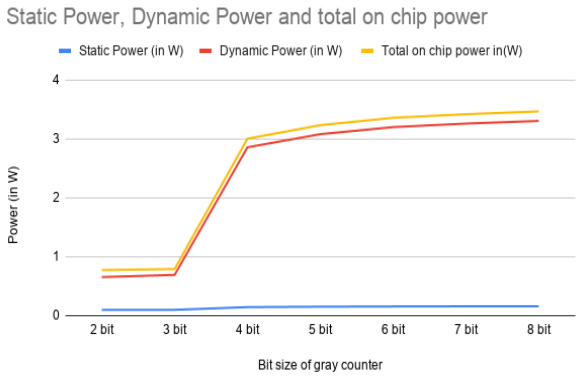

Fig. 2. Static Power, Dynamic Power, Total Power

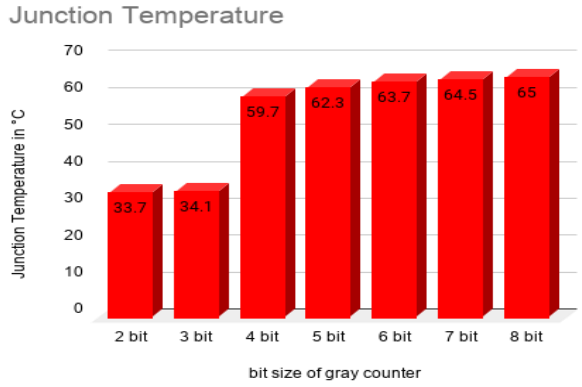

Fig. 3. Junction temperature graph

TABLE III: Utilization

Bit size	Resource	Utilization	Available	Utilization %
2 bit	LUT	2	17600	0.01
	FF	2	35200	0.01
	IO	5	100	5
	BUFG	1	32	3.13
3 bit	LUT	3	17600	0.02
	FF	3	35200	0.01
	IO	6	100	6
	BUFG	1	32	3.13
4 bit	LUT	4	17600	0.02
	FF	4	35200	0.01
	IO	7	100	7
	BUFG	1	32	3.13
5 bit	LUT	5	17600	0.03
	FF	5	35200	0.01
	IO	8	100	8
	BUFG	1	32	3.13
6 bit	LUT	7	17600	0.04
	FF	6	35200	0.02
	IO	9	100	9
	BUFG	1	32	3.13
7 bit	LUT	9	17600	0.05
	FF	7	35200	0.02
	IO	10	100	10
	BUFG	1	32	3.13
8 bit	LUT	11	17600	0.06
	FF	8	35200	0.02
	IO	11	100	11
	BUFG	1	32	3.13

The power, junction temperature, and utilization are calculated using the Vivado software tool. From TABLE II, it is observed that for 2-bit and 3-bit gray code the dynamic power and total on-chip power is very less but for 4-bit to 8-bit gray code, the dynamic and total on-chip power is very high. In the case of junction temperature also, it's less for 2-bit and 3-bit but high for other bits gray code, which is shown in Fig 3. Table III contains the report of utilization parameters of LUT, IO, BUFG, and FF.

RTL SCHEMATICS

Fig.4, Fig.5, and Fig.6 contain the schematics of the 3, 4, and 5 bits respectively. The schematics of other bits are also evaluated.

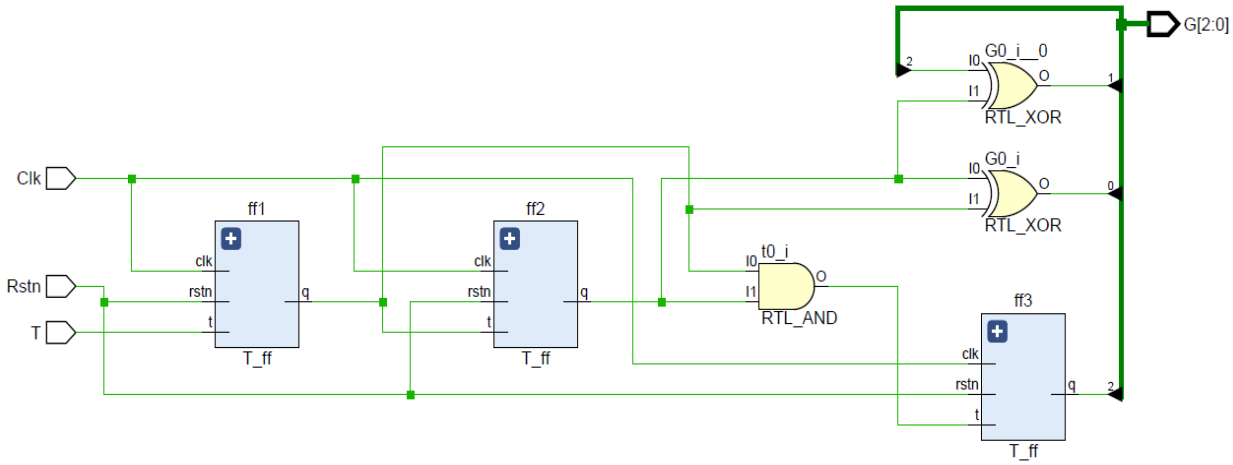

Fig.4. 3 bit schematics

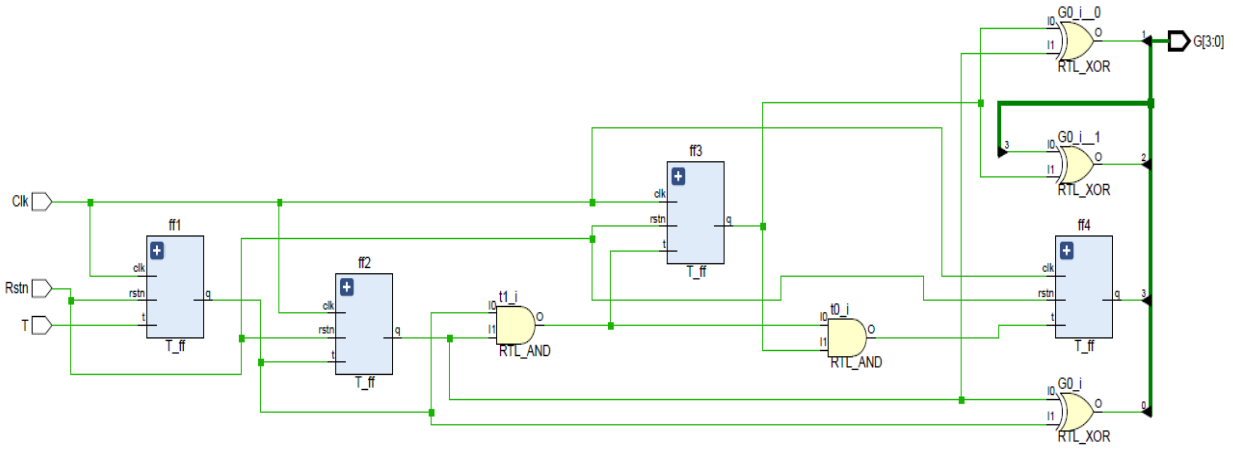

Fig.5. 4 bit schematics

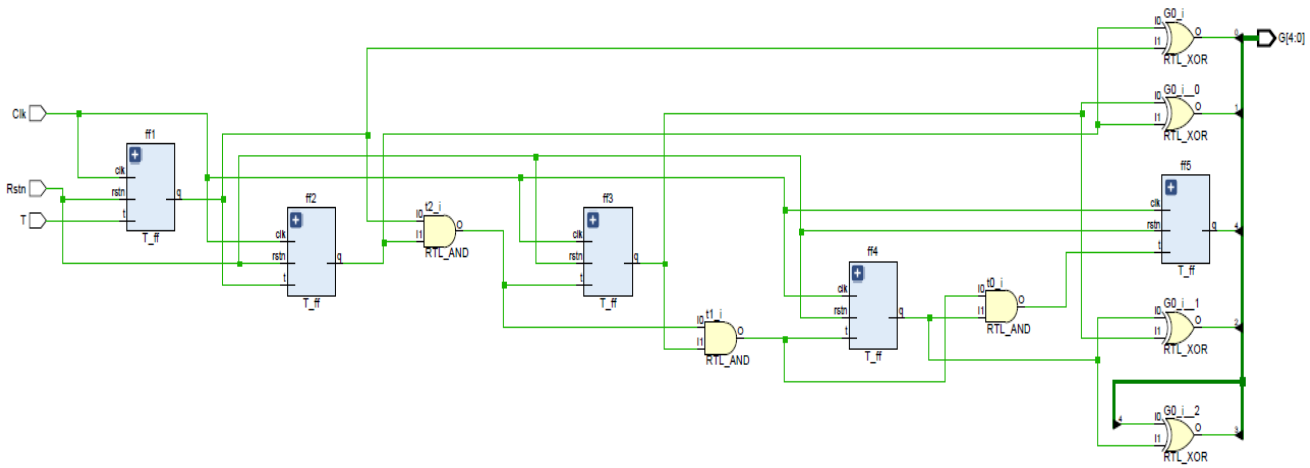

Fig.6. 5 bit schematics

VI. CONCLUSION AND FUTURE WORK

In this paper, we generated the sequence of gray counter and implemented it using the Vivado software tool. Here we have made the comparison of the lower-bit gray code with the higher ones. We observed that for lower bits (2-bit and 3-bit), power consumption is relatively low when compared with higher bit gray code (4-bit to 8-bit) power consumption. When compared with the lower bits, the higher bit gray code contains high dynamic power and low static power. This indicates that power leakage decreases when we move from the lower bits to the higher bit gray code. The utilization of LUT, IO, and FF starts increasing when we move from the lower bits of the higher-bit gray code. So, the utilization percentage is more for higher bit gray codes than the lower ones. The junction temperature also increases when we move from the lower bits to the higher bit gray code. Further, we are willing to continue our analysis of other types of gray codes [15].

VII. REFERENCES

[1] C. Li, "A Gray Code Based Time-to-Digital Converter Architecture and its FPGA Implementation," pp. 178–180, 2015.

[2] C. Savage and A. G. Bell, "A survey of combinatorial gray codes ∗," vol. 39, no. 4, pp. 605–629, 1997.

[3] Y. Zhou, K. Panetta, S. Agaian, S. Member, and C. L. P. Chen, "(n , k , p) -Gray Code for Image Systems," vol. 43, no. 2, pp. 515–529, 2013.

[4] A. Ahmad and F. Bait-Shiginah, "A nonconventional approach to generating efficient binary gray code sequences," *IEEE Potentials*, vol. 31, no. 3, pp. 16–19, 2012, doi: 10.1109/MPOT.2011.2178193.

[5] V. Hdl, *Digital design with an introduction to the verilog hdl*. 2010.

[6] S. S. S. G. Seeram, S. N. N. Polireddi, R. S. Geethu, and R. Bhakthavatchalu, "Synthesis of Synchronous Gray Code Counters by Combining Mentor Graphics HDL Designer and Xilinx VIVADO FPGA Flow," *Proc. 2020 IEEE Int. Conf. Commun. Signal Process. ICCSP 2020*, pp. 738–742, 2020, doi: 10.1109/ICCSP48568.2020.9182333.

[7] P. S. Dilip, G. R. Somanathan, and R. Bhakthavatchalu, "Comparative Study of Test Pattern Generation Systems to Reduce Test Application Time," *Proc. 2019 Int. Symp. Embed. Comput. Syst. Des. ISED 2019*, pp. 59–62, 2019, doi: 10.1109/ISED48680.2019.9096234.

[8] B. N. Sumanth, B. L. Reddy, and G. R. Somanathan, "Synchronous Counters," no. Icoei, pp. 80–84, 2020.

[9] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design*, no. 1. 2014.

[10] R. K. Megalingam, G. Popuri, and P. Ravisankar, "Low power consumption coarse grained reconfigurable

adder," *2009 Int. Conf. Comput. Electr. Eng. ICCEE 2009*, vol. 2, pp. 503–506, 2009, doi: 10.1109/ICCEE.2009.195.

[11] L. Grover, "Weighted code approach to generate Gray code," *IEEE Potentials*, vol. 34, no. 3, pp. 39–40, 2015, doi: 10.1109/MPOT.2013.2295874.

[12] J. Schostak *et al.*, *No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title*, vol. بعد الى حتا دالا, no. Mm. 2003.

[13] K. Kamel and E. Kamel, "Process Control Ladder Logic Trouble Shooting Techniques Fundamentals," *IRO J. Sustain. Wirel. Syst.*, vol. 1, no. 04, pp. 206–214, 2019, doi: 10.36548/jsws.2019.4.001.

[14] R. Bhakthavatchalu and G. R. Deepthy, "32-bit Reconfigurable Logic-BIST Design Using Verilog for ASIC Chips," pp. 386–390, 2011.

[15] R. Saligram and T. R. Rakshith, "Contemplation of synchronous Gray Code counter and its variants using reversible logic gates," *2013 IEEE Conf. Inf. Commun. Technol. ICT 2013*, no. Ict, pp. 661–665, 2013, doi: 10.1109/CICT.2013.6558177.