Int J Syst Assur Eng Manag (October 2023) 14(5):1726–1737
https://doi.org/10.1007/s13198-023-01977-2

ORIGINAL ARTICLE

Architectural analysis of 1-D to 2-D array conversion of priority encoder

Alok Kumar Mishra[1] · Shail Anand[1] · Nishant Singh[1] · Vaithiyanathan Dhandapani[1] · Baljit Kaur[1]

Received: 17 October 2021 / Revised: 13 January 2022 / Accepted: 26 May 2023 / Published online: 4 July 2023

Abstract In this paper, a high-performance priority encoder of the 2-dimensional array is investigated and modified. This work involves 64-bit priority encoder design and verification using Verilog and cadence virtuoso. For the Verilog code, Vivado has been used for the Artix-7 FPGA board. For Cadence virtuoso, GPDK 180 nm CMOS technology has been used. By using a reduced 2-D priority encoder, 1000 transistors have been saved. Due to this, the delay of the whole 64-bit priority encoder is reduced by 62.17% as compared to PE64 and 61.68% as compared to modified PE64.

Keywords High performance · Priority encoder · Artix-7 · FPGA · Multiplexer

1 Introduction

In the huge complex processing computer system, always priority needed, for this critical decision-making situation priority encoders (PEs) are required (Nguyen et al. 2017; Delgado-Frias and Nyathi 2000). Nowadays, there is a huge demand for not only a large number of inputs of priority encoder, but along with this, the performance of priority encoder (PE) should also meet our requirements. A high-performance priority encoder is important for the massive amount of data storage and extraction (Huang et al. 2002; Huang and Chang 2010). Priority encoder is widely used in deciding the interrupt request, data compression to optimize the number of wires required (Kun et al. 2004; Le et al. 2012). A normal encoder cannot be used as it cannot handle multiple inputs at a time and hence, the priority encoder overcomes this disadvantage (Nguyen et al. 2017; Tsai et al. 2017; Balobas and Konofaos 2016). The main challenge of a priority encoder is along with the handling of a large number of inputs, the performance of the priority encoder should also not deteriorates. To implement a large number of inputs of priority encoder, a matrix method is designed (Ghosh et al. 2016; Yershov 2018; Wang et al. 2012). This matrix method reduces the complexity of the priority encoder and improves the performance like delay, power, etc. PE detects the position of the highest priority bit in the input (Balobas and Konofaos 2016, 2017). In other words, if more than one inputs are nonzero at the same time then the input having the highest priority will take precedence (Maurya and Clark 2010; Huang and Chang 2010). If more than one operation or event is happening simultaneously, then PEs are used to select one of many events or operations (Das et al. 2017; Rabeya et al. 2019). In digital systems, PEs are widely used where priority decisions are needed (Kumar et al. 2011, 2015). PEs are used to encode the output of a flash analog to a digital converter (Nguyen et al. 2018; Hashemian 1989). It is also used to control interrupt requests by acting on the highest priority request in various processors (Sen et al. 2017; Balobas and Konofaos 2017). PEs are also required in the design of interconnection network routers (INR) and in

✉ Alok Kumar Mishra
alokkmishra@nitdelhi.ac.in; alokmishra014@gmail.com

Shail Anand
192221008@nitdelhi.ac.in

Nishant Singh
202221009@nitdelhi.ac.in

Vaithiyanathan Dhandapani
dvaithiyanathan@nitdelhi.ac.in

Baljit Kaur
baljitkaur@nitdelhi.ac.in

[1] Department of Electronics and Communication Engineering, National Institute of Technology Delhi, Delhi 110040, India

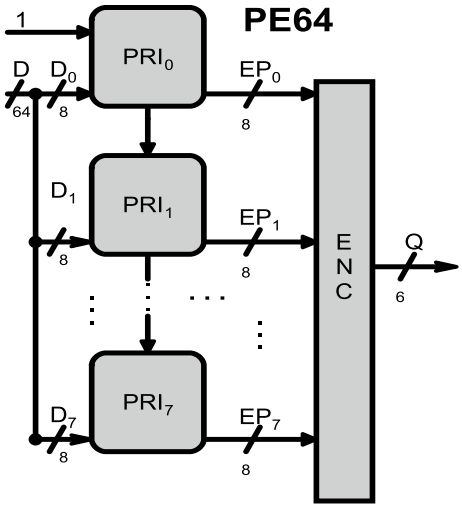

Fig. 1 Block diagram of conventional priority encoder PE64

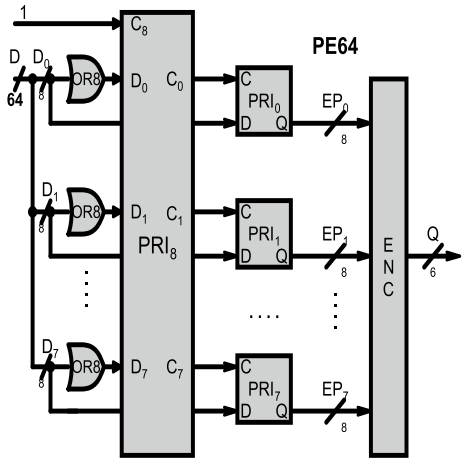

Fig. 2 Parallel priority encoder of 64-bit (PE64) (Kun et al. 2004)

the design of the sequential address encoder (SAE) of a content associate memory (CAM) (Kathuria and Sharma 2019). During the past dozen years, several high and low-priority encoders have been rushed into development. In this paper, 64 input priority encoder is designed using Verilog. After the implementation of a 64×6 priority encoder, a large size of inputs can be designed using this 64×6 priority encoder. Priority encoder is an improved design of a simple encoder (Bao et al. 2020; Saidi et al. 2021).

2 Background of the priority encoder

The conventional architecture of a 64-bit input priority encoder is depicted in Fig. 1. This arrangement includes the set of prioritizers (PRIs) and encoder (ENC). Prioritizers are controlled by a signal C to enable PRI_{i+1}.

The control signal C is generated from the PRI_i, and so on. Initially, they have divided 64-bit input data into groups of 8-bit. Every PRIs which is 8-bit, resolves the top priority bit of that group, whereas the encoder outputs a mapping location into binary format. Figure 2 shows a different architecture in which PRI_0 to PRI_7 can restore their priority maps in parallel because of the enable signal contributed by PRI_8. Although the latency is decreased, the resources utilization increases due to the supplementary PRI_8 and logical gates.

Figure 3 again shows a new architecture of a 64-bit priority encoder which is basically based on four (4) one-hot encoders. This encoder is designed by Le et al. (2012). In this design, each encoder transforms a corresponding sixteen bits groups into the four-bit group and then enable signal C takes the decision that whether the outcomes are passed to the next multiplexers or not.

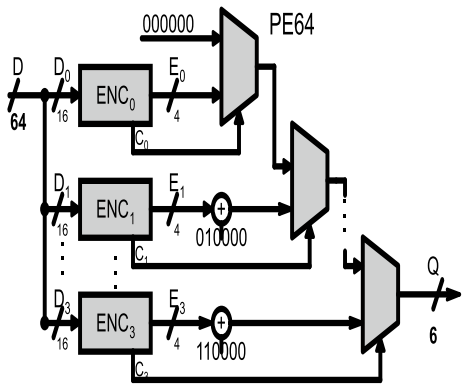

Fig. 3 64-bit priority encoder (PE64) based on one-hot encoder (Le et al. 2012)

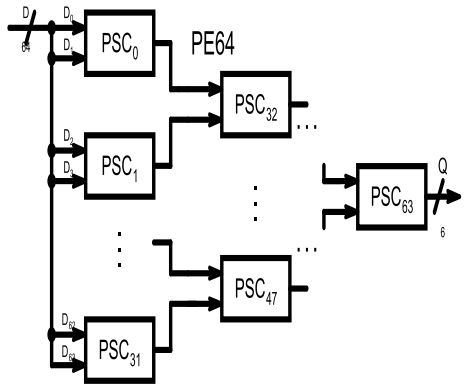

Fig. 4 64-bit priority encoder (PE64) using comparison and sort (Maurya and Clark 2010)

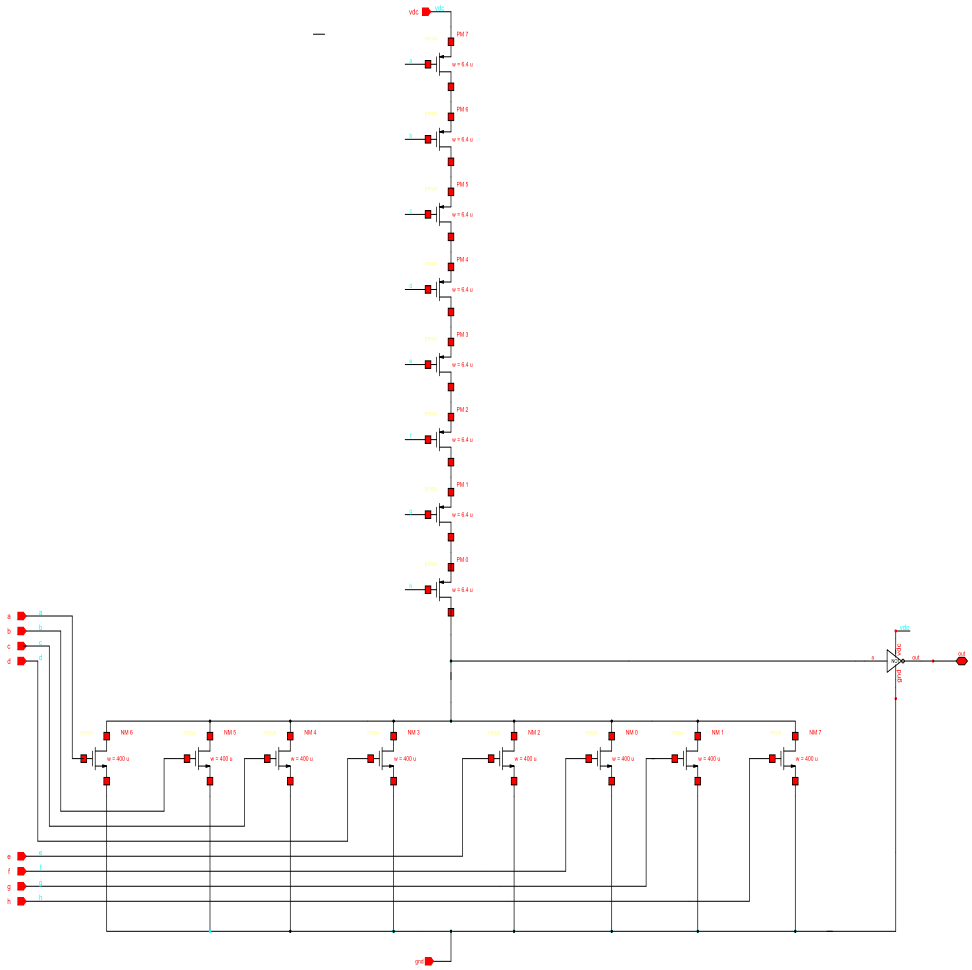

Fig. 5 CMOS schematic of 8-input OR

Figure 4 shows a new approach to design a 64-bits priority encoder. They have proposed a set of the comparators and sort circuits expend check every combination of bits of input data therefore the top priority bit is determined.

2.1 Boolean expression of priority encoder

2.1.1 2-input Priority encoder:

$$Y_0 = \overline{X_2}.X_1 + X_3 \tag{1}$$

$$Y_1 = X_2 + X_3 \tag{2}$$

The above equation 1 and 2 are representing 2-bit priority encoder output.

2.1.2 4-input Priority encoder:

$$Y_0 = \overline{X_{14}X_{13}X_{12}}(\overline{X_{11}X_{10}X_9X_8}(\overline{X_7X_6X_5X_4}(\overline{X_2}.X_1 + X_3) + \overline{X_6}.X_5 + X_7)) + \overline{X_{10}}.X_9 + \overline{X_{11}}) + \overline{X_{14}}.X_{13} + X_{15} \tag{3}$$

$$Y_1 = (\overline{X_{13}X_{12}}(\overline{X_{11}X_{10}X_9X_8}(\overline{X_7X_6X_5X_4}(X_2 + X_3) + X_6 + X_7)) + X_{10} + X_{11}) + X_{14} + X_{15} \tag{4}$$

$$Y_2 = (\overline{X_{11}X_{10}X_9X_8}((X_7 + X_6\overline{X_7X_6}(X_5 + X_4))) + X_{12} + X_{13}) + X_{14} + X_{15} \tag{5}$$

$$Y_3 = X_8 + X_9 + X_{10} + X_{11} + X_{12} + X_{13} + X_{14} + X_{15} \tag{6}$$

We can observe the complexity of expressions increases drastically as the input of the priority encoder varies from 4-bit to 16-bit, which makes the implementation of large input priority encoder impracticable.

3 The reduced architecture of 64-bit priority encoder

In this paper, a reduced transistor 64-bit priority encoder has been implemented. This architecture basically includes two 8:3 priority encoders, eight 8-bit OR gates, and eight 8×1

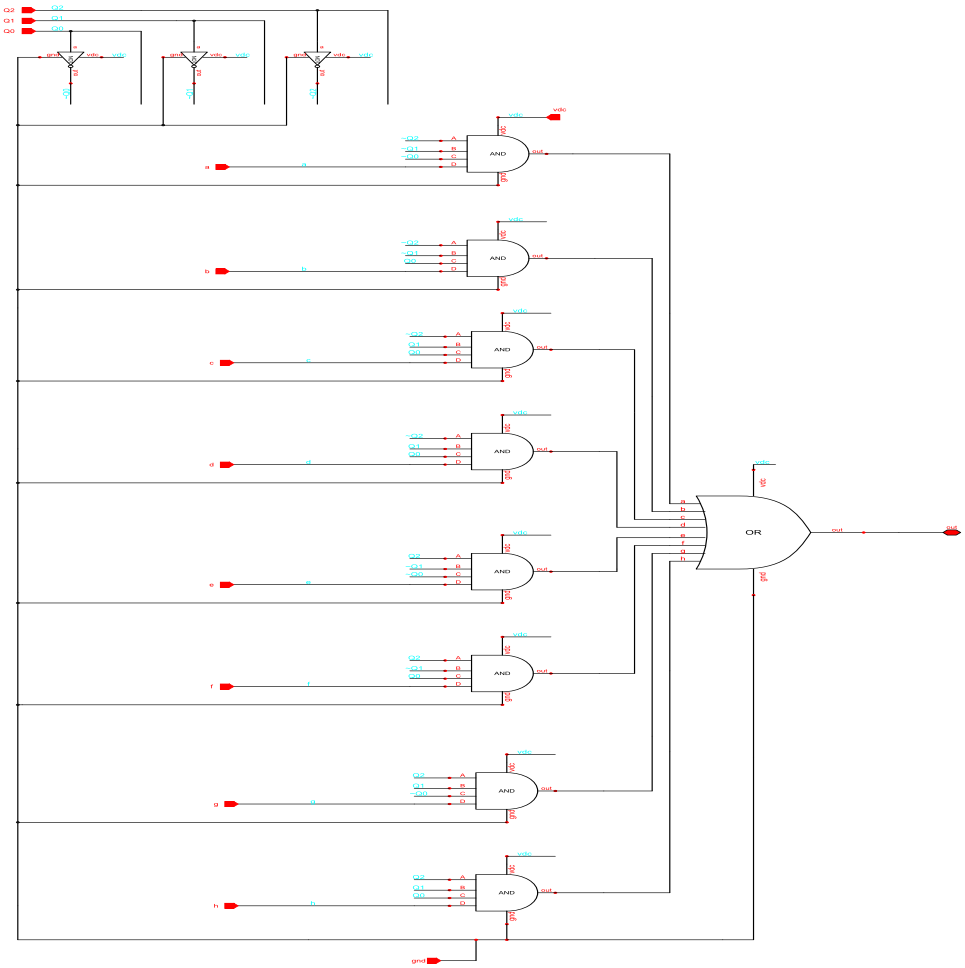

Fig. 6 Gate level schematic of 8:1 Multiplexer

multiplexers. These small architectures are arranged in such a manner to form a 64:6 priority encoder. Here each of the blocks is explained. Modifications are performed to increase the performance of the 64-bit priority encoder. This is done by the reduction of output OR gate and a shift register.

3.1 8 input data OR gate

Figure 5 shows the 8-input OR gate. In the 64-bit priority encoder, eight OR gates have been used which is of 8 input. For proper flow of the current, sizing of the transistor is carefully done. That can charge the output load capacitor through the pull-up network by using vdd → out → load capacitor path. Similarly discharging of load capacitor is done through pull-down network via load capacitor → out → ground.

3.2 Multiplexer 8:1

Figure 6 depicts the 8:1 multiplexer. In this architecture of a 64-bit priority encoder, a total of eight (8:1) multiplexers have been used. In this 8:1 MUX eight 4 - input AND gates, three inverters, and an 8 - input OR gate are used to perform a MUX combined. The 4-input AND gate have three inputs coming from the three inverters and another one is the data for all AND gate.

3.3 Priority encoder 8:3

In this priority encoder architecture design second stage or last stage provides an output which is of 6-bit binary format. This 6-bit output gives the output according to priority. There are two priority encoders each provides 3-bit output. Figure 7 illustrates an 8:3 priority encoder.

4 Architecture description

The architecture in Nguyen et al. (2017) described the architecture of the N-bit Priority Encoder in which they converted 1- Dimensional array into two 2-dimensional (Nguyen et al. 2017). The figure shows that how they have changed N-bits input data into XxY bit matrix. X is represented as columns and Y is represented rows.

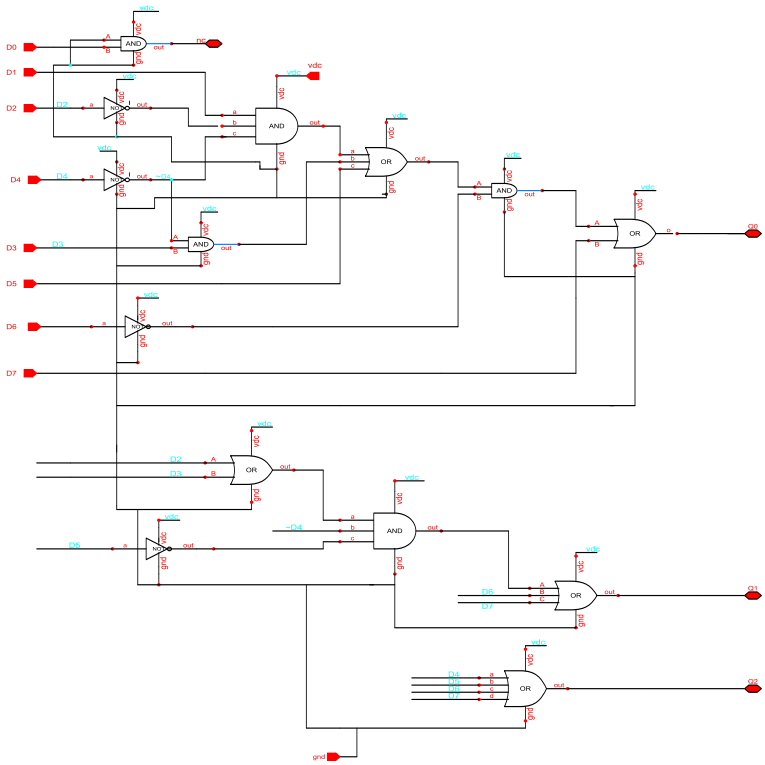

Fig. 7 Gate level schematic implementation of 8:3 priority encoder

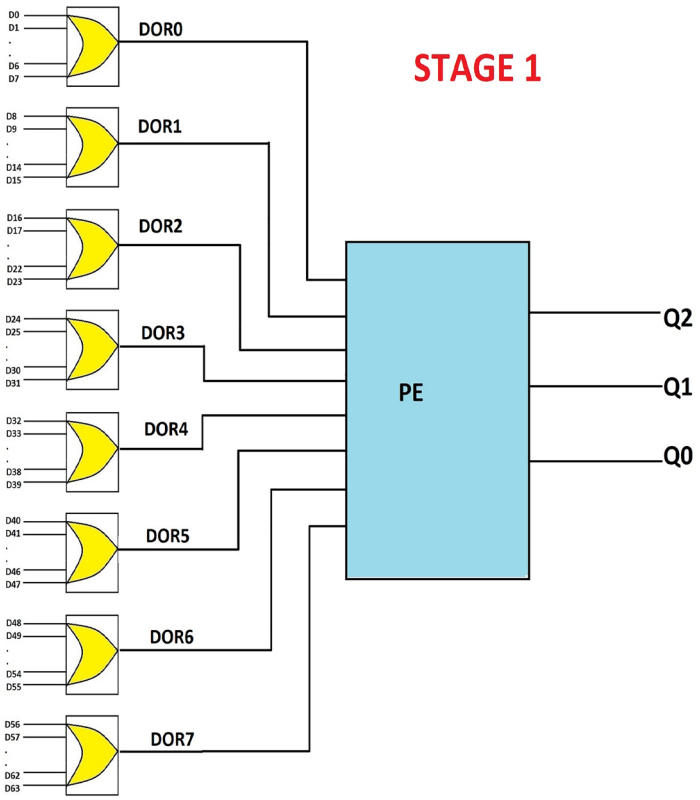

Fig. 8 Priority encoder data-OR (DOR) in the stage 1 (Nguyen et al. 2017)

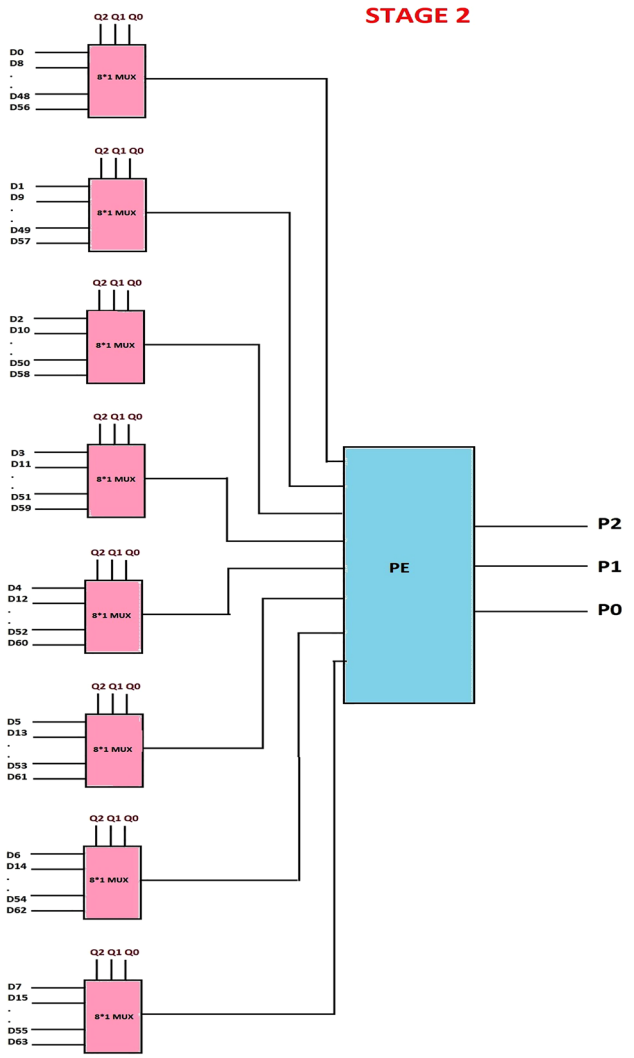

Fig. 9 Priority encoder multiplexer in the stage 2 having 64-inputs and 3-outputs with selection lines coming from first stage (Nguyen et al. 2017)

4.1 PE64*6 priority encoder architecture

To implement the whole circuit, separate rows and columns blocks are to be implemented. At first, 64 inputs are given parallel to eight 8 input OR gates as shown in Fig. 5 which will help to identify in which group data is present. Priority Encoder Data-OR (DOR) in stage 1 having 64- inputs and 3-outputs is depicted in Fig. 8. There will be eight 8 input OR gates and the sequence of inputs are given as D0, D1, D2, D3···D7 for the first 8 input OR gate. Likewise, all inputs are given ranging from D0 to D63. The output lines are then given to an 8*3 priority encoder.

The three output lines of the priority encoder are actually rows of the 2D matrix. After implementation of the row, for columns, 64 inputs are parallel given to eight 8X1

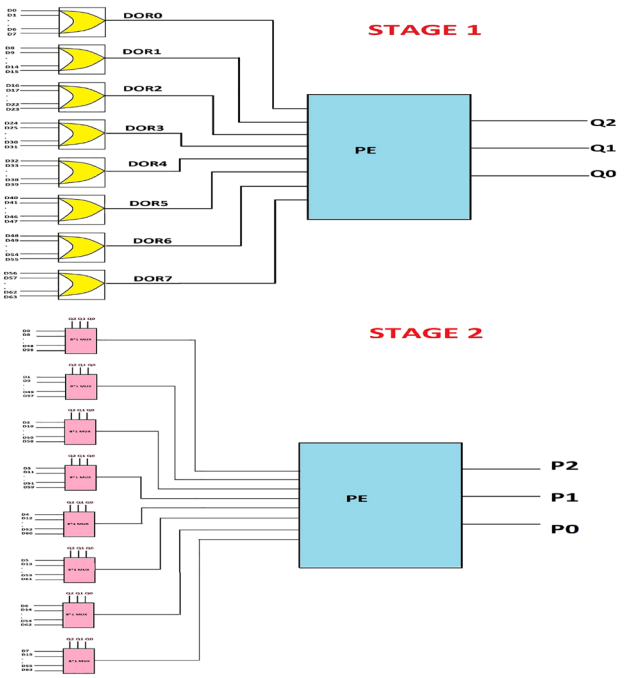

Fig. 10 Combined stages of the Priority Encoder PE64 (Nguyen et al. 2017)

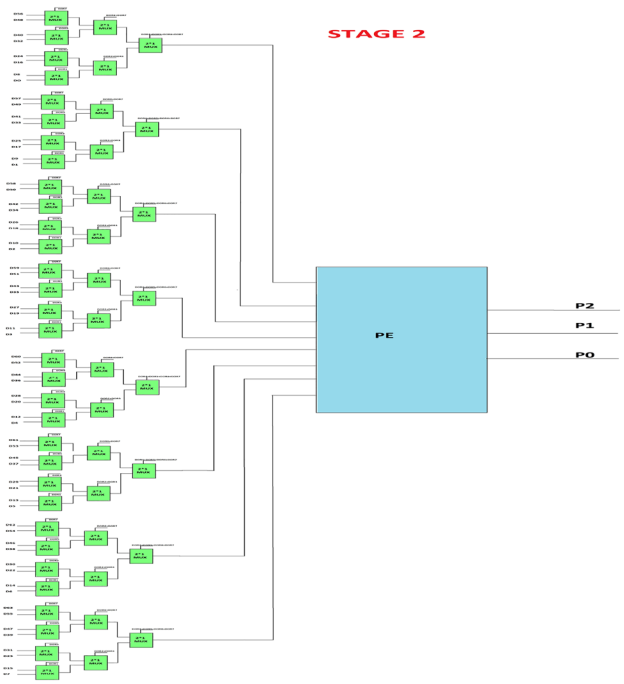

Fig. 11 Priority encoder multiplexer in the stage 2 having 64-inputs and 3-outputs with selection lines coming out from DOR stage (Nguyen et al. 2017)

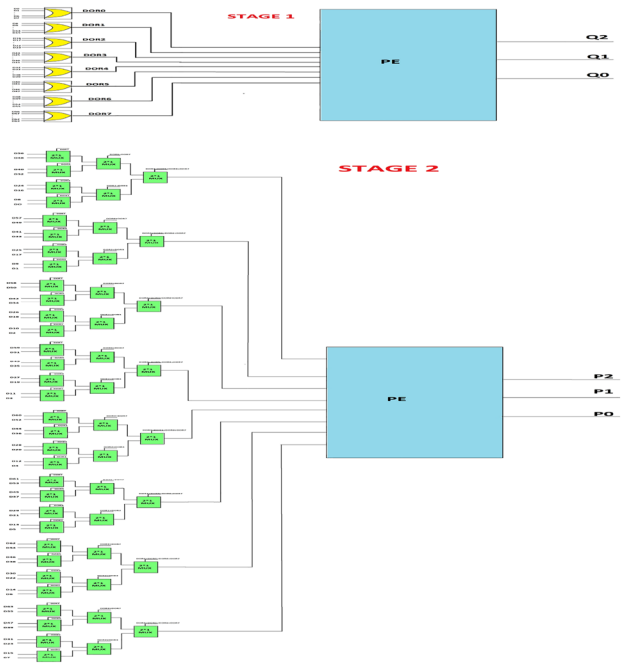

Fig. 12 Combined stages of the Priority Encoder PE64 when selection lines coming out from DOR stage

multiplexers. The output lines of rows are used as select lines for multiplexers. Priority Encoder Multiplexer in the stage 2 having 64-inputs and 3-outputs with selection lines coming from the first stage are shown in Fig. 9. The sequence of inputs to the first 8:1 multiplexer are given as D0, D8, D16, D24....so on.

Likewise, inputs to the 2nd multiplexer are given as D1, D9, D17....so on. In this pattern, all inputs ranging from D0 to D63 are given to multiplexer. The output lines of the multiplexer are given to 8*3 priority encoder which will be the column of the 2 D matrix.

4.2 Modified PE64*6 architecture

The 3 output lines of rows and 3 output lines of columns are merged to form 6 output lines which will be the final output of 64*6 priority encoder. Figure 10 clearly illustrates the combined stages of the Priority Encoder PE64. In the conventional PE64 circuit, it is clear that for the column circuit to operate, it has to wait till the row circuit operation is completed. This is because the select lines for multiplexer in the column stage are actually the row lines i.e. output of the priority encoder. So to reduce the delay of the circuit, a different approach is taken. In this modified circuit, the select lines are taken as the output lines of 8 input OR gates. In this circuit, 64 inputs are given to eight 8 input OR gates as given in the conventional circuit. The outputs of eight 8 input OR gates are named as DOR0, DOR1, DOR2...DOR7. The main difference comes in the multiplexer stage. Figure 11

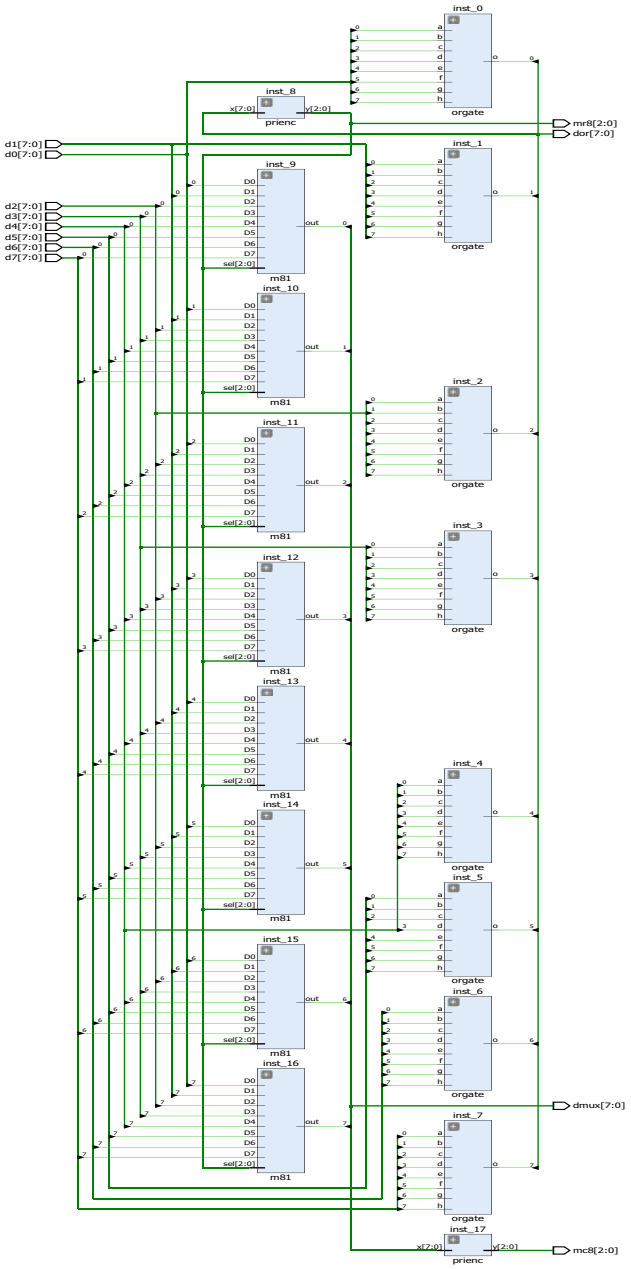

Fig. 13 64×6 Priority Encoder using 2-dimensional array pre-synthesis simulated schematic

shows Priority Encoder Multiplexer in the stage 2 having 64-inputs and 3-outputs with selection lines coming out from the DOR stage. The multiplexer stage is divided into 3 stages and in each stage, there will be different select lines. The select lines of 1st stage of the multiplexer are DOR7, DOR5, DOR3, and DOR1. The select lines of 2nd stage of the multiplexer are DOR6+DOR7. The select lines of 3rd stage of the multiplexer are DOR4 +DOR5 +DOR6+DOR7. The eight lines of output of multiplexer are connected to 8:3 priority encoder. Hence column is identified and merged

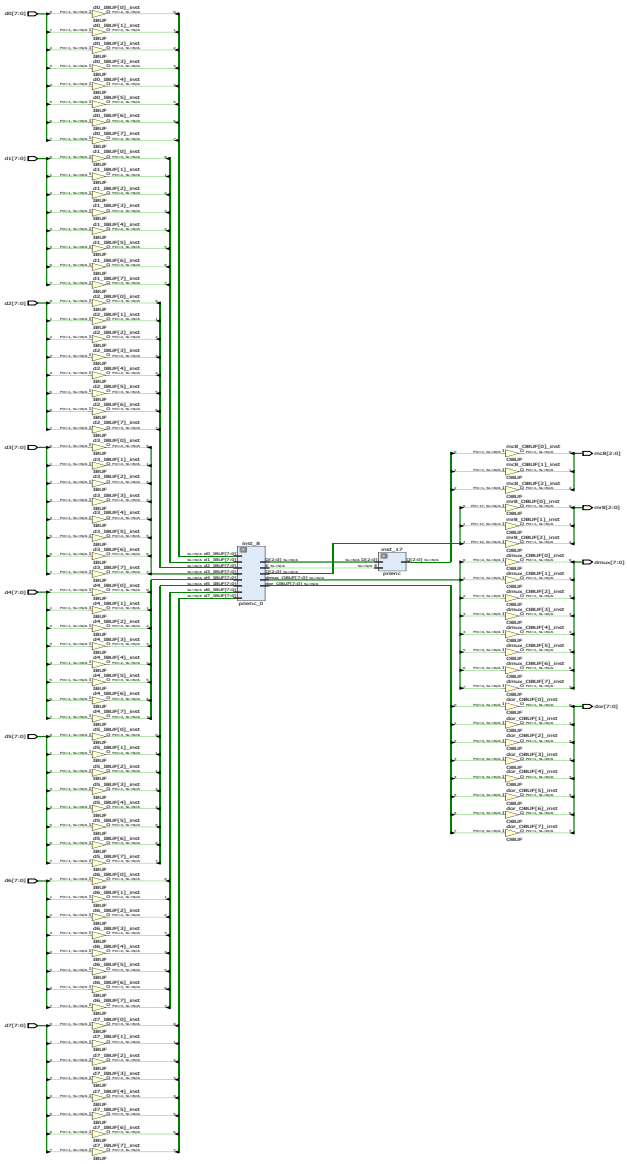

Fig. 14 64×6 Priority Encoder using 2-dimensional array post-synthesis simulated schematic

with row to give 6 line output of 64X6 priority encoder. These 6 lines of output are the actual encoded bits of the data given to the input of the 64X6 Priority Encoder. Combined stages of the Priority Encoder PE64 when selection lines are coming out from DOR stage is depicted in Fig. 12. In this circuit, the delay is reduced and hence performance of the circuit is increased.

5 Simulated result

In this section, all the experimented and implemented results are being discussed. The same priority encoder is implemented using Verilog and synthesized in Artix FPGA board using Vivado tool, as well as in Cadence virtuoso to count the number of transistors in the priority encoder. As we know FPGA only shows the number of LUTs and flip-flops. FPGA board is fabricated at less than 180nm so to verify at 180nm implemented on cadence virtuoso. After implementation of the priority encoder, the number of transistors, Delay, and power consumed by the circuit has been compared to investigate the performance of the architecture.

5.1 Verilog implementation

Figure 13 is resulted after running a behavior simulation of the 2-dimensional priority encoder Verilog code. It is observed from the schematic that only inputs, outputs, multiplexers, OR gates, and priority encoders are depicted. Figure 14 depicts the post-synthesis schematic of the 2- dimensional array of the 64-bit priority encoder. After running the synthesis of the Verilog code priority encoder. Post synthesis design not only shows the multiplexers, OR gates, and priority encoders but also buffers at the inputs and outputs ports. In Fig. 15, most of the input combinations are given to investigate the performance of the priority encoder. In each of the given cases, the top priority of this priority encoder is successfully detected.

5.2 Cadence virtuoso Implementation

Figure 16 shows the full custom circuit design implemented using cadence virtuoso. All the 64 inputs are clearly shown at each of the OR gate and Multiplexers. The output of the OR gate and multiplexer are feed as input to the priority encoders. Finally, the Output of the 8:3 priority encoder arranges in such a manner to work as 6-bit output.

In Fig. 17, the Modified priority encoder of 64-bit is shown. This architecture includes the extension of multiplexer due to DOR selection lines. The output waveform of the implemented circuit in the cadence virtuoso is clearly depicted in Fig. 18. All the 64-bit input are given in the waveforms and their input/output is also reflected accordingly. It is observed from Table 1 that 1000 transistors have been saved by 2-D Priority encoder as compared to PE64X6 and 1026 transistors have been saved as compared to Modified PE64X6 and the performance of the encoder is also increased in terms of the the delay involved from input to output.

The Verilog implementation data is tabulated in Table 2. In this table, the total number of lookup tables, number of flip-flops, and number of input/output involved to implement this 64-bit priority encoder have been shown. The percentage of utilization of the hardware of the Artix-7 FPGA board is also tabulated

The total power consumed in the Artix-7 FPGA board is tabulated in Table III. This table clearly shows

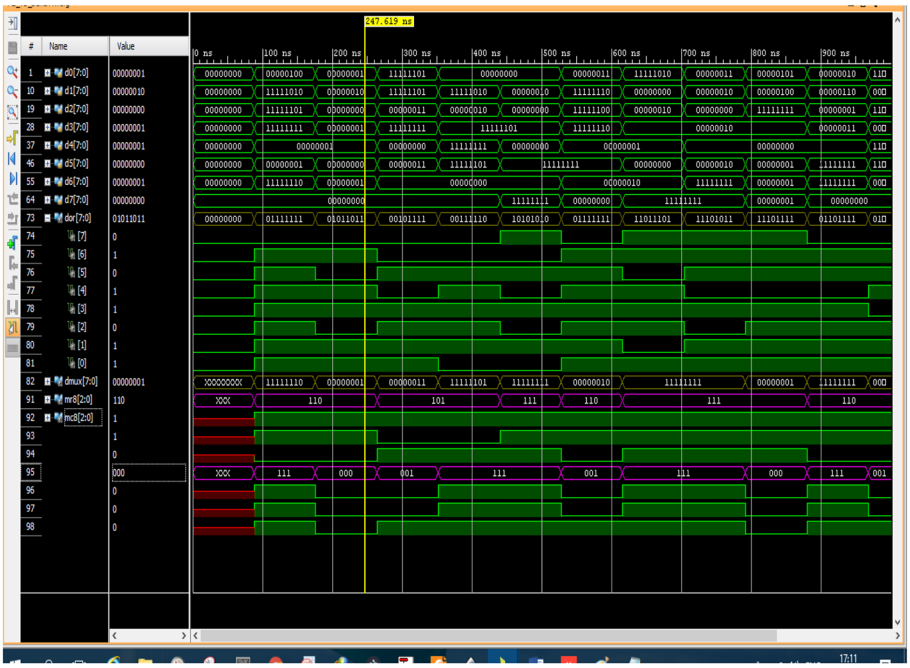

Fig. 15 Output waveform of the 64-bit Priority Encoder at different data input

Table 1 Observation of priority encoder PE64 CMOS 180 nm technology

	2-D PE64×6	Modified 2-D PE64×6	PE64×6 (Nguyen et al. 2017)	Mod. PE64×6 (Nguyen et al. 2017)
No. of Transistors	1128	1438	2128	2464
Output delay (ps)	1123.5	844.11	2970	2203
Avg. power (uW)	127.5	96.28	-	
PDP (fJ)	143.27	81.27	-	

Table 2 Observation of PE64 on FPGA ARTIX-7 board

Resource	Utilization	Available	Utilization %
Look up tables	46	20800	0.22
Number of FF	6	41600	0.01
Input/output (I/O)	86	106	81.13

Table 3 Power observation of PE64 on FPGA ARTIX-7 board

Dynamic power	Power Value	97%
Signal	0.412 W	14%
Logic	0.118 W	4%
Input/output (I/O)	2.428 W	82%
Device static power	0.079 W	3%

the dynamic power (97%) as well as static power (3%). Dynamic power is also bifurcated in terms of signal power, logic power, and input/output power in watts.

6 Conclusion

In this paper, a 2-dimensional array has been implemented and evaluated. By reducing the output OR gates, PE, and shifter increase the performance of the 64- bit priority encoder. This work has been implemented using Vivado. To count the number of transistors this is also implemented using cadence virtuoso. The architecture analysis has been performed for the higher performance of the priority encoder. The performance of the modified 2-D priority encoder is increased more than a minimum of 61.68 % as compared to state-of-art.

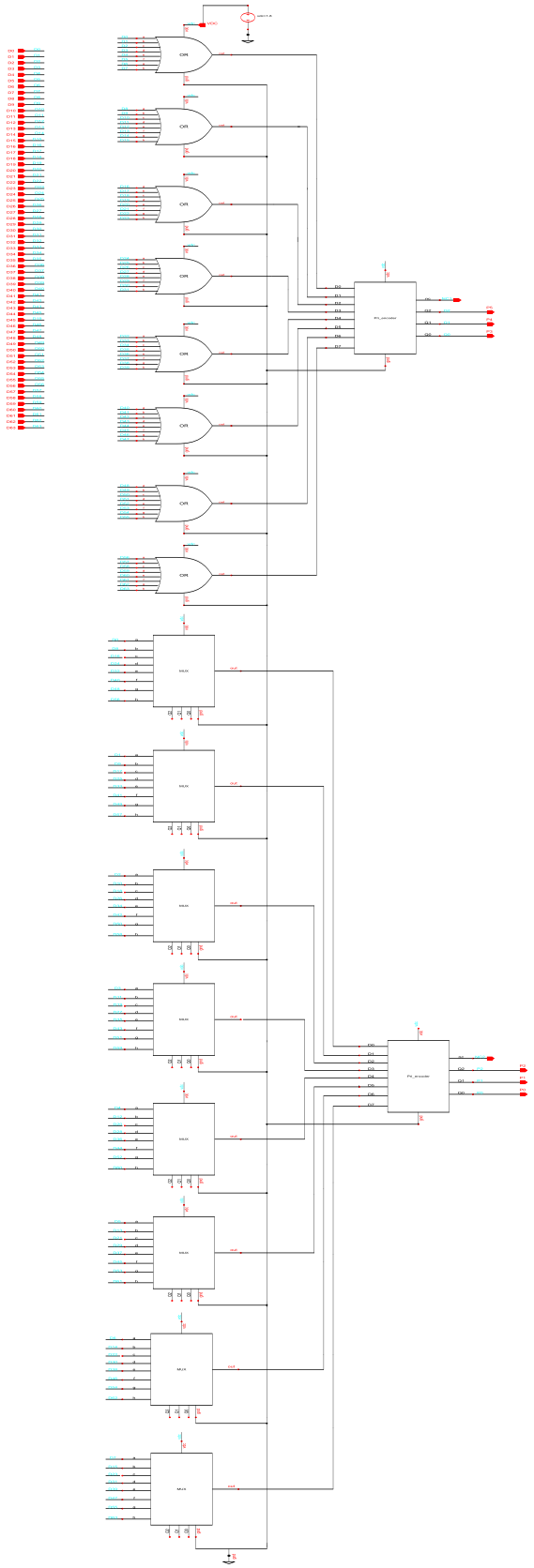

Fig. 16 64-bit Priority Encoder using 2-dimensional array implemented in cadence Virtuoso

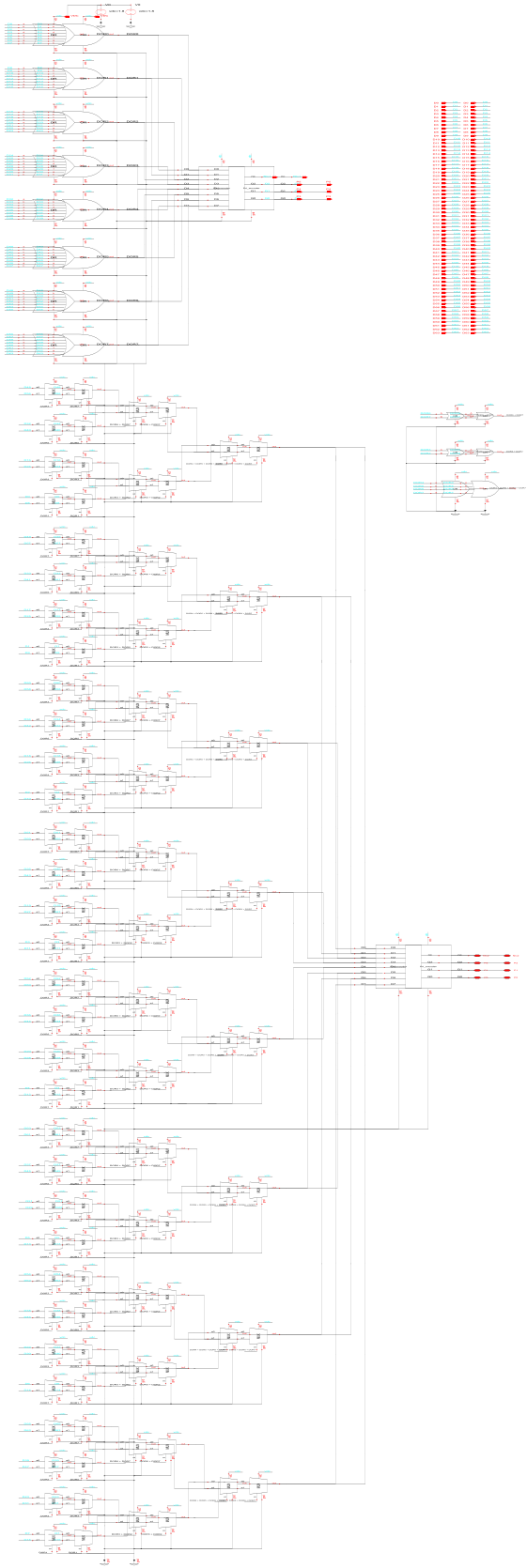

Fig. 17 2-D architecture of modified 64×6 priority encoder

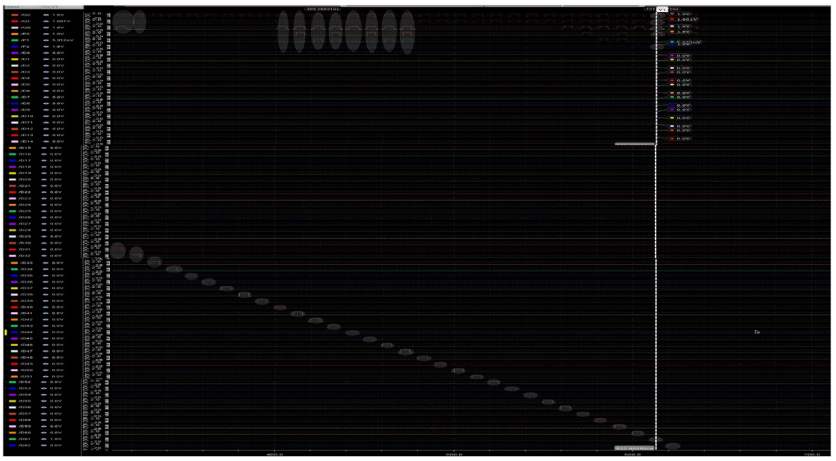

Fig. 18 Cadence virtuoso Implemented 64×6 Priority Encoder using 2D array output waveforms

Funding Not Applicable.

Declarations

Conflict of interest Author declares that they have no conflict of interest.

Informed consent N/A.

Research involving human participants and/or animals N/A.

References

Balobas D, Konofaos N (2017) High-performance and energy-efficient 256-bit cmos priority encoder. In: 2017 IEEE computer society annual symposium on VLSI (ISVLSI), pp 122–127 IEEE

Balobas D, Konofaos N Low-power, high-performance 64-bit cmos priority encoder using static-dynamic parallel architecture. In: 2016 5th International conference on modern circuits and systems technologies (MOCAST), pp 1–4 (2016). IEEE

Bao Z, Guo Y, Li X, Zhang Y, Xu M, Luo X (2020) A robust image steganography based on the concatenated error correction encoder and discrete cosine transform coefficients. J Ambient Intell Human Comput 11(5):1889–1901

Das S, Sen R, Dutta S, Banerjee S, Thakuria B, Jangid A, Basu R (2017) Design and simulation of all optical 4 to 2 priority encoder at 40gbps using xgm in wideband travelling wave semiconductor optical amplifier without additional input beam. In: 2017 8th IEEE annual information technology, electronics and mobile communication conference (IEMCON), pp 314–318. IEEE

Delgado-Frias JG, Nyathi J (2000) A high-performance encoder with priority lookahead. IEEE Trans Circuits Syst I Fund Theory Appl 47(9):1390–1393

Ghosh K, Haque MM, Chakraborty S (2016) Design of reversible ternary adder/subtractor and encoder/priority encoder circuits. In: 2016 International conference on communication and signal processing (ICCSP), pp 1290–1295. IEEE

Hashemian R (1989) A high speed compact priority encoder. In: Proceedings of the 32nd midwest symposium on circuits and systems, pp 197–200 IEEE

Huang SW, Chang YJ (2010) A full parallel priority encoder design used in comparator. In: 2010 53rd IEEE international midwest symposium on circuits and systems, UK, pp 877–880. IEEE

Huang C-H, Wang J-S, Huang Y-C (2002) Design of high-performance cmos priority encoders and incrementer/decrementers using multilevel lookahead and multilevel folding techniques. IEEE J Solid State Circuits 37(1):63–76

Kathuria J, Sharma M (2019) Novel tree based priority encoder design technique. In: 2019 International conference on computing, power and communication technologies (GUCON), pp 593–599, IEEE

Kumar VC, Phaneendra PS, Ahmed SE, Sreehari V, Muthukrishnan NM, Srinivas M (2011) A reconfigurable inc/dec/2's complement/priority encoder circuit with improved decision block. In: 2011 International symposium on electronic system design, pp 100–105 IEEE

Kumar S, Bisht A, Amphawan A (2015) Four bit priority encoder using lithium niobate based mach-zehnder interferometers. In: 2015 Workshop on recent advances in photonics (WRAP), pp 1–4. IEEE

Kun C, Quan S, Mason A (2004) A power-optimized 64-bit priority encoder utilizing parallel priority look-ahead. In: 2004 IEEE international symposium on circuits and systems (IEEE Cat. No. 04CH37512), vol 2, p 753, IEEE

Le D-H, Inoue K, Sowa M, Pham C-K (2012) An fpga-based information detection hardware system employing multi-match content addressable memory. IEICE Trans Fund Electronics Commun Comput Sci 95(10):1708–1717

Maurya SK, Clark LT (2010) A dynamic longest prefix matching content addressable memory for ip routing. IEEE Trans Very Large Scale Integration (VLSI) Syst 19(6):963–972

Nguyen XT, Hoang TT, Nguyen HT, Inoue K, Pham CK (2018) A 219-μw 1d-to-2d-based priority encoder on 65-nm sotb cmos. In: 2018 IEEE international symposium on circuits and systems (iSCAS), pp 1–4. IEEE

Nguyen X-T, Nguyen H-T, Pham C-K (2017) A scalable high-performance priority encoder using 1d-array to 2d-array conversion. IEEE Trans Circuits Syst II Exp Briefs 64(9):1102–1106

Rabeya M, Mahmood M, Das B, Bardhan R, Tareque MH (2019) An efficient design of 4-to-2 encoder and priority encoder based on 3-dot qca architecture. In: 2019 International conference on electrical, computer and communication engineering (ECCE), pp 1–6. IEEE

Saidi H, Turki M, Marrakchi Z, Abid M, Obeid A (2021) Soft-core embedded fpga based system on chip. Analog Integrated Circuits Signal Process, pp 1–17

Sen R, Das S, Mazumder GG, Yadav P, Neogy B, Pandey R, Sharma S, Jana B (2017) Priority encoder using reversible logic gates in qca. In: 2017 8th IEEE annual information technology, electronics and mobile communication conference (IEMCON), pp 319–323. IEEE

Tsai HJ, Yang KH, Peng YC, Lin CC, Tsao YH, Chang MF, Chen TF (2017) Energy-efficient tcam search engine design using priority-decision in memory technology. IEEE Trans Very Large Scale Integration (VLSI) Syst 25(3):962–973

Wang JC, Pang Y, Xia Y A bcd priority encoder designed by reversible logic. In: 2012 International conference on wavelet active media technology and information processing (ICWAMTIP), pp 318–321 (2012). IEEE

Yershov RD A scalable vhdl-implementation technique of the priority encoder structure into fpga. In: 2018 IEEE 38th international conference on electronics and nanotechnology (ELNANO), pp 727–732 (2018). IEEE