Nuclear Engineering and Technology 53 (2021) 3406–3412

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

A real-time sorting algorithm for in-beam PET of heavy-ion cancer therapy device

Lingyun Ke [a, b], Junwei Yan [a], Jinda Chen [a], Changxin Wang [a], Xiuling Zhang [a], Chengming Du [c], Minchi Hu [a, b], Zuoqiao Yang [a, b], Jiapeng Xu [a, b], Yi Qian [a], Qianshun She [a], Haibo Yang [a], Hongyun Zhao [a], Tianlei Pu [a], Changxu Pei [a, d], Hong Su [a, b], Jie Kong [a, b, *]

[a] Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou, 730000, China
[b] School of Nuclear Science and Technology, University of Chinese Academy of Sciences, Beijing, 100049, China
[c] School of Electronic Engineering, Chaohu University, Hefei, 238000, China
[d] School of Nuclear Science and Technology, Lanzhou University, Lanzhou, 730000, China

ARTICLE INFO

Article history:
Received 14 September 2020
Received in revised form
1 February 2021
Accepted 6 April 2021
Available online 13 April 2021

Keywords:
Real-time sorting algorithm
FPGA
Coincidence events identification
In-beam PET

ABSTRACT

A real-time digital time-stamp sorting algorithm used in the In-Beam positron emission tomography (In-Beam PET) is presented. The algorithm is operated in the field programmable gate array (FPGA) and a small amount of registers, MUX and memory cells are used. It is developed for sorting the data of annihilation event from front-end circuits, so as to identify the coincidence events efficiently in a large amount of data. In the In-Beam PET, each annihilation event is detected by the detector array and digitized by the analog to digital converter (ADC) in Data Acquisition Unit (DAQU), with a resolution of 14 bits and sampling rate of 50 MS/s. Test and preliminary operation have been implemented, it can perform a sorting operation under the event count rate up to 1 MHz per channel, and support four channels in total, count rate up to 4 MHz. The performance of this algorithm has been verified by pulse generator and ^{22}Na radiation source, which can sort the events with chaotic order into chronological order completely. The application of this algorithm provides not only an efficient solution for selection of coincidence events, but also a design of electronic circuit with a small-scale structure.

1. Introduction

Heavy ion therapy is considered as one of ideal radiotherapy method in the treatment of cancer for its unique physical and biological characteristics. In order to implement an in-vivo and non-invasive monitoring of the treatment dose distribution, and verification of beam position and dose, An In-beam positron emission tomography (In-beam PET) system located on the beam line and at the treatment site is developed for the Heavy-Ion cancer Therapy Device (HICTD), which has been carried out by the Institute of Modern Physics of the Chinese Academy of Sciences (IMP-CAS) [1–3].

PET is a diagnostic system based on the molecular imaging technology, an observing of metabolic activity in vivo can be performed with it [4–9]. Present, it has been widely used in the diagnosis, condition judgment, efficacy evaluation of cancer and other diseases. The identification method for coincidence events generated from positron-electron annihilation events at the irradiation area in ion beam therapy is one of key techniques, which is used for extraction of valid information, particle energy, time, position, etc [10,11]. The identification method and electronics employed in general PET system operating under the relative low or medium count rate are challenged by the requirements of real-time and low cost for the coincidence and identification system utilized in In-beam PET.

There are two main implementation methods for traditional coincidence system, one is based on the AND gate, called AND-gate based architecture; another is based on the time-stamp technique,

[*] Corresponding author. Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou, 730000, China.
E-mail address: kongjie@impcas.ac.cn (J. Kong).

https://doi.org/10.1016/j.net.2021.04.006

called time-stamp based architecture. The former has been widely used in coincidence system recently as its lower cost and high efficiency. However, it is highly sensitive to the scale and structure of the front-end electronics (FEEs), the application flexibility of this method is limited as the geometric scale of customized FEEs [12]. The Time-stamp based architecture is gathering more attention lately, especially in the application of fully digital system with multiple channels and high count rate. At this situation, the data of all events detected from per channel is recorded, and then, the postpone coincidence identification processing could be done in host computer with the data from all channels.

A new digital sorting algorithm based on the time-stamp technique is presented in this paper, which is developed for the Data Acquisition Unit (DAQU) of In-beam PET. This sorting algorithm can effectively sort all events needed to be rearrange in chronological order, and makes it easy to choose the coincidence events in a valid time window. An ideal processing method is supplied for identifying coincidence events in a real time system [13], and performing the sorting of events under high count rate. The sorting method has been developed successfully based on a FPGA. The preliminary experimental verification shows that the digital sorting algorithm can effectively perform the events sorting and coincidence events identification in real time in the In-Beam PET of HICTD.

2. Description of system

As shown in Fig. 1, the prototype of In-Beam PET system consists of two parts, a fast scintillation detector, and a high performance readout electronics system. The detector part consists of four detector units (DUs), and readout electronics system contains four Data Acquisition Units (DAQUs). The Data Acquisition Unit (DAQU) is the core part of readout electronics system. DAQU is mainly composed of the pre-processing circuits for energy signals, the pre-processing circuits for timing signal, analog to digital convertors (ADCs), time to digital convertors (TDCs) designed based on a FPGA, and some digital signal processing algorithms loaded in the FPGA [14–17]. The linear sorting algorithm is a main algorithms for coincidence discrimination.

DUs in the prototype of In-Beam PET system are the detector arrays, which have been employed for detecting the gamma rays radiating from positron annihilation. Each DU is made up of four crystal blocks, and each crystal block is composed of 22 × 22 LYSO crystals with the dimension of 2 mm × 2 mm × 15 mm. An 8 × 8 multi-anode position sensitive photomultiplier tube (PSPMT) is coupled with each crystal block, and a DPC (discretized positioning circuit, DPC) circuit is matched to PSPMT to reduce its 64 outputs to 4 outputs as energy signal outputs (named as E output) [18–20]. Another signal from the last dynode of PSPMT (named as T output) is used for time measurement. E outputs are digitized by the ADCs in DAQU, and T output is processed by the TDC module in FPGA. Digitized energy signals and time signal from each DU are processed and organized into an event data frame in a fixed frame format, which is also stamped with a time-stamp under 100 MHz system clock. The linear sorting algorithm developed by us acquires these event data frames from all DUs, and sorts them according to the timestamps [21,22]. Then the sorted event stream is provided for a fast identification of the γ-ray pairs generated by the annihilation event during the time window configured, and further the image reconstruction of In-Beam PET online is supported.

3. Schematic and implement

In readout electronics of In-Beam PET, it is an ideal situation for real-time coincidence that the data frames of events are output in chronological order. However, the energy measurement and time extraction for each event are performed by electronics channels of DAQU individually. After signal processing, digitizing, data packaging and multiplexing output, the data frames of events detected by the DU can't be sent out from DAQU in chronological order of occurrence strictly, especially in the situation of high count rate. Therefore, the improved linear sorting algorithm is implemented for the ordering correction in In-Beam PET, which is benefit for identification of the effective coincidence events online under high event count rate.

3.1. Design architecture of sorting algorithm

Considering the multi-channel accessing in DAQU, the sorting algorithm with two levels is implemented, shown as Fig. 2.

A token ring structure is adopted as the first level, which accesses sequentially to each processing channel of the front-end in DAQU, and outputs the event data in the principle of first in first out. The token ring structure is used in our design to access different channels. The token is generated when at least one First Input First Output (FIFO) is not empty in channels, it comes out from the token manager circuit and starts to travel in the ring. Once

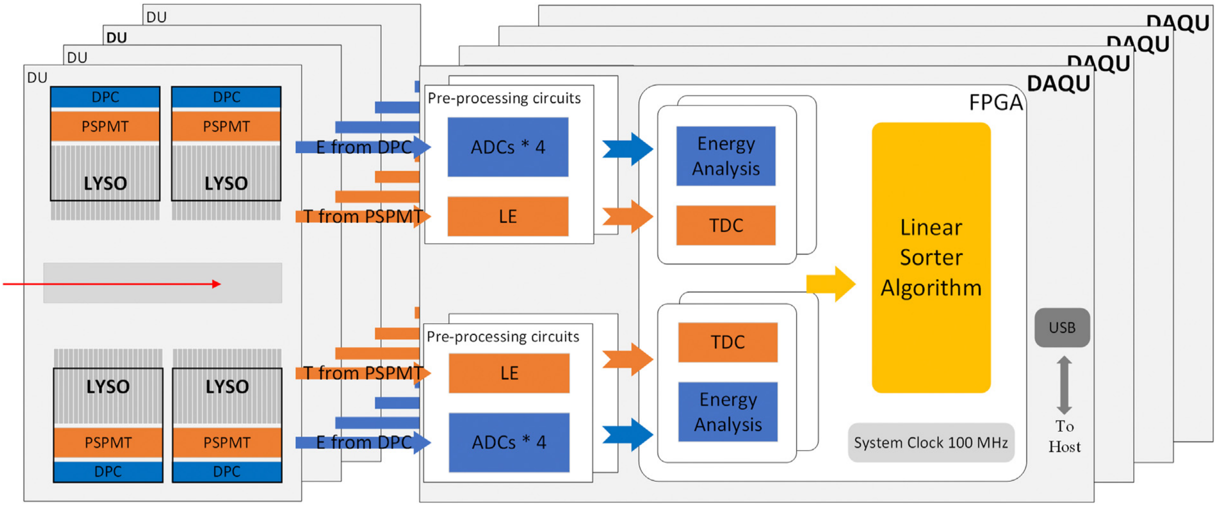

Fig. 1. The block-diagram of prototype of In-Beam PET system.

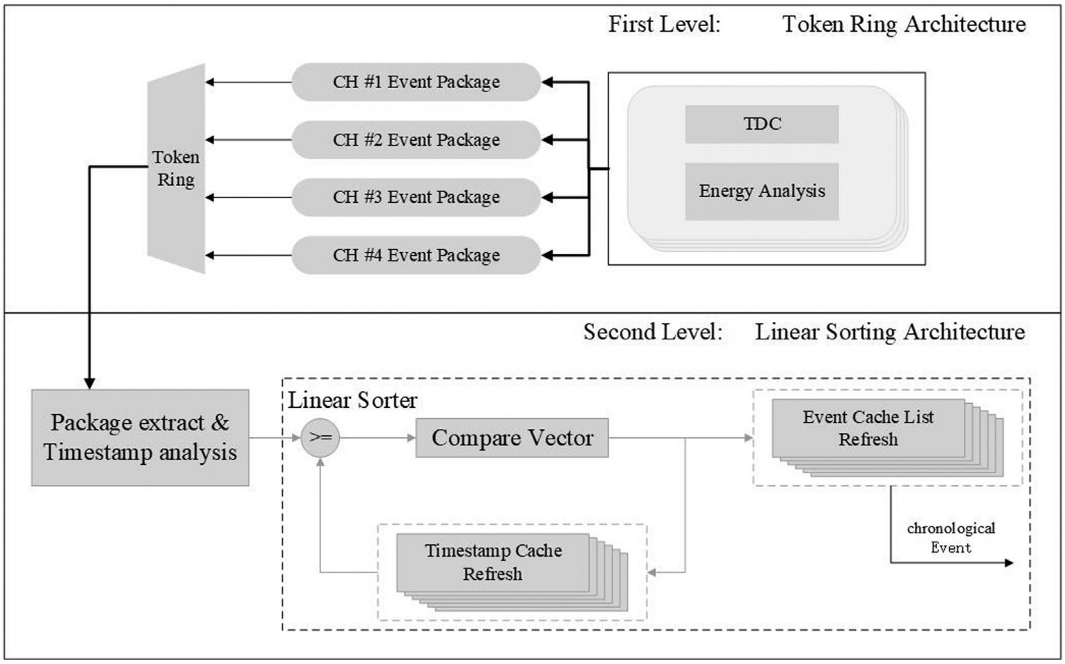

Fig. 2. Architecture of sorting algorithm.

the token accesses one channel which contains data, it stops and reads the data from FIFO of this channel. Then, at the next read out cycle, it goes on and stops at the next channel where there is data also. If there is none channel which has data, the token returns to the manager. When the token manager recognizes the condition, i.e. at least one FIFO has data, it restarts this procedure and repeats all the steps mentioned above. Therefore, the multiple inputs are converted into one channel data stream by the token ring.

Linear sorting architecture is utilized as the second level. As shown in Fig. 2, it is mainly composed of package extract module, timestamp analysis module, and linear sorter module. At this level, the data stream from the first level is sorted in real time by the sequential caching, the schematic of this method is shown in Fig. 3.

When a new event data enters linear sorting unit, its timestamp information is extracted by the package extract module. Then, timestamp of the new event is analyzed by the timestamp analysis module and compared with time stamp of each event in the cache, and a feature vector corresponding to the caching sequence is obtained. The feature vector is a judgment basis for sorting the events, it is gotten by comparing the timestamp of the new event and the timestamps of the events in the cache. Both the refresh action in time-stamp cache and event cache are controlled by the same feature vector. As the events in cache are arranged sequentially, here arranged in a monotonous decrease sequence from left to right, the cache and corresponding feature vectors have the following characteristics:

As shown in Fig. 3, the events (Ei, i = 1–16) in Event Cache have the corresponding timestamps (Si, i = 1–16) arranged in Timestamp Cache. The events with the timestamps larger or equal than the timestamp of new event (Sin), are concentrated on the left side, the corresponding vector value is set to ‘1’, and the events with smaller timestamps are concentrated on the right side accordingly, the vector value is set to ‘0’. Since the timestamp is arranged to decrease monotonically, the event at most right end of cache is event with the smallest timestamp.

According to the feature vector, the position of the new event in

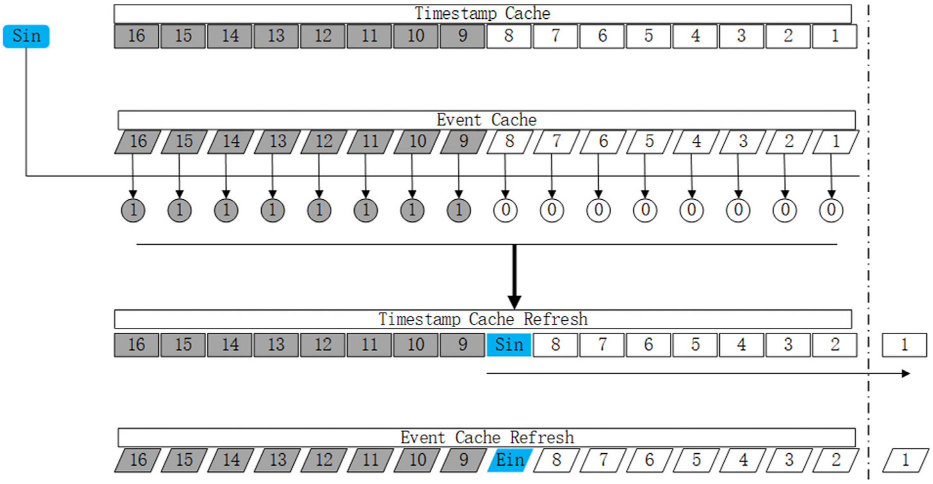

Fig. 3. Schematic of linear sorting method.

the cache can be determined, the new event and its timestamp are inserted into Event Cache and timestamp cache and the event with smallest timestamp in the cache is output with the shift operation, meanwhile the reorganization of the event and timestamp sequence in cache are completed. As an example shown in Fig. 3, it can be judged that the new event Sin generated before S9 and after S8, its position in the Timestamp Cache should be between S9 and S8. Correspondingly, the data of new event is plugged in position between E9 and E8 in Event Cache. All events with a vector value of ‘0' (S8, S7, …, S1) in the cache are shifted to the right by one cache unit synchronously, while events with a vector value of ‘1' (S16, S15, …, S9) are remained in its place, the vacated cache unit is used for placing the new event. At the same time, the minimum event timestamp S1 is shifted to right and an overflow is generated, which is used as a cache output and employed for subsequent coincidence identification.

The single processing cycle of the linear sorting algorithm mainly depends on the inherent operating time of combine logic for comparing the timestamps of new input event and events in cache. The rearrangement of the event sequence can be performed synchronously with input of the new event in this design, the unnecessary time overhead is avoided.

3.2. Implementation of linear sorting algorithm

In the design of DAQU of In-Beam PET, as the data width of transmission is 16 bits between the FPGA and the host computer through the Small Form-factor Pluggable (SFP) interface, as a result, 18 clocks are required for transmitting one frame of data with 18*16 bits. In the design of sorting method, an online data sorting without dead time can be achieved by utilizing the time of data transmission. The principle structure of the real-time sorting algorithm used in DAQU is shown in Fig. 4.

When a frame of data enters the shift register group, the shift register group performs the identification of the data packet by counting the header, tail and packet length, also extracts the timestamps as well. During the data packet is confirmed, meanwhile, the timestamp comparison between the new events and the events in cache is started. Since the timestamp length is 64 bits, 20–30 ns of inherent time is needed for the combine logic running of comparison module in the actual test. The data packet is processed in three clocks in the design. As long as the recovery and reorganization of Timestamp Cache is completed before the next data packet arriving, the next sorting can be guaranteed to be executed normally. After three clock periods, the time relationship of new entering event and events already existing in cache can be determined by judging the Feature Vector generated. As mentioned above, the event with a timestamp that is larger than the timestamp of new event is remained in its original position in cache, the all events with the timestamps that are smaller than the timestamp of new event are shifted by one cache unit synchronously, the event with smallest timestamp is output, and the new event is inserted into the empty unit in cache, now, the sorting and cache refresh are completed.

For the sorting of timestamps can be completed as soon as possible, the registers are employed to store the timestamp information in the design. Thus, the sequence recovery of timestamps can be completed in one clock after the Feature Vector generated.

Because a complete data packet of the event in DAQU is relatively large, a cache with the memory structure is adopted to store the event data so as to reduce the utilization quantity of registers. The memory width is 16 bits, the depth is equal to the packet length, and the address corresponds to the format sequence of packet. When Feature Vector arrives, the value of Vector is used to determine which RAM needs to be read and written. All RAM blocks share the same address for reading and writing. The RAM is used for the new event, when the header of data package of the new event is imported, at the same time, the original data of the current address is read out and written to the corresponding address in the next-level RAM. For the RAM where the shift operation is needed to be performed, when the data is written in the corresponding address, meanwhile, original data in this address is read out first, then, the data shift operation in each RAM can be completed simultaneously within a clock. The data in each address of RAM is transferred sequentially by the shift operation, with entering of the new data, the update and reorganization of the event cache, and output of event data with minimum timestamp is completed synchronously.

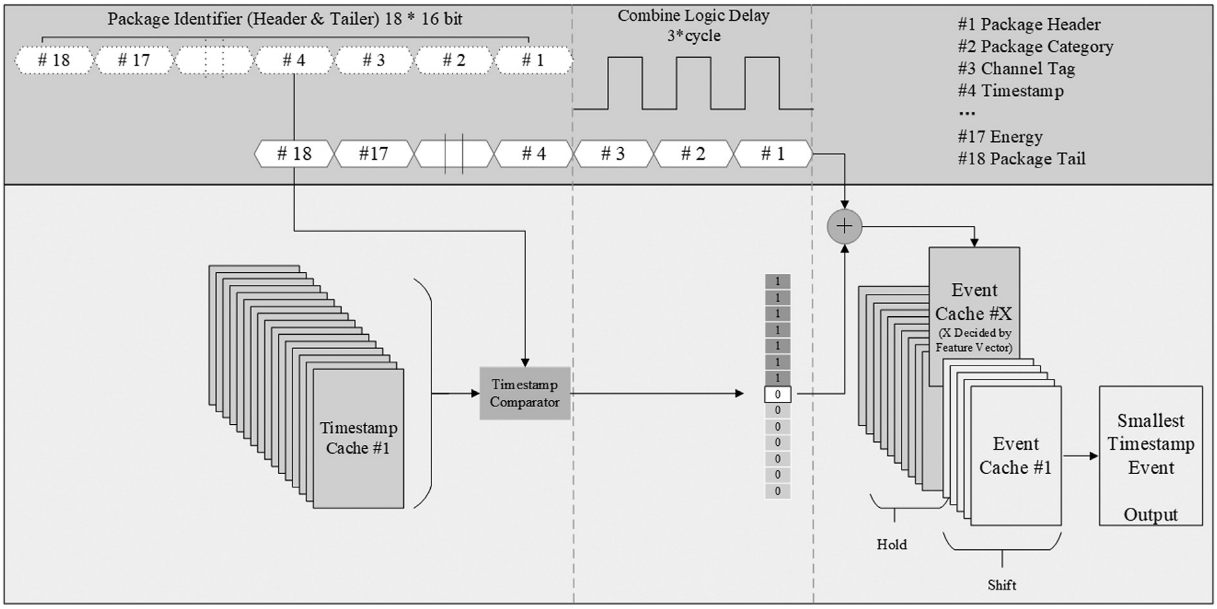

Fig. 4. Principle structure of the real-time sorting algorithm in DAQU.

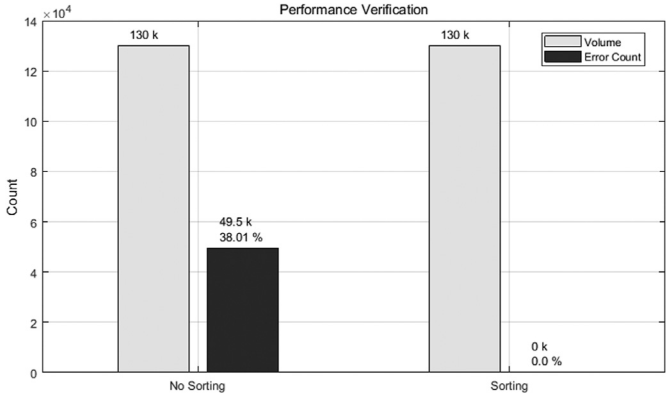

Fig. 5. Performance verification of no sorting vs. sorting in DAQU.

4. Experiment and test

In order to analysis and evaluate the application performance of this sorting algorithm, some verification experiments have been implemented under the conditions of utilizing the digital signal source and radioactive source. The data unsorted and the data sorted from DAQU were collected by the host computer and dealt with MATLAB process tool. The count rate, sorting function and real-time performance of the linear sorting algorithm have been acquired.

4.1. Sorting performance test

As shown in Fig. 5, for the sake of the evaluation of the sorting performance of linear sorting algorithm, the data from DAQU which are unsorted and the data from DAQU which are sorted are acquired and compared under the condition of 10 kHz event rate using a digital pulse generator. The events with wrong ordering in data package under two modes (i.e. unsorted and sorted) are counted up, the gray bar in Fig. 5 is the total event count and the black bar is the statistics of the event in error list. According to analysis, the error events has occupied up to 38.01% of the whole event volume under the un-sorting mode as shown at the left of Fig. 5. The higher error rate is badly for the coincidence event selection. As shown at the right of Fig. 5, there is not error count after the sorting algorithm is adopted in DAQU. The problem, appearing events with wrong ordering, can be avoided.

4.2. Characteristics of count rate

This linear sorting algorithm, loaded in 4 channels of DAQU, is checked at different event rates from 1 kHz to 1 MHz per channel for evaluating its performance. The total number of the events received in each test are 6.5 × 10^6 events. The count rate characteristics of each channel are compared under two modes, i.e. using sorting algorithm or without sorting algorithm, utilizing a digital pulse generator. The relationship curves between event count and the time spent at four different count rates, 1 kHz, 10 kHz, 100 kHz and 1 MHz are presented in Fig. 6.

As shown in Fig. 6.a, because there is not sorting algorithm used in data processing, the wrong ordering events appears, the error rate is above 40%. In fact, for a data processing system without sorting algorithm, the values of error rate gotten from the tests of different times vary randomly in the range of 27%–47% under the input signals with the frequencies from 1 kHz to 1 MHz in our experiments. The curves are thickened to indicate that the data of events with wrong ordering are included in the curves. Other, the relationship curves of event count and the time spent are given in Fig. 6.b, as the sorting algorithm is used in data processing here, there is no an error detected, it means that an error rate of 0.0% is gained in this situation. The detail test results got under two modes are given in Table 1. The results show that the linear sorting algorithm features the excellent capability, it can support the sorting processing at a high count rate up to 1 MHz for each channel of DAQU.

4.3. Combine logic delay

In this design, as the width of timestamp used in the DAQU is 64 bits, the running of the combine logic for timestamp comparison will bring in an inherent delay time. An illustration diagram is given in Fig. 7. The multi-period path constraint method on a single clock domain is applied in this design to meet the requirement of getting the optimum logic delay time. The evaluation test of the valid combine logic delay of comparator has been performed. The data is read out after one period, two periods, and three periods separately after starting the combine logic of timestamp comparator. The sorting results gained through different period paths are showed in Table 2. The experiment results got via the three periods (clk = 100 MHz) path constraint have a lowest error event rate. In other words, 3-period path constraint is required for the comparator logic, which can avoid the error of metastability.

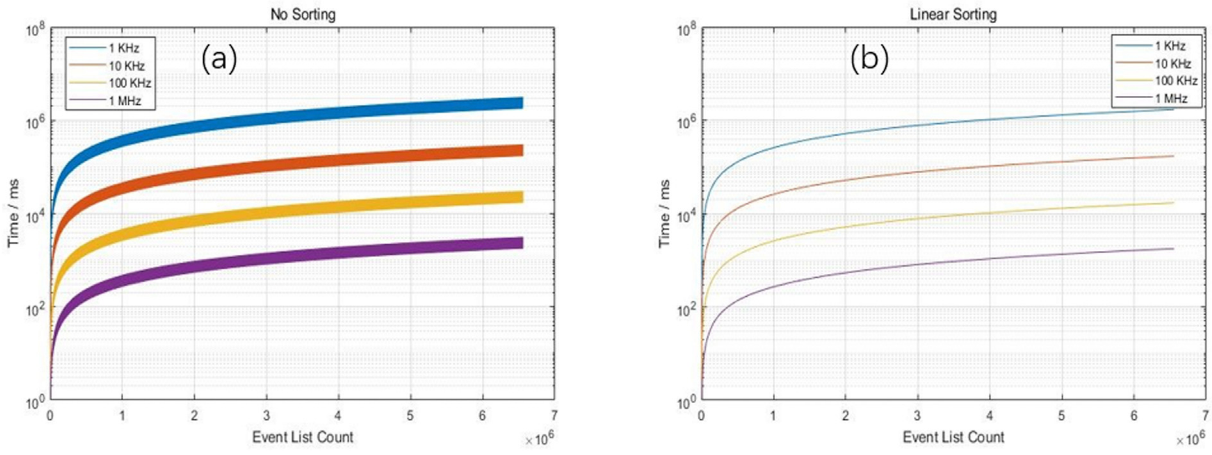

Fig. 6. Relationship curves between event count and the time spent in no sorting (a) and sorting (b) DAQU test.

Table 1
The test results got under two modes (No Sorting and sorting).

Test Module	Test index	1 kHz	10 kHz	100 kHz	1 MHz
No sorting	Total event Count in 4 channel DAQU	6553590	6553590	6553590	6553590
	Count for error list event	2773364	3052779	3067903	3069900
	The error rate	42.32%	46.59%	46.82%	46.85%
Sorting	Total event Count in 4 channel DAQU	6553590	6553590	6553590	6553590
	Count for error list event	0	0	0	0
	The error rate	0.0%	0.0%	0.0%	0.0%

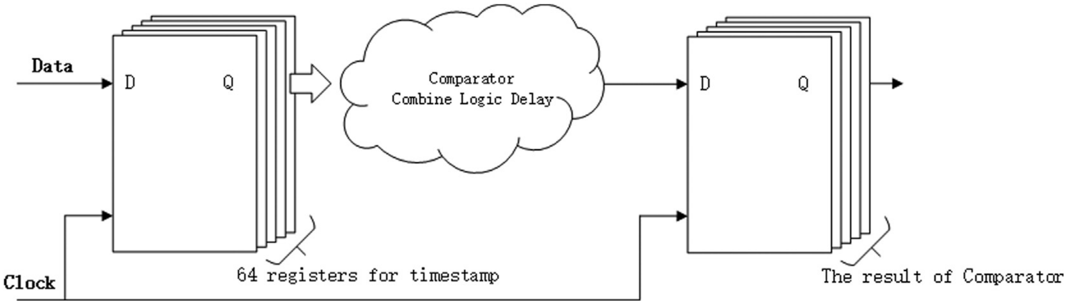

Fig. 7. The illustration diagram for the test of the combine logic delay of comparator.

Table 2
Evaluation of the valid combine logic delay of comparator.

Delay Period: N	Error Event Rate	Access
1 * period	2.56%	No
2 * period	0.03%	No
3 * period	0.0%	Acceptable

4.4. Application test with radioactive source

After performing the test with a pulse generator, the linear sorting algorithm is also verified with the ^{22}Na radioactive source, its count rate is 73.28 kcps. And also a comparison has been done with the measure results gained without sorting algorithm used in system. As shown in Fig. 8, the left bar is the analysis result without sorting algorithm in DAQU, the test results show that the events with wrong ordering is up to 58.38 k counts in total volume of event counts of 130 k counts, the error event rate reaches 44.9%. The right bar indicates the result from the DAQU where the sorting algorithm is used, there is not the event with wrong ordering found, and the error event rate is 0.0%.

After that, it is evaluated that the effect of selecting the coincidence events from the event data that is sorted first with the linear sorting algorithm. A measurement has been implemented, the coincidence counts of 10.63 kcps is gained during the time window of 60 ns, the proportion of coincidence events is 14.5%. This result is similar to the one got from the coincidence circuit with AND-Gate architecture. The time window in the algorithm is flexible and adjustable. Thus, the linear sorting algorithm based on the information of timestamp can be utilized effectively for the preliminary judgment of coincidence events online.

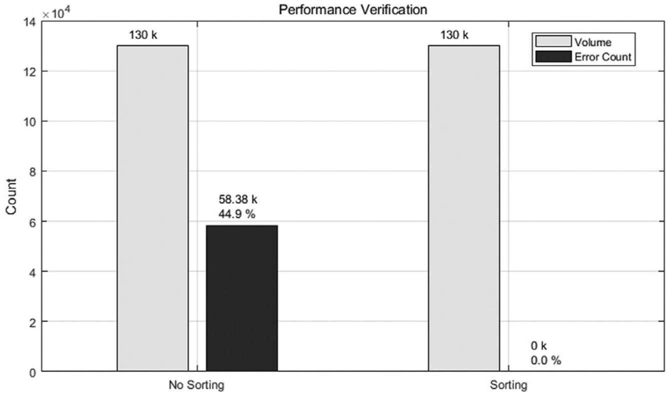

Fig. 8. Application test results with radioactive source.

5. Conclusion

A real time sorting algorithm based on timestamp, described in this paper, has been developed successfully. It is also tested and verified together with the DAQU of the In-Beam PET under the conditions of digital signal source and ^{22}Na radioactive source. As linear sorting modules are adopted in the algorithm architecture, a number of tasks are implemented in parallel without the need of complex analog and control units, and also several tasks can be completed correctly in one process period. The algorithm exhibits the excellent sorting performances, it can be operated at a high count rate without wrong sorted events appeared. This algorithm has been applied in DAQU of the In-Beam PET, an ideal, real time and low-cost approach for discriminating the coincidence events online has been provided. This linear sorting algorithm has the features of good portability characteristics and less resource utilized. It could be easy applied not only in the In-Beam PET, but also in small animal PET and other data process circuit.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 11775285 and 11975293), the Science and Technology Research Foundation of Gansu Province (NO.

20JR10RA066), the Youth Innovation Promotion Association, CAS (No. 2019408), the Key Deployment Project of Chinese Academy of Sciences (Grant No. KFZD-SW-222) and Fujian Institute of Innovation, CAS(FJCXY18040201).

References

[1] J.W. Yan, H.B. Yang, L.Y. Ke, et al., Prototype design of read-out electronics for in-beam TOF-PET of heavy-ion cancer therapy device, Nucl. Instrum. Methods Phys. Res., Sect. A 959 (2020) 163492, https://doi.org/10.1016/j.nima.2020.163492.

[2] G.Q. Xiao, H. Zhang, Q. Li, et al., Progresses of heavy-ion cancer therapy, Chin. Phys. C 32 (2008) 8–12. http://hepnp.ihep.ac.cn/en/article/id/ec79cbbd-ec39-4816-8b5f-51351da2bd75.

[3] J.D. Chen, H.S. Xu, Z.G. Hu, et al., A TOF-PET prototype with position sensitive PMT readout, Chin. Phys. C 35 (2011) 61–66, https://doi.org/10.1088/1674-1137/35/1/013.

[4] X.A. Chen, J.H. Zhao, J.H. Song, et al., Dual time point FDG PET imaging in evaluating pulmonary nodules with low FDG avidity, Nucl. Sci. Technol. 21 (2010) 306–311, https://doi.org/10.13538/j.1001-8042/nst.21.306-311.

[5] S.Q. Weng, V. Bettinadi, AN automatic segmentation method for fast imaging IN PET, Nucl. Sci. Technol. 2 (1993) 114–119.

[6] W.W. Moses, Time of flight in PET revisited, IEEE Trans. Nucl. Sci. 50 (2003) 1325–1330, https://doi.org/10.1109/TNS.2003.817319.

[7] S. Heecht, W. Enghardt, F. Fiedler, et al., In-beam pet at clinical proton beams with pile-up rejection, Z. Med. Phys. 27 (2017) 202–217, https://doi.org/10.1016/j.zemedi.2016.07.003.

[8] E. Fiorina, V. Ferrero, F. Pennazio, et al., Monte Carlo simulation tool for online treatment monitoring in hadrontherapy with in-beam PET, Phys. Med. 51 (2018) 71–80, https://doi.org/10.1016/j.ejmp.2018.05.002.

[9] C.R. Park, Y. Lee, Comparison of PET image quality using simultaneous PET/MR by attenuation correction with various MR pulse sequences, Nucl. Eng. Technol. 51 (2019) 1610–1615, https://doi.org/10.1016/j.net.2019.04.008.

[10] A.E. Perkins, M. Werner, A. Kuhn, et al., Time of flight coincidence timing calibration techniques using radioactive sources, IEEE Nucl. Sci. Symp. Conf. Rec. 5 (2005) 2488–2491, https://doi.org/10.1109/NSSMIC.2005.1596845.

[11] B. Goldschmidt, D. Schug, C.W. Lerche, et al., Software-based real-time acquisition and processing of PET detec tor raw data, IEEE Trans. Biomed. Eng. 63 (2015) 316–327, https://doi.org/10.1109/TBME.2015.2456640.

[12] C. Wang, H.D. Li, R.A. Ramirez, et al., A real time coincidence system for high count-rate TOF or non-TOF PET cameras using hybrid method combining AND-logic and time-mark technology, IEEE Trans. Nucl. Sci. 57 (2010) 708–714, https://doi.org/10.1109/TNS.2009.2039228.

[13] M.A. Tetrault, J.F. Oliver, M. Bergeron, et al., Real time coincidence detection engine for high count rate timestamp based PET, IEEE Trans. Nucl. Sci. 57 (2010) 117–124, https://doi.org/10.1109/TNS.2009.2038055.

[14] X.Y. Niu, J.W. Yan, X.H. Wang, et al., Evaluation of energy correction algorithm for signals of PET in heavy-ion cancer therapy device, Nucl. Eng. Technol. 52 (2020) 101–108, https://doi.org/10.1016/j.net.2019.07.008.

[15] J.H. Wang, S.B. Liu, Q. Shen, et al., A fully fledged TDC implemented in field-programmable gate arrays, IEEE Trans. Nucl. Sci. 57 (2010) 446–450, https://doi.org/10.1109/TNS.2009.2037958.

[16] J. Song, Q. An, S.B. Liu, A high-resolution time-to-digital converter implemented in field-programmable-gate-arrays, IEEE Trans. Nucl. Sci. 53 (2006) 236–241, https://doi.org/10.1109/TNS.2006.869820.

[17] K.B. Kim, H.T. Leem, Y.H. Chung, et al., Feasibility study of multiplexing method using digital signal encoding technique, Nucl. Eng. Technol. 52 (2020) 2339–2345, https://doi.org/10.1016/j.net.2020.03.027.

[18] J.W. Yan, X.Y. Niu, L.Y. Ke, et al., Development of new front-end readout electronics for γ-γ coincidence measurement system, Atomic Energy Sci. Technol. 54 (2020) 1500–1507.

[19] C.J. Borkowski, M.K. Kopp, Some applications and properties of one- and two-dimensional position-sensitive proportional counters, IEEE Trans. Nucl. Sci. 17 (1970) 340–349, https://doi.org/10.1109/TNS.1970.4325710.

[20] G.Z. Shi, R.F. Chen, K. Chen, et al., Development of a position sensitive CsI(Tl) crystal array, Nucl. Eng. Technol. 52 (2019) 835–840, https://doi.org/10.1016/j.net.2019.09.003.

[21] J. Ortiz, D. Andrews, A configurable high-throughput linear sorter system, IEEE INT SYMP (2010) 1–8, https://doi.org/10.1109/IPDPSW.2010.5470730.

[22] R. Perez-Andrade, R. Cumplido, C. Feregrino-Uribe, et al., A versatile linear insertion sorter based on an FIFO scheme, Microelectron. J. 40 (2009) 1705–1713, https://doi.org/10.1016/j.mejo.2009.08.006.