International Journal of Engineering & Technology, 7 (4.40) (2018) 90-93

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Hardware Platform Design Analysis of K-Means Clustering Algorithm Implementation

Ferry Wahyu Wibowo[1]*, Sudarmawan[2], Mulia Sulistiyono[3]

[1]*Informatics Department, Universitas Amikom Yogyakarta*
[2]*Informatics Department, Universitas Amikom Yogyakarta*
[3]*Informatics Department, Universitas Amikom Yogyakarta*
**Corresponding author E-mail: ferry.w@amikom.ac.id*

Abstract

The K-Means clustering algorithm is an unsupervised data mining technique and it has also been used widely to solve the problem in real life. This paper addresses a design analysis of the algorithm of K-Means that is implemented in the field programmable gate arrays (FPGAs). These devices are integrated circuits and have a hardware platform and applied in many implementations. The K-Means clustering algorithm has a simple method that is choosing objects from data to become the center point or centroid and then assign each object to the cluster which is nearest and update the cluster means. The approach of the software can also be implemented in the hardware, but both of them have a different method in programming.

Keywords: *Centroid; Clustering, FPGA; Hardware, K-Means.*

1. Introduction

The clustering is a classification of some objects into different groups or in other words, it could be said that the partitioning of some data sets into subsets. So ideally the data in each subset have the same common characteristics. The clustering has applied in unsupervised learning application, data mining, object tracking, pattern recognition, image quantization, etc. There are two types of clustering namely partitioning and hierarchical clustering. The partitioning clustering is data object sets those separate each other in a universal set so the data object exists in one subset while the hierarchical clustering is a group of nested clusters structured as a hierarchical tree. The hierarchical clustering consists of a single link and complete link, while the partitioning clustering contains square error, graph-theoretic, mixture resolving, and mode seeking. The mixture resolving is an expectation maximization, whereas the square error is a K-Means [1].

The K-Means clustering algorithm is an unsupervised data mining technique and has been used to solve the problem in real life. This paper presents a design analysis of the algorithm of K-Means that is implemented in the field programmable gate arrays (FPGAs). The FPGAs have a hardware platform and applied in many implementations. It can also be reconfigured by the hardware-platform design entries using hardware description language (HDL) or schematic [2]. The aim of this clustering is grouping the objects within a similar to one another and different from the other object groups. So the K-Means clustering algorithm has a centroid or center point of the cluster and each point is assigned to the cluster with the closest centroid. The number of clusters, κ, must be defined and specified.

In the aspect of the FPGAs' implementation always has a thread-off using this algorithm. One of the examples is the implementation of the FPGAs applied in different heterogeneous devices. It showed that the combination between both symmetric multiprocessing (SMP) and graphical processing unit (GPU) is powerful than a combination between SMP and FPGA. But the GPU can't improve the best SMP, while the FPGA can manage the data movements and decoupling accelerator computation and communication [3]. The combination of software and hardware could accelerate the process in a heterogeneous system for big data [4]. The K-means implementation using FPGAs in the heterogeneous computer cluster has shown an efficient performance and autonomous data movement between multiple FPGAs [5].

2. Algorithm of K-Means

The basic of the K-Means algorithm is grouping objects based on a minimum distance of centroids. The initial centroids are usually chosen randomly so the clusters could have resulted in many variations and different centroid when it has run. The centroid is commonly a mean of the point in the cluster. The closest values or features in point 2 can be measured using some approaches such as a correlation, Euclidean distance, cosine similarity, etc. The points will converge using those similarity measures in this algorithm. The equation of the Euclidean distance, ε, could be written as pointed in (1).

$$\varepsilon = \sqrt{\sum_{i=1} |\alpha_i - \beta_i|^2} \tag{1}$$

where i is the iteration and α, β are point values. To update the centroid point, ς, can use the equation (2) in calculating the centroid of the n-dimensional.

$$\varsigma = \left(\frac{\sum_{i=1} \alpha_i}{\kappa}, \frac{\sum_{i=1} \beta_i}{\kappa}, ..n \right) \quad (2)$$

where κ is the number of clusters. For evaluating the K-Means clusters, it could employ the sum of squared error (SSE), ρ, as pointed in (3). For each point, it could be calculated the error using this equation. The error, in this case, means the distance to the nearest cluster.

$$\rho = \sum_{i=1}^{\kappa} \sum_{\alpha \in \gamma} \delta^2 (\psi_i, \alpha) \quad (3)$$

where γ is a cluster, ψ_i is corresponding to the mean/center point of the cluster and δ is the distance. To reduce the SSE, it can increase the numbers of the clusters. K-Means algorithm has a limitation when the clusters have differences on densities, sizes, and shapes.

There are two main steps in applying the K-Means clustering algorithm which are assignment and update steps [6]. The pseudo-code of this algorithm can be written shortly as below,

1. **Input:** κ, α
2. **Output:** a group of κ clusters
3. **Method:** choose κ objects from α as the centroid
4. **Repeat:**
    a. assign each object to the cluster which is nearest
    b. update the cluster means
5. **Until** no change;

The design of the K-Means can be depicted as figure 1.

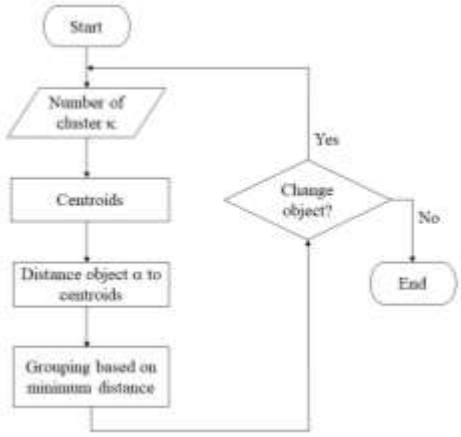

Fig. 1: Flowchart of the K-Means

3. Methodology for K-Means Algorithm Hardware Implementation

Many hardware engineers realize that the software development lifecycle can be applied to the FPGA development that is a hardware-platform [7]. The implementation of the FPGA-based is always mirroring to the software development but both of them have a different method [8]. Not in that case only, but hardware engineers have also realized many functions those could be handled by microprocessors (μP) are also capable of transferring to the FPGAs. The implementation of the K-Means clustering algorithm has been implemented to process many different datasets for a server problem. The result has provided about more 50x speed-up than using a software model [9]. The methodology for this implementation starts with finding the minimum distance of the 8 input data and thus applying the comparator blocks to make comparison each other with the centroids then the data will be output to the appropriate clusters. Every input of data, it will process like that. The data results are classified into 2 outputs i.e. cluster1 and cluster2.

For large data sets that the FPGAs couldn't handle it in the internal storage memory. The global design of the K-means in this paper used FPGA Xilinx Artix7. The main circuitry system designed using VHDL code consists of comparators. These components are designed separately and invoked using component VHDL code. So it is more simple to detect the error. A clock that has been applied in this design is using 50 MHz or 20 ns as this FPGA Xilinx development board has. This design is shown in Figure 2.

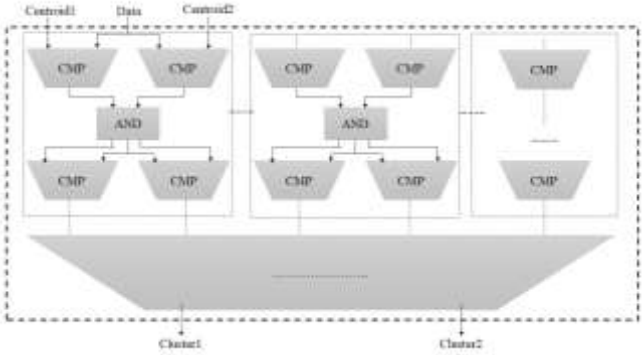

Fig. 2: The design of the K-Means

Figure 2 shows the design of the modular, the components work as the functions. The work principals of this system are initialized with entering 3 inputs i.e. 2 centroids and 1 data with 8-bits. Each input is 1 byte and it is using a random signal input. After choosing 8-bits of input signals, each comparator will choose signals as the minimal value that represents as the closest distance. This comparison is always done to group it to the closest value with each centroid. The principle of this design can be shown in figure 3.

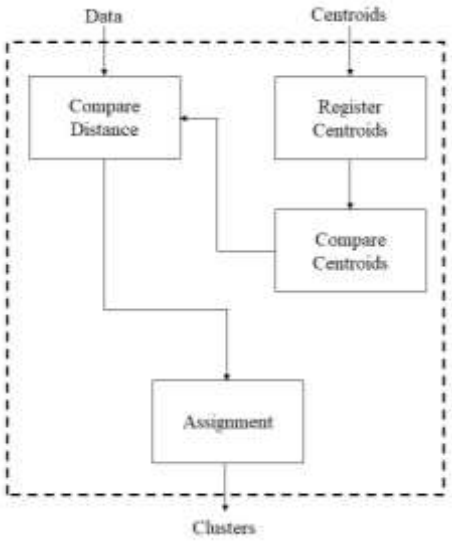

Fig. 3: The diagram of the K-Means

4. Result and Discussion

The design of the K-Means that has been made consists of inputs and outputs. The input ports consist of two centroids, 1 input data, while the output ports consist of 2 clusters. The ports of the K-Means design are depicted as figure 4.

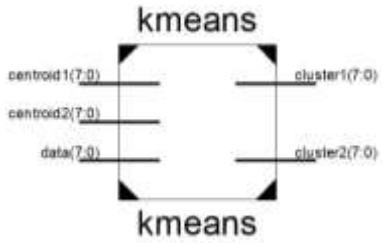

Fig. 4: The design of the K-Means

In the design using very high speed integrated circuit hardware description language (VHDL), the warning indicator should be noticed. As in this design, the warning notification has emerged as a latch. The latch in the design could make an unstable output signal. Some error in the placement of the signal in the process is also warned when it uses the sensitivity list. The latch error could be handled by placing an initial value "00000000" on the output ports. The register transfer logic (RTL) for this design is shown in figure 5.

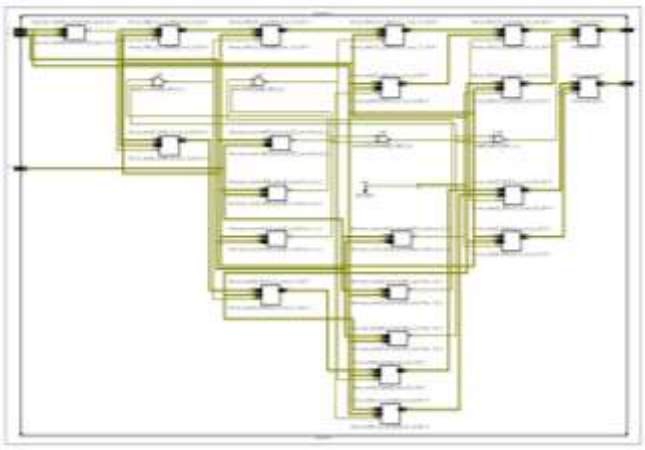

Fig. 5: RTL of the design of the K-Means

The ports of centroid1 and centroid2 are used to make the data will be clustered. In this design, the cluster placement uses two types i.e. cluster1 and cluster2. This concept is using a comparison between two signals those are handled by the centroids. While the waveform of the design of the K-Means is shown in figure 6.

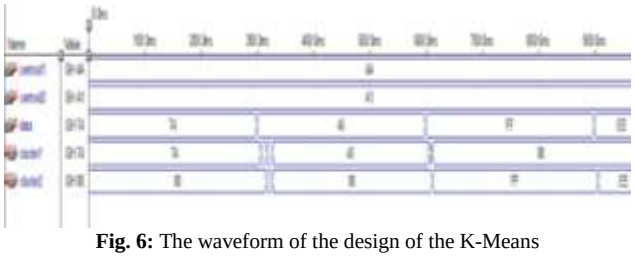

Fig. 6: The waveform of the design of the K-Means

Figure 6 uses clock 20ns and the time needed before the data signal execution is 5ns. In the sensitivity list, the initial cluster should be valued by "00000000". The centroid and data in the design must be compared then the data will be chosen to be clustered in one of the two clusters. The model is very simple and it needs more skills in VHDL programming.

In macro statistics, the design is only consuming comparator and multiplexer. There are two types of comparators which are comparator equal and comparator greater. While for the 8-bits design of the K-Means it is only used one of the 2-to-1 multiplexers. There is a delay about 4.945ns when it is used at the source of centroid2 to the destination of cluster1. The total memory usage of making the 8-bit design of K-Means is taking a size of 344672 kilobytes. While the floorplan of this design is shown in figure 7.

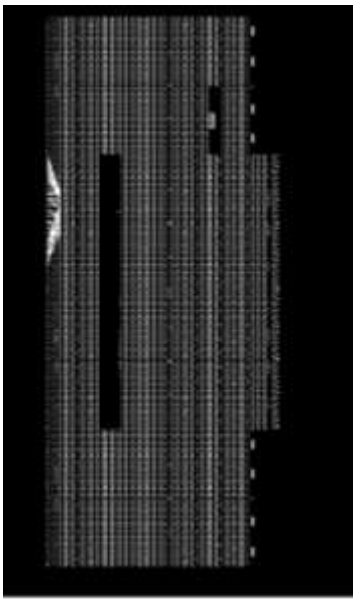

Fig. 7: The floor plan of the design of the K-Means

A consumed time that is needed to run this design is about 13 seconds of real-time and 12.78 seconds of CPU time. The time used in running the logic and route is about 4.945ns which contains 0.777ns logic (15.7%) and 4.168ns route (84.3%). The consumption of components used in this design is shown in table 1.

Table 1: Device utilization summary

No.	Component Consumptions	
	Component Names	Numbers
1	Slice LUTs	48 out of 63400
2	LUT Flip Flop pairs	48 out of 48
3	Bonded IOBs	40

5. Conclusion

In making the design of the K-Means clustering algorithm in the hardware platform that is using K=2 and 8-bits inputs, whereas the outputs are clusters in 8-bits, there are no asynchronous control signals found in this design and the maximum combinational path delay is about 4.945ns. The design synthesized is only using 7 comparators and 14 multiplexers. Time consumed by this design is about 4.945ns. Future work for this design can be combined with the design of the hardware-based database for its clustering and also can be developed in making a dynamic K-means.

Acknowledgment

This research has been funded by the ministry of technology research and higher education (RISTEKDIKTI) grant no. 019/HB-LIT/II/2018.

References

[1] SC Candrakala, TSG Basha & K Anjani (2014), Implementation of medical image segmentation using K-Means clustering technique on FPGA. *International Journal of Science, Engineering and Technology Research*, Vol. 3, Issue 10, 2861-2867

[2] Wibowo FW (2011), Interoperability of Reconfiguring System on FPGA Using a Design Entry of Hardware Description Language, *Proceedings of the 2011 Computation and Communication Technologies: 3rd International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT 2011 - Computer Science Series 1*, pp. 79-83

[3] Ying HX, Miquel V, Beñat A, Javier D, Carlos A, Daniel JG, Xavier M & Filippo M (2017), Implementation of the K-means algorithm on heterogeneous devices: a use case based on an industrial dataset, *Advances in Parallel Computing*, pp. 642-651.

[4] D. Lee, Alric A, Dustin R & Ryan K (2017), A streaming clustering approach using a heterogeneous system for big data analysis, *Proceedings of the 2017 IEEE/ACM International Conference on Computer-Aided Design* (ICCAD), pp. 699-706
[5] Yuk MC & Hayden KHS (2014), Map-reduce processing of K-means algorithm with FPGA-accelerated computer cluster, *Proceedings of the 2014 IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors*, pp. 9-16.
[6] José C, FPGA implementation of a multi-processor for cluster analysis, unpublished
[7] Wibowo FW (2015), Implementation of Viterbi algorithm based-on field programmable gate array for wireless sensor network, *Advanced Science Letters*, Vol. 21, No. 11, pp. 3521-3525
[8] Awos K, Fayez G & Atef I (2016), Fast and area-efficient hardware implementation of the K-Means clustering algorithm, *WSEAS Transactions on Circuits and Systems*, Vol. 15, pp. 133-142
[9] Hanna MH, Khaled B, Huseyin S & AT Erdogan (2011), FPGA implementation of K-Means algorithm for bioinformatics application: an accelerated approach to clustering microarray data, *Proceedings of the 2011 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*
[10] Wibowo FW, (2018), Implementation of FPGA in index data storage as database, *International Journal of Engineering & Technology*.