

## Evaluation of Computer Architecture and Organization Using the SPSS Method

Nitesh Kumar Ramancha\*

\*Sr. SAP Consultant, SAP, TX, USA

### ARTICLE INFO

#### Article history:

Received: 20250208

Received in revised form: 20250213

Accepted: 20250215

Available online: 20250219

#### Keywords:

Arithmetic logic unit;

Data memory;

Instruction memory;

Hardwired control unit and Micro  
programmed control unit.

### ABSTRACT

The process of creating a computer's internal structure and layout, as well as the electronic components' interactions with one another, is referred to as computer architecture design. It entails making choices about the architectural of the entire system as well as the CPU, memory hierarchy, input/output (I/O) systems, and memory hierarchy. (ISA) Instruction Set Architecture The interface that exists between software and computer hardware is described by ISA. It contains the set of instructions the CPU is capable of processing, the addressing modes, and the register structure. The ISA decision has an impact on CPU architecture and programmed compatibility. CPU Design: The brains of a computer system, the CPU. A control unit, an arithmetic logic unit (ALU), registers, and data routes are all included in its architecture. Pipelining, superscalar implementation, out-of-order execution, and branched prediction algorithms are design options that can boost performance. Research on computer design is crucial for developing computer technologies and scientific knowledge.

Here are several main justifications for the significance of computer architecture design research: Performance Enhancement: Research in computer architecture design focuses on enhancing the functionality of computer systems. Researchers can develop processors, memory layers, and I/O systems that are more efficient by investigating novel architectural methodologies and designs. This results in quicker calculations, quicker responses, and better system performance all around. Energy & Efficiency in Technology: With the demand for energy-efficient computer systems growing, computer design research is crucial in creating designs that use the least amount of electricity possible. Researchers are able to contribute to more sustainability computing solutions and lessen the impact of technology by enhancing design choices and introducing low-power strategies. Scalability and Parallelism: Scalable and parallel computing systems are crucial as computing demands continue to increase.

Techniques including multiple cores, parallel programming methods, and distributed systems for computation are explored in this field of study. Researchers permit high-performance computing and effectively carry out complicated tasks by inventing architectures that make optimum use of parallelism. SPSS statistics is multivariate analytics, business intelligence, and criminal investigation data management, advanced analytics, developed by IBM for a statistical software package. A long time, spa inc. Was created by, IBM purchased it in 2009. The most contemporary versions are marketed under the designation IBM SPSS statistics. Arithmetic logic unit, Data memory, Instruction memory, Hardwired control unit, Micro programmed control unit The Cronbach's Alpha Reliability result. The overall Cronbach's Alpha value for the model is .860 which indicates 86% reliability. From the literature review, the above 50% Cronbach's Alpha value model can be considered for analysis.

© Nitesh Kumar Ramancha.

\*Corresponding author. e-mail: [nkr112024@gmail.com](mailto:nkr112024@gmail.com)

## Introduction

The capabilities of the technology underneath versus the requirements of the applications are separated by computer architecture. System designers must constantly innovate to develop technologies that can achieve the necessary performance as well as price effectiveness when application needs change and technology overcome various limits. To offer end-to-end improvement in performance at historical levels in the face of technology limitations is our challenge as system designers. By concentrating on energy optimization at all levels, we can overcome this difficulty [1]. The implementation of chip-level multiprocessors, the usage of accelerates and offload engines, the broad adoption of scale-out systems as a whole and system-level power optimization are important levers. Accelerators in computer architecture. Multiple cores must be interconnected to the CPU using a method that is separate from the network architecture. Thus, a network around chip (NoC) is suggested. Picture. In 2(b), the processor cores are linked electrically by means of on-chip routers [2] and each one has a tiny amount computational memory on-hand without any contention [2].

However, as additional cores are required to solve the optimal service-demand matching, the NoC's confined data interchange rate becomes the bottleneck. Computer architecture therefore emerges as the cognitive systems' accelerator in the optical NoC (ONoC), which chooses the best service-demand fitting for the sharing economy in place of copper wires in place of fast the waveguides to build the optical connectivity of cores [3]. The controller of the intelligent system enables the worldwide service-demand comparison to be divided into smaller tasks for different subgroups. The tasks from subgroups are then assigned using task-core mapping to the appropriate cores of potential accelerator designs, where they can be merged to build a task graph. The concepts of workload graph, accelerator architecture, and project-core mapping are essential for computer architectures that can efficiently deal with such unpredictable behavior [4].

The final critical characteristic of many undertakings is their capacity to scale out. A scale-out architecture consists of a number of inexpensive, networked, modular computers working together to provide users with applications, system resources, and data. Systems that are considerably parallel, such as clusters and rack-mounted blade systems are examples of expansion platforms [5]. Traditional symmetric multiprocessors (SMP) systems, on the opposing hand, are scale-up platforms. Fifth-generation architectures of computers are theoretically viable. Which architectural concepts and characteristics from the numerous research programmers will be included in this new generation of computers, then, is the question. There are two main study topics that make up the work on the reduction and analysis of data circulation (DENN79b, Gosw79a) for the data-driven and demand-driven architecture [6]. The arrangement of computation, stored programmers, and machine resources makes these areas distinct from one another.

Although research teams in each field have a common set of fundamental ideas, in order to get around obstacles, each team has frequently included ideas from other fields, including conventional control flow structures. The purpose of this essay is to define the ideas and connections that exist both inside and across various academic fields. We begin by outlining straightforward operational structures for reduction, both control and data flow [7]. Within and within the three groups, how calculation, stored initiatives, and computer resources are organized is then classified and examined. In terms of these classifications, an overview of many innovative computer architectures currently under development is provided. Due to its continued emphasis on desktop and server applications, computer architecture development continues to have a bias towards the past. According to our predictions, the next ten years will see a major advancement in personal portable computing [8]. The fundamental personal computer and communication devices in this paradigm will be mobile, battery-operated, and capable of multi-media functions including speech recognition. These gadgets could change the focus of computer design and present a new set of demands on microprocessors.

The physical, tangible computer in relation to computer architecture. Because of this, there is still a great deal of uncertainty in the computer design a commonality even in the presence of design, description, and review tools: the idea that a "architecture" in and of itself is a nebulous concept whose reality is in question even in the absence of a physical implementation [9]. The "plausibility paradox" in computer design, in our opinion, is a significant issue. It has major consequences for the kinds of claims that can or should be made about a design proposal from a practical perspective, as well as how much confidence can be put into such statements prior to physical execution. From a more theoretical perspective, we think that grasping the plausibility problem could help with the effort to turn computer design from the rather esoteric craft that it now is into a scientific field [10]. Dasgupta has written extensively on this subject, but it is nevertheless important to briefly explain how our findings relates to this ideal.

Therefore, the computer architect is a designer of a certain kind of data handling system: those that are directly realized through a fusion of hardware and firmware. Additionally, it indicates that the architect is interested in two levels of description, for which Dasgupta proposed the words Ex structure and end building [11]. When we refer to the hardware/firmware system's logical structure, behavior, and capabilities as seen by the assembly computer programmer or compilers writer, we are referring to the "external architecture" (also known as the "outer architecture"). The "internal the building" of a computer, in contrast, illustrates the capacities and performance traits of its main functioning organs, the information flow patterns between the organs, along with the logic and methods used to control how data is processed and flow [12]. The machine's organization and behavior as they must be known by the micro programmed

the microcode compiler are a specific instance of the end architecture.

The structure is the standard term used to describe this. From a methodological standpoint, we suggest that limitations might be seen as design facts, design objectives, assumptions, and design styles in terms of computer architectures. The interpretation of a constraint can change based on the context in which it appears, as will be seen in Example 2. Actual aspects of the artefact being developed are defined by design facts [13]. During the design phase, these features could undergo changes. On the other hand, design objectives specify desired qualities for both the design process and the final artefact. Keep in mind that design facts can evolve into objectives that lower-level design facts (and assumptions) may be able to satisfy. Making assumptions can help narrow the range of potential solutions by making comments about the environment or the building process itself. During the design phase, the prerequisites mentioned in these assertions are taken for granted [14]. When designing, the designer doesn't take any steps to prevent or identify the consequences of breaking assumptions. Methodologies gap in computer architecture research.

The separate Functions level (FL), cyclic language (CL), and register transferring level (RTL) frequently used languages, structures for design, and tools modeling are the manifestation of this divide. We think the lack of a vertically integrated framework for computer architecture study techniques highlights a fundamental requirement for quick execution of hardware and design-space explorations [15]. In a perfect world, such a framework would allow multi-level simulations that include models at various abstraction levels, use a single specification language for FL, CL, and RTL modeling, and offer a way to create automated tool flows for the extraction of reliable area, energy, and temporal data. One of the important knowledge domains of computer engineering involves the architecture of computers and organization. Therefore, numerous goals must be accomplished by courses in this field. The main goal of these courses is to give students an overview of topics related to computer technology and organization as well as an understanding of how Workings of a typical computing systems [16].

Additionally, they should highlight all of the the significant problems with computer design and organization that confront working engineers. Additionally, it is anticipated that the courses will reinforce concepts related to several computer science disciplines, such as programming languages, operating systems, and database management systems, etc. Students must also be introduced to the many tools they will need to use in order to do research and development when they graduate from school. As a result, the majority of courses in computer architecture and organization contain hands-on laboratory work [17]. By watching and examining the traits and behaviors of actual systems, this study enables students to validate the theory they have learned from lecture lectures. Designing experiments as well as implementing, testing, and documenting hardware and

software are all included in the practical work. The goal of the multifaceted multimedia system on "Computer architecture, organization, and design" gaining a comprehensive understanding of fundamental ideas that are sometimes overlooked while studying specific computing tasks [18].

Students are therefore able to visualise and connect fundamental ideas that are typically taught in fragments in conventional classroom lectures thanks to an integrated knowledge. The approach makes it easier to understand by presenting concepts with unified images or animation that combine talks that are typically found in many textbook chapters. Students are given the opportunity to reply to information displayed on the screen, allowing them to further their understanding of the subject. This kind of engagement, which is unrealistic in books, allows students to go at a pace fit for their specific learning levels and leads them through the learning process in a particular fashion, offering the level of personalization that is most suitable for each student.

A fog computing architecture is suggested [19]. Anticipating the requirement and frequency of transmission between sensors reduces superfluous transmission at a lower level. The sections below make up this essay. The second section goes over the Internet of Things, the computing environment that was suggested in section two children, attribute-based encryption for computation in fog and secure storage of data, as well as the sensor system that underlies device communication. In Section 3, a hierarchical management approach that anticipated the use of energy along with sensor usage frequency is suggested as a description of the proposed system to optimize the overall structure and communication between devices [20].

### **Materials and Method**

**Arithmetic logic unit:** An algebraic-logic unit, a part of the central processing unit (CPU), performs both mathematical and logical calculations on the operands of computer instruction words. In some processors, the ALU is divided into a logical unit (LU) and a numerical unit (AU), respectively. unit of arithmetic. An ALU is made up of two units, the one that performs arithmetic and the logic component unit, as we already discussed. All of the mathematical operations in a computer, like addition and subtraction, are handled by the arithmetic unit. The incremental function and other basic operations are also performed by it.

**Data memory:** Data needed for program me functionality is stored in data memory (RAM). It may or may not be divided into two portions, depending on the program me in use. The data memory is believed to have two parts for the DSP2 instruction set. A computer's primary storage, usually referred to as primary Data is held in a segment called storage or memory so that the CPU can access it quickly. Primary or main storage is commonly referred to as random access memory, also known as RAM, or memory.

**Instruction memory:** The instruction is taken from one memory location; instructions are loaded and stored; and data is read or written from a different memory location. As a result, it is practical to split the main memory into two separate portions, each holding data and instructions separately. The control unit retrieves any data needed by the instruction from memory and stores it in the data path. The data path is used for the operation. The operation's outcome could be recorded to memory.

**Hardwired control unit:** A technique for producing control signals using Finite State Machines (FSM) is known as hardwired control. It is created by physically joining parts like flip-flops, gates, and drums to create a continuous logic circuit. It is referred to be a stiff controller as a result. What is a connected control unit? An approach to generating control signals with finite states Machines (FSM) is known as hardwired control. It is created by physically connecting parts like flip-flops, gates, and drums to create a serial logic circuit.

**Micro programmed control unit:** A control unit that saves binary controls as words in storage is known as a micro programmed control unit. A controller generates instructions by producing a precise set of information at each system's clock pulse. A series of micro operations is necessary for an instruction. Control signals are used to carry out micro-operations. These signals of control are produced in this instance via micro-instructions. This indicates that an inventory of micro-instructions is needed for every command. A micro-program is a collection of micro-instructions.

**Method:** A popular piece of software called SPSS (Statistical Package for Social Sciences) is utilized for statistical

analysis across a range of disciplines, including the social sciences, business, health, and market research. With its extensive set of instruments for data management, data manipulation, and statistical analysis, IBM's SPSS is a valuable resource for academics and data analysts. One of SPSS's standout qualities is its intuitive user interface, which both inexperienced and seasoned users may utilise. Users of the software can enter data in a manner resembling a spreadsheet, with variables sorted in column and cases (observations) arranged in rows. Numerous data kinds are supported by SPSS, including textual, category, and numeric data. A large variety of statistical methods are available in SPSS for data exploration and analysis. It offers both descriptive statistics for data summarization and inferential statistics for hypothesis testing and forecasting.

Common statistical tests including all be carried out by users. SPSS enables sophisticated data integrations, transformations, and manipulations. SPSS has facilities for data visualization in addition to its main statistical skills. Users can design graphs, charts, and plots to show their data visually, facilitating the interpretation and sharing of findings. For use in reporting and presentation, SPSS also permits the construction of customized tables and reports. Due to its reliability, adaptability, and thorough documentation, SPSS is an extensively utilized and regarded tool within the discipline of data analysis. Its popularity is a result of its capacity for handling huge datasets, wide variety of statistical techniques, and user-friendly interface. With the help of SPSS, researchers and analysts may carry out sophisticated statistical analyses and get insightful conclusions from their data.

## Result and Discussion

**TABLE 1. Descriptive Statistics**

	N	Range	Minimum	Maximum	Mean		Std. Deviation	Variance
Arithmetic logic unit	231	4	1	5	3.68	.062	.948	.898
Data memory	231	4	1	5	3.82	.065	.983	.967
Instruction memory	231	4	1	5	3.63	.070	1.071	1.147
Hardwired control unit	231	4	1	5	3.58	.067	1.017	1.035
Microprogrammed control unit	231	4	1	5	3.80	.065	.989	.978
Valid N (listwise)	231							

Table 1 shows the descriptive statistics values Arithmetic Logic Unit is Showing the Number: 231 Range: 4, Minimum: 1, Maximum: 5, Mean: 3.68, Standard Deviation: 0.062, Variance: 0.898. Data Memory is Showing the Number: 231, Range: 4, Minimum: 1, Maximum: 5, Mean: 3.82, Standard Deviation: 0.065, Variance: 0.967. Instruction Memory is Showing the Number: 231, Range: 4, Minimum: 1, Maximum: 5, Mean: 3.63,

Standard Deviation: 0.070, Variance: 1.147. Hardwired Control Unit is Showing the Number: 231, Range: 4, Minimum: 1, Maximum: 5, Mean: 3.58, Standard Deviation: 0.067, Variance: 1.035. Microprogrammed Control Unit is Showing the Number: 231, Range: 4, Minimum: 1, Maximum: 5, Mean: 3.80, Standard Deviation: 0.065, Variance: 0.978. The "Valid N (listwise)" value of 231 indicates that there were no missing values in the dataset.



**TABLE 2.** Frequencies Statistics

		Arithmetic logic unit	Data memory	Instruction memory	Hardwired control unit	Microprogrammed control unit
N	Valid	231	231	231	231	231
	Missing	0	0	0	0	0
Mean		3.68	3.82	3.63	3.58	3.80
Std. Error of Mean		.062	.065	.070	.067	.065
Median		4.00	4.00	4.00	4.00	4.00
Mode		4	4	4	4	4
Std. Deviation		.948	.983	1.071	1.017	.989
Variance		.898	.967	1.147	1.035	.978
Skewness		-.669	-.680	-.616	-.394	-.925
Std. Error of Skewness		.160	.160	.160	.160	.160
Kurtosis		.682	.085	-.260	-.395	.798
Std. Error of Kurtosis		.319	.319	.319	.319	.319
Range		4	4	4	4	4
Minimum		1	1	1	1	1
Maximum		5	5	5	5	5
Sum		849	882	839	828	878
Percentiles	25	3.00	3.00	3.00	3.00	3.00
	50	4.00	4.00	4.00	4.00	4.00
	75	4.00	5.00	4.00	4.00	4.00

Table 2 Show the Frequency Statistics in Computer Architecture Design Arithmetic logic unit, Data memory, Instruction memory, Hardwired control unit and Micro programmed control unit curve values are given.

**TABLE 3.** Reliability Statistics



Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.531	.531	5

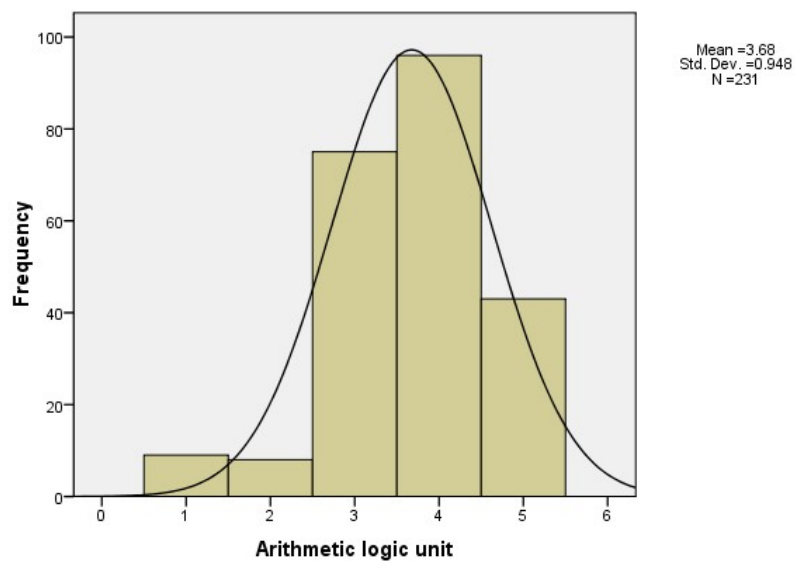
Table 3 shows The Cronbach's Alpha Reliability result. The overall Cronbach's Alpha value for the model is .860 which indicates 86% reliability. From the literature review, the above 50% Cronbach's Alpha value model can be considered for analysis.

**Table 4.** Reliability Statistic Individual

	Cronbach's Alpha if Item Deleted
Arithmetic logic unit	.418
Data memory	.517
Instruction memory	.409
Hardwired control unit	.541
Microprogrammed control unit	.476

Table 4 Shows the Cronbach's Alpha values provided for each parameter are as follows: Arithmetic logic unit: .418, Data memory: .517, Instruction memory: .409, Hardwired control unit: .541, Microprogrammed control unit: .476. Cronbach's Alpha is a measure of internal consistency reliability, and it quantifies how well the items in a scale or test are correlated with each other. Generally, higher Cronbach's Alpha values

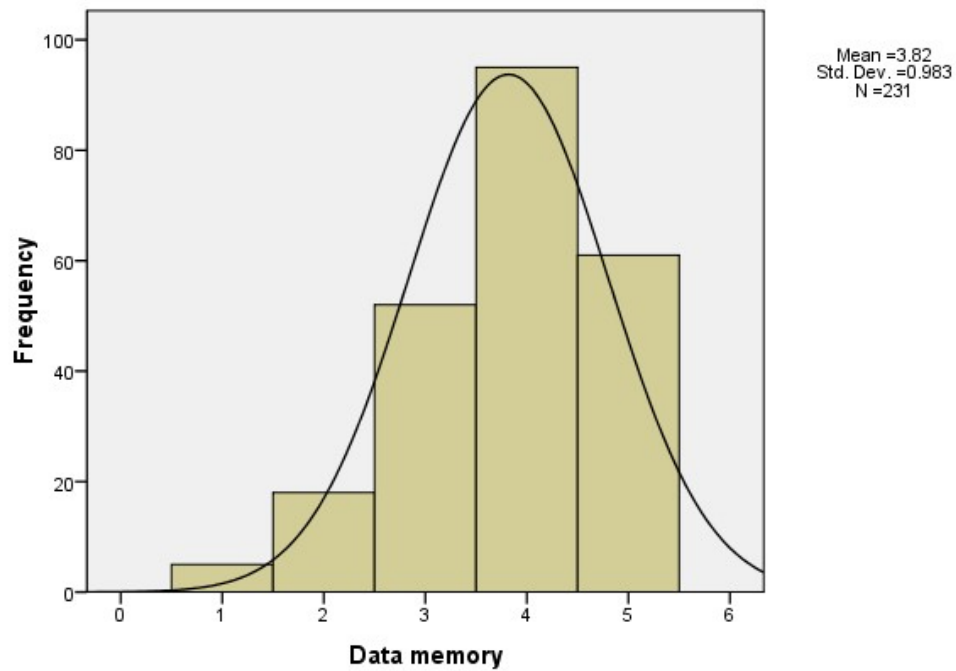
indicate greater internal consistency and reliability of the scale. In this case, all the parameters have Cronbach's Alpha values above .4, which suggests a moderate level of internal consistency. While these values may not be considered high, they still indicate a certain degree of reliability for each parameter.



**Figure 1.** Arithmetic logic unit

Figure 1 shows the histogram plot for Arithmetic logic unit. From the figure, it is clearly seen that the data are slightly right-skewed due to more respondents choosing 4 for Arithmetic logic

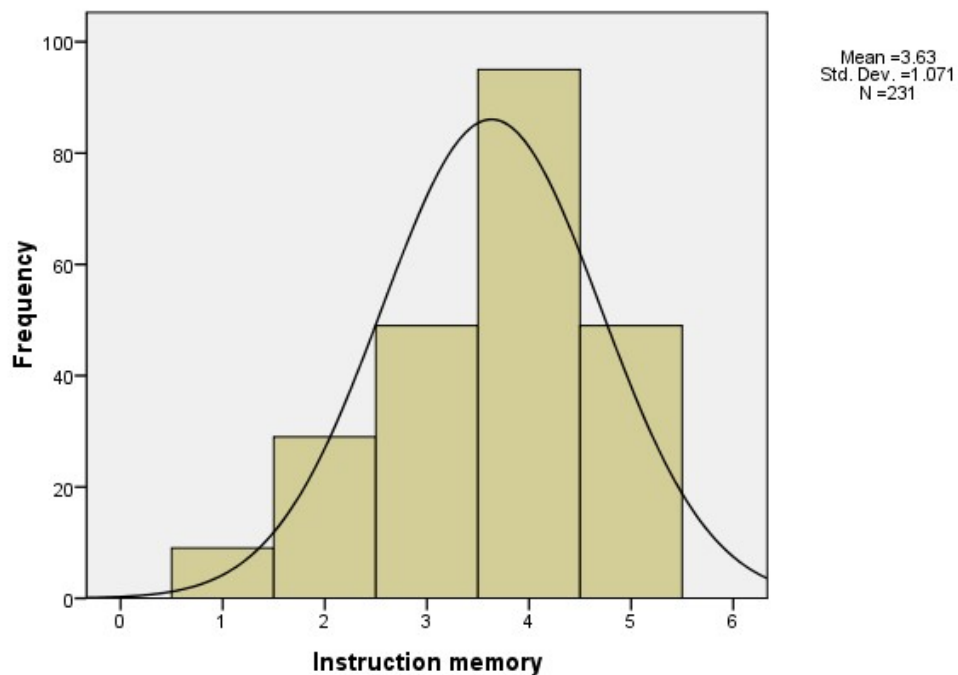
unit. Except for the value 2, all other values are under the normal curve. The model is significantly following normal distribution.



**Figure 2.**Data memory

Figure 2 shows the histogram plot for Recurrent neural networks from the figure it is clearly seen that the data are slightly Right skewed due to more respondent chosen 4 for Data

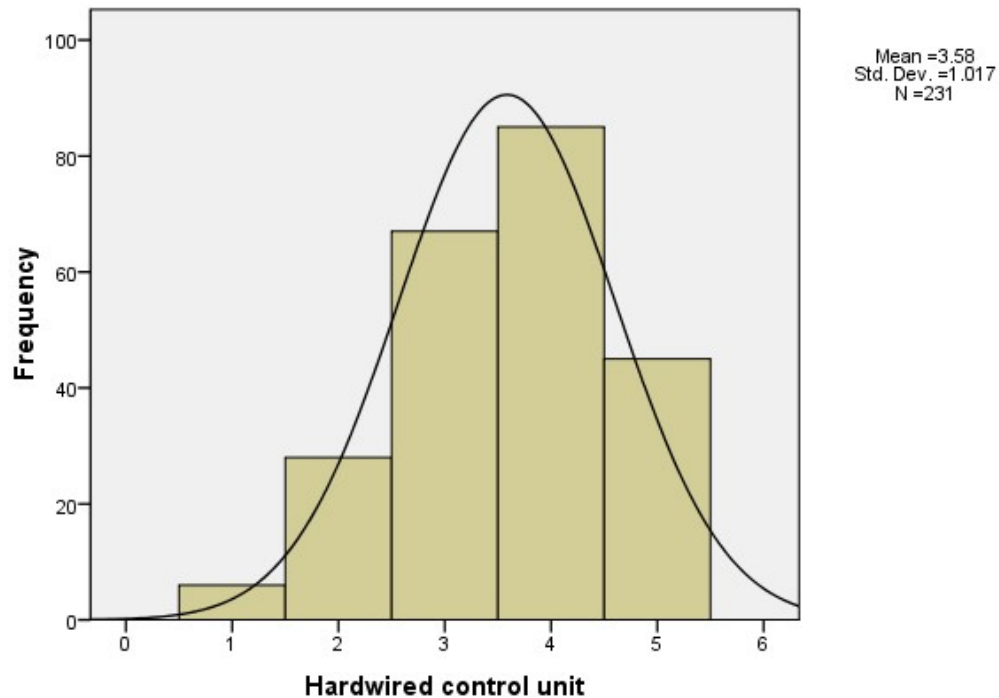
memory except the 2 value all other values are under the normal curve shows model is significantly following normal distribution.



**Figure 3.**Instruction memory

Figure 3 shows the histogram plot for Instruction memory from the figure it is clearly seen that the data are slightly Right skewed due to more respondent chosen 4 for Instruction

memory except the 2 value all other values are under the normal curve shows model is significantly following normal distribution.

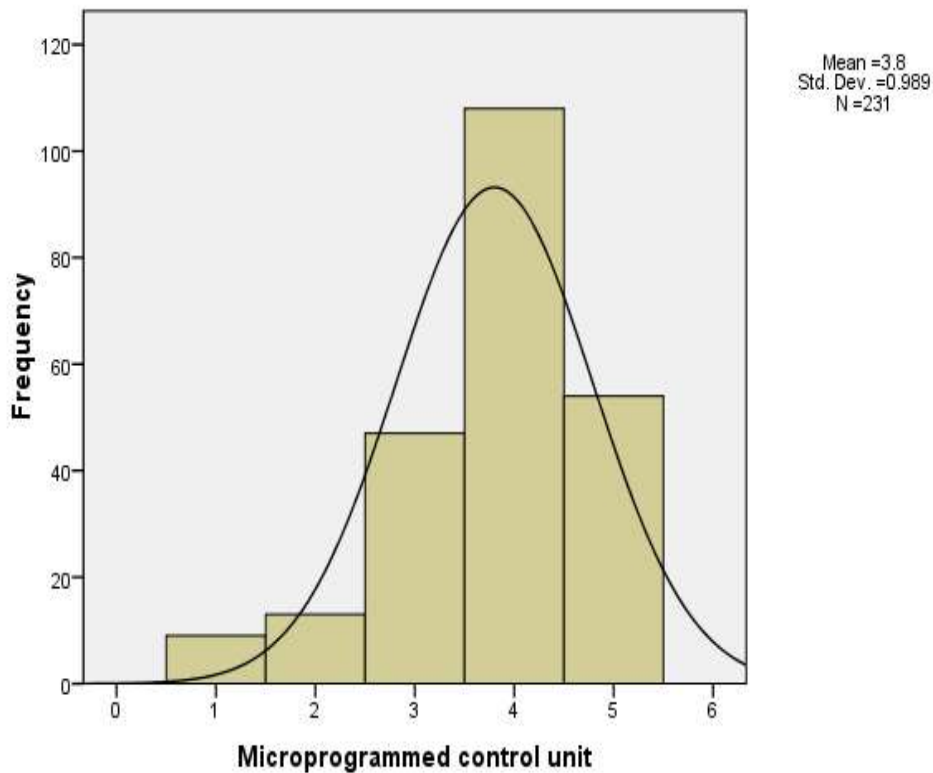


**Figure 4.**Hardwired control unit

Figure 4 shows the histogram plot for Hardwired control unit from the figure it is clearly seen that the data are slightly Right skewed due to more respondent chosen 4 for Hardwired

control unit except the 2 value all other values are under the normal curve shows model is significantly following normal distribution.





**Figure 5.**Microprogrammed control unit

Figure 5 shows the histogram plot for Micro programmed control unit from the figure it is clearly seen that the data are slightly Right skewed due to more respondent chosen 4 for Micro programmed control unit except the 2 value all other

values are under the normal curve shows model is significantly following normal distribution.



**TABLE 5.** Correlations

	Arithmetic logic unit	Data memory	Instruction memory	Hardwired control unit	Microprogramm ed control unit
Arithmetic logic unit	1	.179**	.396**	.112	.255**
Data memory	.179**	1	.130*	.254**	.012
Instruction memory	.396**	.130*	1	.051	.382**
Hardwired control unit	.112	.254**	.051	1	.077
Microprogrammed control unit	.255**	.012	.382**	.077	1

\*\* . Correlation is significant at the 0.01 level (2-tailed).

\* . Correlation is significant at the 0.05 level (2-tailed).

Table 5 shows the correlation between motivation parameters for Based on the provided correlation values, the highest correlation is observed between the Arithmetic Logic Unit (ALU) and the Instruction Memory, with a correlation coefficient of .396\*\*. This suggests a moderately strong positive

relationship between these two components. The next highest correlation is between the ALU and the Micro programmed Control Unit, with a correlation coefficient of .255\*\*. Again, this indicates a moderate positive relationship between these components. The correlation between the ALU and the Data

Memory is .179\*\*, which suggests a weaker positive relationship compared to the previous two correlations. The correlation between the ALU and the Hardwired Control Unit is .112, indicating a relatively weak positive relationship. The

#### 4. Conclusion

The process of creating a computer's internal structure and layout, as well as the electronic components' interactions with one another, is referred to as computer architecture design. It entails making choices about the architectural of the entire system as well as the CPU, memory hierarchy, input/output (I/O) systems, and memory hierarchy. (ISA) Instruction Set Architecture The interface that exists between software and computer hardware is described by ISA. It contains the set of instructions the CPU is capable of processing, the addressing modes, and the register structure. The ISA decision has an impact on CPU architecture and programmed compatibility. CPU Design: The brains of a computer system, the CPU.

A control unit, an arithmetic logic unit (ALU), registers, and data routes are all included in its architecture. Pipelining, superscalar implementation, out-of-order execution, and branched prediction algorithms are design options that can boost performance. Research on computer design is crucial for developing computer technologies and scientific knowledge. The capabilities of the technology underneath versus the requirements of the applications are separated by computer architecture. System designers must constantly innovate to develop technologies that can achieve the necessary performance as well as price effectiveness when application needs change and technology overcome various limits. To offer end-to-end improvement in performance at historical levels in the face of

weakest correlation is observed between the Data Memory and the Micro programmed Control Unit, with a correlation coefficient of .012. This suggests a very weak positive relationship between these components.

technology limitations is our challenge as system designers. By concentrating on energy optimization at all levels, we can overcome this difficulty Data needed for programmer functionality is stored in data memory (RAM).

It may or may not be divided into two portions, depending on the programmer in use. The data memory is believed to have two parts for the DSP2 instruction set. A computer's primary storage, usually referred to as primary Data is held in a segment called storage or memory so that the CPU can access it quickly. Primary or main storage is commonly referred to as random access memory, also known as RAM, or memory. A technique for producing control signals using Finite State Machines (FSM) is known as hardwired control. It is created by physically joining parts like flip-flops, gates, and drums to create a continuous logic circuit. It is referred to be a stiff controller as a result. What is a connected control unit? An approach to generating control signals with finite states Machines (FSM) is known as hardwired control.

It is created by physically connecting parts like flip-flops, gates, and drums to create a serial logic circuit. The Cronbach's Alpha Reliability result. The overall Cronbach's Alpha value for the model is .860 which indicates 86% reliability. From the literature review, the above 50% Cronbach's Alpha value model can be considered for analysis.

#### References

1. Guo, Lei, Zhaolong Ning, Weigang Hou, Bin Hu, and Pengxing Guo. "Quick answer for big data in sharing economy: Innovative computer architecture design facilitating optimal service-demand matching." *IEEE Transactions on Automation science and engineering* 15, no. 4 (2018): 1494-1506.
2. Agerwala, Tilak, and Siddhartha Chatterjee. "Computer architecture: Challenges and opportunities for the next decade." *IEEE Micro* 25, no. 3 (2005): 58-69.
3. Treleaven, Philip C., David R. Brownbridge, and Richard P. Hopkins. "Data-driven and demand-driven computer architecture." *ACM Computing Surveys (CSUR)* 14, no. 1 (1982): 93-143. <https://doi.org/10.1145/356869.356873>
4. Kozyrakis, Christoforos E., and David A. Patterson. "A new direction for computer architecture research." *Computer* 31, no. 11 (1998): 24-32.
5. Agüero, Ulises, and Subrata Sasgupta. "A plausibility-driven approach to computer architecture design." *Communications of the ACM* 30, no. 11 (1987): 922-932. <https://doi.org/10.1145/32206.32208>
6. Lockhart, Derek, Gary Zibrat, and Christopher Batten. "PyMTL: A unified framework for vertically integrated computer architecture research." In *2014 47th Annual IEEE/ACM International Symposium on Micro architecture*, pp. 280-292. IEEE, 2014.
7. Nikolic, Bosko, Zaharije Radivojevic, Jovan Djordjevic, and Veljko Milutinovic. "A survey and evaluation of simulators suitable for teaching courses in computer architecture and organization." *IEEE Transactions on Education* 52, no. 4 (2009): 449-458.
8. Barua, Susamma. "An interactive multimedia system on" computer architecture, organization, and design". *IEEE Transactions on Education* 44, no. 1 (2001): 41-46.
9. Martínez-Monés, Alejandra, Eduardo Gómez-Sánchez, Yannis A. Dimitriadis, Iván M. Jorrín-Abellán, Bartolomé Rubia-Avi, and Guillermo Vega-Gorgojo. "Multiple case studies to enhance project-based learning in a computer

- architecture course." *IEEE Transactions on Education* 48, no. 3 (2005): 482-489.
10. Kudithipudi, Dhireesha, Qutaiba Saleh, Cory Merkel, James Thesing, and Bryant Wysocki. "Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing." *Frontiers in neuroscience* 9 (2016): 502. <https://doi.org/10.3389/fnins.2015.00502>
  11. Ferrández-Pastor, Francisco Javier, Juan Manuel García-Chamizo, Mario Nieto-Hidalgo, and José Mora-Martínez. "Precision agriculture design method using a distributed computing architecture on internet of things context." *Sensors* 18, no. 6 (2018): 1731. <https://doi.org/10.3390/s18061731>
  12. Lee, Jong Hyuk, Seung Eun Lee, Heon Chang Yu, and Taeweon Suh. "Pipelined cpu design with fpga in teaching computer architecture." *IEEE Transactions on Education* 55, no. 3 (2011): 341-348.
  13. Trenas, Maria A., Julián Ramos, Eladio D. Gutierrez, Sergio Romero, and Francisco Corbera. "Use of a new moodle module for improving the teaching of a basic course on computer architecture." *IEEE transactions on Education* 54, no. 2 (2010): 222-228.
  14. Bader, David A., Yue Li, Tao Li, and Vipin Sachdeva. "BioPerf: A benchmark suite to evaluate high-performance computer architecture on bioinformatics applications." In *IEEE International. 2005 Proceedings of the IEEE Workload Characterization Symposium, 2005.*, pp. 163-173. IEEE, 2005.
  15. Annaratone, Marco, Emmanuel Arnould, Thomas Gross, H. T. Kung, Monica Lam, OnatMenzilcioglu, and Jon A. Webb. "The Warp computer: Architecture, implementation, and performance." *IEEE transactions on Computers* 100, no. 12 (1987): 1523-1538.
  16. Holland, Mark, James Harris, and Scott Hauck. "Harnessing FPGAs for computer architecture education." In *Proceedings 2003 IEEE International Conference on Microelectronic Systems Education. MSE'03*, pp. 12-13. IEEE, 2003.
  17. Cha, Hyun-Jong, Ho-Kyung Yang, and You-Jin Song. "A study on the design of fog computing architecture using sensor networks." *Sensors* 18, no. 11 (2018): 3633.
  18. August, David, Jonathan Chang, Sylvain Girbal, Daniel Gracia-Perez, Gilles Mouchard, David A. Penry, Olivier Temam, and Neil Vachharajani. "Unisim: An open simulation environment and library for complex architecture design and collaborative development." *IEEE Computer Architecture Letters* 6, no. 2 (2007): 45-48.
  19. Djordjevic, Jovan, Bosko Nikolic, and Aleksandar Milenkovic. "Flexible web-based educational system for teaching computer architecture and organization." *IEEE Transactions on Education* 48, no. 2 (2005): 264-273.
  20. Zou, Caifeng, Huifang Deng, and Qunye Qiu. "Design and implementation of hybrid cloud computing architecture based on cloud bus." In *2013 IEEE 9th international conference on Mobile ad-hoc and sensor networks*, pp. 289-293. IEEE, 2013.