FPGA-based Design of Multiple Models for Industry 4.0 Cyber Security

Husham Ibrihem Mahdi AL-Salman¹, Phaklen Ehkan¹, Muataz Hamed AL-Doorii

¹Centre of Excellence Advanced Computing (AdvComp), Faculty of Electronic Engineering Technology, Universiti Malaysia Perlis, Pauh Putra Campus, 02600 Arau, Perlis, Malaysia
²Flex System Penang, Batu Kawan Industrial Park, PMT 719 Lingkaran Cassia Selatan, 14110 Simpang Ampat, Penang, Malaysia

husham.alsalman90@gmail.com, phaklen@unimap.edu.my

Abstract. Industry 4.0 computing infrastructure depends on traditional processing platforms, and all these platforms perform well and achieving acceptable throughput. Two significant problems confront with these platforms are the delay in data processing and security hazard for potential threats. A design coding has been written using VHDL based on behaviour temporal parallelism for Industry 4.0 cybersecurity model. The models are divided into three namely, single, dual and quad models. Every model has three main components include UART, FIFO and DES algorithm for data encrypton and decryption. Encryption is achieved more than one block at a time and over the number of designs to find the most suitable application for data processing. A DES algorithm based efficient implementing the design of 16-round multiple parallel models for low-cost, scalable, and robust encryption solution of a maximum clock frequency of 227MHz and it is capable of encrypting or decrypting data blocks at a rate of throughput 58,288Mbps.

1. Introduction

Cryptography is the science of using mathematics to transform intelligible information to unintelligible data. Cryptography allows sensitive information to be stored or transmitted across unsecured networks, so that no one other than the intended recipient can read it. Two techniques, the symmetric key and the asymmetric key, can ensure this. For encryption and decryption, symmetric key cryptography includes the use of the same key. The asymmetric key, on the other hand involves the use of an encryption key and another distinct decryption key. Data Encryption Standard (DES), Advanced Encryption Standard (AES), Triple Data Encryption Algorithm (3DES), International Data Encryption Algorithm (IDEA), and Blowfish algorithms are those among included in secret key cryptography, and Rivest–Shamir–Adleman (RSA), Digital Signature and Message Digest algorithms are included in public key cryptography [1][2]. An encryption standard for the protection of confidential information is the DES [3] and it was developed at IBM in the 1970s and adopted by the National Institute of Standards and Technology as a Federal Information Processing Standard since 1977. Many applications that require data confidentiality have used DES pervasively [4][5].

This paper describes the design procedure which applied to research that is the implementation of embedded concurrent security architecture using FPGA-based technology. FPGAs have been widely
used in computing acceleration [6], image processing [7] [8], communication system [9], and other areas due to the continuous improvement in quality and the decrease in production cost [10]. All mentioned algorithms have been taken into consideration and study has been carried out to find a suitable design for the requirements and to fully utilize the possibilities of this technology. This study is also characterizing a choice of tools in software and hardware as well as the performance and efficiency of it.

2. Related Works
All the computational systems are made of interconnecting components and based on their abstraction level at which these systems are studied. The components include gates, memories, transistors, arithmetic units, registers, and processors. At every abstraction level, there are two basic methods in which these components are composed for creating parallel computing structures, i.e., temporal parallelism and spatial parallelism [11]. The major difference between the spatial and temporal parallelisms is that the spatial parallelism consists of several tasks that are simultaneously executed [12]. Figure 1 shows the different process between temporal and spatial parallelisms.

![Spatial and temporal parallelism process flow](image-url)
Table 1 shows the related work on spatial and temporal parallelism. In some device architectures that cover interconnections with all operating units, the spatial parallelism function is used. Previously, most researches were focused on spatial parallelism at the level of interconnection and connectivity.

| Authors                        | Platform          | Method                                    | Outcomes                                                                 |
|--------------------------------|-------------------|-------------------------------------------|--------------------------------------------------------------------------|
| S. Debnath [14]                | PC/Simulation     | Modular Multilevel Converters (MMCs)       | Up to 3.47 times faster simulation than traditional serial-in-time simulation algorithms. |
| H. M. Waidyasooriya, M. Hariyama [15] | PC/Quantum Monte Carlo | Simulated Quantum Annealing (SQA)         | Processing velocity of the planned implementation is more than 97.7 times greater. |
| N. A. A. Khalid, M. H. Salih [11] | FPGA             | Tracking system                           | Frequency of the board gave a maximal value of 1.3 GHz.                  |
| K. Nagasu et. al. [16]         | FPGA              | Stream Processing Part (SPEs)             | Achieves the highest output per power, 3.18 (GFlop/s)/W                  |

The robot monitoring device for their analysis has been observed in [11]. The DE0 Nano board frequency was set to a maximum value of 1.3GHz and this presented the frequency requirement of the platform was achieved. The logic elements used in this project is 4,022. This algorithm to simulate modular multilevel converters (MMCs) seems as one of the most difficult large-scale electronics programs. The use of this algorithm in verification of the MMCs results in up to 3.47 times faster than the conventional serial-in-time simulation algorithms when simulated in five parallel cores. The different between them is that the spatial parallelism refers to a decomposition wherein the data can be spatially partitioned for all the available processes. Every process applies the same computational set on its data. After carrying out spatial parallelism, the time steps are serially processed, i.e. initially; the 0-time step is processed, followed by time step 1, and so on. However, one disadvantage of using the spatial parallelism is that an increase in the number of processes leads to every time step being partitioned spatially into many pieces, where every piece gets smaller. This seems to adversely affect the reading pattern and decreases the I/O performance. In temporal parallelism, many time steps are processed in parallel. It is a type of pipeline parallelism, wherein several pipelines are instantiated for processing the multiple inputs simultaneously [11]. The temporal parallelism generally needs larger memory, as every process loads the complete time step.

3. Methodology

3.1. Single Model Design

This research activity was performed based on the fast design approach. For increasing the throughput, pipeline architecture had been used. As can be seen from figure 2, this particular architecture provided a block of cipher for each block cycle which excluded the initiation of the architecture in which a pipeline needed to be filled up. Latency containing sixteen clock cycles decreased the critical path to a single stage. It could be performed by using two 32 bits registers at the end of every round to hold a partial cipher. Main target of the design was to include various solutions for both sides by using the research work and providing security aspect of communication such as IoT and iOS. This ability of
FPGA technology had been adopted as a solution for resolving this kind of challenges and other benefits of this technology. It has two main parts as follow:

a) *Encryption process:* It helps to make two permanent permutations which could be called as initial, and final permutations and contains sixteen rounds. The overall representation for DES encryption had been elaborated in figure 2. There are two input functions: encrypted plaintext as well as the key. In this particular case, the plaintext and key were in 64 bits and 56 bits, respectively.

![Figure 2. General Structure of DES (encryption and decryption processes)](image)

b) *Decryption process:* Figure 3 shows the detailed description regarding DES algorithm was performed. It was also highlighted various parts of RAM as input data. At the end, component of the output was highlighted in an efficient manner (*dec_tx_out* as encrypted data, and *en_tx_out* as decrypted data). Table 2 disclosed the specifications performances of single model.

![Figure 3. RTL view of single model design](image)
Table 2. Specifications performances of project for single model

| FPGA Resources    | Usage / Total | Percentage |
|-------------------|---------------|------------|
| Total logic elements | 7,047 / 49,760 | 14%        |
| Total memory bits  | 36,864 / 1,677,312 | 2%        |
| Total PLLs         | 1 / 4         | 25%        |
| Total registers    | 2317          | -          |

3.2. Dual model design

The main part of this architecture is the efficient use of FPGA feature. There was a transmission time of sixteen cycles on the clock. For achieving the maximum creation, configuration of two paths needed to be provided.

![DES encryption and decryption algorithm in dual model](image)

Figure 4. DES encryption and decryption algorithm in dual model

It had been performed by using two paths having parallel configuration. Clock independent procedures were exploited at each level of paring calculations for developing an improved parallel design. In this exact case, it was observed that, the upper level of design followed by the internal parts is shown figure 4. The RTL view for dual model design and the specifications performances of a project for dual model are revealed in figure 5 and table 3, respectively.
### Table 3. Specifications performances of project for dual model

| FPGA Resources          | Usage / Total | Percentage |
|-------------------------|---------------|------------|
| Total logic elements    | 14,042 / 49,760 | 28%        |
| Total memory bits       | 73,728 / 1,677,312 | 4%        |
| Total PLLs              | 1 / 4         | 25%        |
| Total registers         | 4575          | -          |

### 3.3. Quad model design

For reaching to maximum outcomes, it was suggested to configure four paths into one phase. Dual method is totally depending on the parallel processing of two motors and provided a productive capacity for dual concept from the initial methodology. It was performed by using logical deduction of latest approaches that would be tested for generation of higher flow capacity than first and second method, respectively.
It was performed by using four parallel path configurations and from the same clock independent procedures at each level of paring calculations to develop an improved parallel design (figure 6). In this case, it was highlighted the upper level of design which was followed by its internal parts. Figure 7 displays the RTL view of multi model design while table 4 is presenting the system performances specification of quad model.

![Figure 7. RTL view of quad model design](image)

**Table 4. Specifications performances of project for quad model**

| FPGA Resources       | Usage / Total      | Percentage |
|----------------------|--------------------|------------|
| Total logic elements | 27,962 / 49,760    | 56%        |
| Total memory bits    | 147,456 / 1,677,312| 9%         |
| Total PLLs           | 1/4                | 25%        |
| Total registers      | 9078               | -          |

**4. Results and Discussion**

The flow of data was measured of the superiority of the designed models from each other to reach the target of methodology. By using the available data, the experiments had been carried out and the obtained results are revealed in Table 5.

**Table 5. Comparison of the designed models**

| Model | Frequency Maximum, Fmax (MHz) | Time (μs) | Input | Throughput (Mbps) |
|-------|-------------------------------|-----------|-------|-------------------|
| Single| 288.77                        | 3.4       | 36,864| 18,432            |
| Dual  | 247.97                        | 4.03      | 73,728| 31,616            |
| Quad  | 227.69                        | 4.3       | 147,456| 58,288           |
The highest frequency was at the single, followed by dual and then, quad model because of the availability of data and logic elements in double and quad models. Second criteria represent the time which is highlighted the level of frequency used. The maximum time required to perform encryption or decryption of data regarding the text of this algorithm is \(1/F_{\text{max}} \text{ (MHz)} = N \text{ nanosecond}\). In this particular case, it was found that the amount of time required for encryption or decryption tends vertically up to the quad design and thus, the most consuming of it. After the calculation of time consumed and frequency, it was clearly showed that there was a direct relation between the two-input data which might increase the complexity of design. Therefore, it was linked with the frequency in an ascending curve with the input of data. The important criteria were to provide a proper flow direction of the research and also highlighted the advantages what it contains in terms of industrial and scientific standing point.

Figure 8 demonstrates the frequency result of single, dual and quad models design. As can be observed that the increased of design capacity for processing which increased in any kind of repetition of algorithm. It provided the ability of FPGA to perform proper processing and provided a huge amount of advantages efficiently. One of these was in the format of optimization and preserved the design by using different principles. It was explained earlier and desired to increase by multiple design. A throughput referred to the amount of data transferred from one place to another and it helped to achieve the frequency and considered clock cycle latency for first time. It performed proper encryption of one data per clock cycle. Therefore, the achieved throughput was \((\text{frequency } \times \text{ data per clock}) = n \text{ Mbps}\).
5. Conclusion
The design presented is describing the working principle of DES algorithms, the top-level structure of FPGA containing its parts, a detailed description of encryption, processing of encryption, and performing parallel modelling by the use of Intel Quartus Prime 15.1 Lite Edition platform. The validation and evaluation were done by applying time quest timing analyser tools. At a clock frequency of 227MHz, the quad model can encrypt and decrypt data blocks at a rate of 58,288 Mbps. The quad design method is the optimal way in terms of data flow which are 85% more than dual model and 300% from a single model, respectively.

References
[1] Vuppala A Roshan R S Nawaz S and Ravindra J V R 2020 An Efficient Optimization and Secured Triple Data Encryption Standard using Enhanced Key Scheduling Algorithm Procedia Computer Science 171 1054–1063.
[2] Patil P Narayankar P Narayan D G and Meena S M 2016 A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish Procedia Computer Science 78 617–624.
[3] Hoobi M M 2020 Improved Structure of Data Encryption Standard Algorithm Journal of Southwest Jiaotong University 55(5) 1-10.
[4] Oukili S and Bri S 2016 High Throughput FPGA Implementation of Data Encryption Standard with Time Variable Sub-keys Int. Journal Electrical and Computer Engineering 6(1) 298–306.
[5] Ikhwon A Raof R A A Ehkan P Yacob Y Syafuddin M 2021 Data Security Implementation using Data Encryption Standard Method for Student Values at the Faculty of Medicine, University of North Sumatra Journal of Physics: Conference Series 1755(1) 1-11.
[6] Ehkan P Allen T and Quigley S F 2011 FPGA Implementation for GMM-Based Speaker Identification International Journal of Reconfigurable Computing 2011 1-8.
[7] Lam C Y Ehkan P Raof R A A and Jungjit S 2021 Real Time Detection of Object Blob Localization Application using 1-D Connected Pixel and Windowing Method on FPGA Journal of Physics: Conference Series 1755(1) 1-11.
[8] Kan P L E Taha T B and Ngadiran R 2020 Design and Implementation of Lifting Wavelet Transform Using Field Programmable Gate Arrays IOP Conference Series: Materials Science and Engineering 767 1-6.
[9] Noh N A M Saparon A and Hashim H 2015 Real Time FPGA Communication System Using Ethernet for Robotics, Procedia Computer Science 76 406-410.
[10] Zhang J and Qu G 2019 Recent Attacks and Defenses on FPGA-based Systems ACM Transactions on Reconfigurable Technology and Systems 12(3), 1–24.
[11] Khalid N A A and Salih M H 2017 Design and Implementation of Embedded Tracking System using Spatial Parallelism on FPGA for Robotics ARPN Journal of Engineering and Applied Sciences 12(23) 6725–6731.
[12] Ibrahim H and Salih M H 2018 Design and Implementation of Embedded True Parallelism Jammer System using FPGA-SoC for Low Design Complexity ARPN Journal of Engineering and Applied Sciences 13(24) 9410–9420.
[13] Guo H et al 2017 FPGA implementation of VLC Communication Technology, International Conference on Advanced Information Networking and Applications Workshops 586–590.
[14] Debnath S 2020 Parallel-in-Time Simulation Algorithm for Power Electronics: MMC-HVdc System IEEE Journal of Emerging and Selected Topic in Power Electronics 8(4) 4100–4108.
[15] Waidyasooriya H M and Hariyama M 2020 A GPU-based Quantum Annealing Simulator for Fully-Connected using Models Utilizing Spatial and Temporal Parallelism IEEE Access 8 67929–67939.
[16] Nagasu K. et al 2016 Parallelism for High-Performance Tsunami Simulation with FPGA: Spatial or Temporal?, International Symposium on Field-Programmable Custom Computing Machines (FCCM), 30-50.