Wadi Alshatti University Journal of Pure and Applied Sciences

مجلة جامعة وادي الشاطئ للعلوم البحتة والتطبيقية

Volume 3, No. 2, July-December 2025 Online ISSN: 3006-0877 المجلد 3، الاصدار 2، يوليو- ديسمبر 2025

RESEARCH ARTICLE **APPLIED MECHANICAL ENGINEERING**

Hardware Implementation of Newton-Raphson Method for Nonlinear Stress Estimation

Shatha AbuShanab[1,*], Ashraf Hadoush[2]

[1]Computer Systems Engineering Department, Palestine Technical University –Kadoorie, Palestine
[2]Mechanical Engineering Department, Palestine Technical University – Kadoorie, Palestine

ARTICLE HISTORY
Received 08 July 2025
Revised 31 August 2025
Accepted 03 September 2025
Online 05 September 2025

KEYWORDS
Stress estimating;
Newton–Raphson;
Digital circuit;
Field Programmable Gate Array;
VHDL Language.

ABSTRACT

This research aims to develop and implement digital design for computing the uniaxial Cauchy stress of an internally balanced hyperelastic material using the Newton-Raphson (NR) iterative algorithm at a hardware level, such as Field Programmable Gate Array (FPGA) technology. FPGA technologies are semiconductor devices which provide a parallel processing technique essential for real-time systems when computational speed is critical. The NR algorithm is investigated based on FPGA to solve the internally balanced equation, which is developed at a high level of abstraction and then implemented using physical instruments to obtain actual data. The NR algorithm is programmed using VHDL language. Then, the design is simulated by ModelSim Intel program. The achieved results using hardware approach are compared with a MATLAB script to verify the functionality of the digital design. The digital system is fully implemented on the Cyclone V FPGA device using the Quartus Intel program. The accuracy and precision of the results obtained at the hardware level are acceptable, within the given threshold, depending on the data type representing the fractional number used for the computation, as well as the resources available, which depend on the FPGA technology used.

تقدير الضغط غير الخطي باستخدام FPGA

شذى أبوشنب [1،*]، أشرف حدوش [2]

الكلمات المفتاحية

تقدير الضغط
نيوتن-رافسون
الدوائر الرقمية
مصفوفة البوابات القابلة للبرمجة
لغة VHDL

الملخص

يهدف هذا البحث إلى تطوير وتنفيذ تصميم رقمي لحساب ضغط كوشي أحادي المحور لمادة فائقة المرونة متوازنة داخليًا باستخدام خوارزمية نيوتن-رافسون التكرارية على مستوى الأجهزة، مثل تكنولوجيا تقنيات مصفوفة البوابات القابلة للبرمجة (**FPGA**). تقنية **FPGA**هي فعليا أجهزة أشباه موصلات توفر تقنية معالجة متوازية أساسية لأنظمة الوقت الفعلي عندما يكون وقت الحساب حرجًا. تمت دراسة خوارزمية نيوتن-رافسون بناءً على **FPGA** لحل المعادلة المتوازنة داخليًا والتي تم تطويرها على مستوى عالٍ من التجريد ثم تنفيذها باستخدام أدوات مادية للحصول على بيانات حقيقيه. استخدمت لغة **VHDL** لبرمجة خوارزمية نيوتن-رافسون ، ويحاكي برنامج **ModelSim Intel** التصميم الموصوف. تمت مقارنة القيم الناتجة من استخدام الهاردوير بالقيم الناتجة من استخدام نص **MATLAB** للتحقق من وظائف التصميم الرقمي. تم تنفيذ النظام الرقمي بالكامل على شريحة **Cyclone V FPGA** باستخدام برنامج **Quartus Intel**. إن دقة وصحة النتائج التي تم الحصول عليها على مستوى الأجهزة مقبولة، فيما يتعلق بالعتبة المحددة، اعتمادًا على نوع البيانات التي تمثل العدد الكسري للعملية الحسابية، وكذلك الموارد المتاحة التي بدورها تعتمد على تقنية **FPGA** المستخدمة.

Introduction

Rubber–like material is highly deformable to a large extent. A typical biological example is a muscle that is stimulated by a nerve impulse, resulting in contraction. This action is modelled form a mechanical point of view as a change of stress resulting from a change of material parameter and length. The stress is often expressed as a nonlinear function in terms of material parameters and stretching [1, 2]. This often requires to perform advanced calculation that needs some software running on a computer to determine the value of stress to make further decisions. The evaluation and rapid development of microelectronics technology pay to a change in the design approach. FPGA technologies with powerful and low-cost parallel processing are targeted for high-performance digital systems. Subsequently, many researchers have enhanced their design when real-time and parallel processing are needed by applying a digital system based on microelectronics technologies, such as FPGA chips [3]. FPGA is a general-purpose programmable logic device that can be reprogrammed completely with a simple digital design or a complicated design on a chip. Nowadays, FPGA technologies are experiencing rapid development in architecture, packaging, and design tools. An appropriate digital design is required for the better utilization of resources in the target FPGA [4–7].

Ebrahimi and Zandsalimy propose using FPGAs as alternatives to CPUs for accelerating the numerical solution of fluid dynamics differential equations. The research aims to enhance the speed of numerical solutions through hardware implementation. A Zynq-7020, a reconfigurable

*Corresponding author
https://doi.org/10.63318/waujpasv3i2_30

device, was utilized. Several IP blocks were built using the C++ programming language to build the FPGA architecture instead of hardware description language (HDL). The results demonstrate that FPGAs significantly increase solution speed compared to CPUs [8].

Armi, Manai, and Besbes present an asymmetrical structure designed to reduce the number of bridges, gate drive circuits, and DC sources. The design undergoes experimental testing using an FPGA to implement it based on the Newton-Raphson (NR) algorithm. The results indicate success in eliminating the 5th, 7th, and 11th harmonics of the output voltage, and the hardware results closely match those of the simulation [9].

Zhang and Sun propose using parallel and distributed computing to meet the speed needed for such real-time simulation, such as an FPGA, due to its parallel technique instead of conventional computing devices. FPGA was used for real-time simulation of power electronic converters and systems to solve the discretized model by the Jacobi iterative method. The results were validated, but some differences can be noticed between the FPGA simulation and SABER simulation, which may be due to the number of iterations employed not being sufficient for convergence [10].

This paper discusses the details of developing and implementing digital design for computing the uniaxial Cauchy stress of an internally balanced hyperelastic material using the Newton-Raphson iterative algorithm at the hardware level. Because FPGAs provide parallel techniques, they are attractive for computational challenges involving the iteration of nonlinear equations. Furthermore, cost arithmetic operations in floating-point require an iterative method that is clearly and simply specified [6, 11]. Design methodology is adopted to ensure better resource utilization and faster computational operations. Due to the need for a real-time system and parallel processing for interactive processes, CMOS technology, such as FPGA technology, is targeted for use. A low-cost Cyclone V SE FPGA is utilized for hardware implementation. HDL is chosen to describe the behavior of the design. Additionally, the digital design is carried out using three sequential steps to ensure the accuracy and success of the hardware implementation using the MATLAB environment, ModelSim Intel, and Quartus Intel programs.

The paper is divided into four sections. This section provides an introduction to the research that developed and implemented the uniaxial Cauchy stress of an internally balanced hyperelastic material on an FPGA device. In the next section, the implementation of the digital design and the methodology of the research have been discussed. The penultimate section presents and discusses the results obtained. Then, the conclusion can be found.

Stress Analysis

The main objective is to calculate the uniaxial Cauchy stress of an internally balanced hyperelastic material at the hardware level. An internally balanced Blatz–Ko constitutive model is used [12, 13]. It is a generalized material model that is formulated in analogy to a special case of the Blatz–Ko constitutive model [14–16] that is suitable to predict the stress of soft material at large deformation. The internal balanced uniaxial Cauchy stress σ for the Blatz–Ko model is given as:

$$\sigma = \frac{\mu(1+\beta)\left(\hat{\lambda}^2 - \hat{\lambda}^{-1/2}\right)}{\lambda^{1/2}} \quad (1)$$

where μ is the infinitesimal shear modulus and λ is the stretch ratio that is defined as the ratio of the deformed length L to the relaxed Length L_0. The internally balanced decomposed stretch ratio $\hat{\lambda}$ is determined by solving the internally balanced equation:

$$f(\eta,\zeta,\beta) = \beta\,\zeta\,\eta^8 + \eta^5 - \beta\,\zeta\,\eta^3 - \zeta^5 = 0 \quad (2)$$

where $\zeta = \lambda^{1/2}$ and $\eta = \hat{\lambda}^{1/2}$. The internally balanced material parameter β is used to quantify the contribution of $\hat{\lambda}$. A normalized Cauchy stress $\sigma`$ can be expressed as:

$$\sigma` = \frac{\sigma}{\mu} = \frac{(1+\beta)\left(\eta^4 - \frac{1}{\eta}\right)}{\zeta} \quad (3)$$

Newton–Raphson Method

The value of η has to be determined by solving the nonlinear internal balance equation in Eq. (2) for given inputs ζ and β to calculate $\sigma`$ as expressed in Eq. (3). The Newton–Raphson method is one of the well-known iterative procedures to solve nonlinear equations, i.e., find the root of the nonlinear equation [17]. It starts by making an initial guess of the root η_i, then it predicts an estimate of the root η_{i+1}, such as Eq. (4)

$$\eta_{i+1} = \eta_i - \frac{f(\eta_i,\zeta,\beta)}{f`(\eta_i,\zeta,\beta)} \quad (4)$$

where $f`$ is the derivative of f with respect to η. This iterative procedure stops when the percentage relative error is less than a given threshold, e.g., $\epsilon_{term} = 1\times10^{-6}$. The percentage relative error ϵ_a is defined as in Eq. (5)

$$\epsilon_a = \frac{\eta_{i+1} - \eta_i}{\eta_{i+1}}\ 100\% \quad (5)$$

Internal Balanced Uniaxial Cauchy Stress Implementation

The objective of this research is to compute internal balanced uniaxial Cauchy stress using the NR algorithm at the hardware level. It involves developing and implementing an NR algorithm based on CMOS technologies, such as FPGAs, in the design. The methodology of this study consists of three sequential steps that achieve a complete and successful design process, as well as verifying the functionality of solving nonlinear equations, as illustrated in Figure 1. The steps are as follows:

- Implementing the algorithm using MATLAB script (.m file)
- Writing, verifying, and simulating VHDL code using the ModelSim Intel program
- Synthesizing and implementing the digital design using the Quartus Intel program on the Cyclone V FPGA device

MATLAB Environment

The MATLAB environment is often used to demonstrate and compare the results obtained from the NR algorithm [18–21]. A MATLAB script is written to implement the NR algorithm and is run in the MATLAB environment. The output is a text file, which is compared and validated in a subsequent step. The inputs are ζ and β values; the ζ values are 0.25, 0.5, 1.0, and 1.5, while the β values vary at 0.0, 0.5, and 1.0 as they undergo the NRM algorithm. The output consists of the solutions to the nonlinear equations η for the given inputs, formatted as text files. Furthermore, by setting β equal to 1.0 and ζ equal to 0.5, the new value of η_{i+1} and the relative error are computed for each iteration until the solution is achieved, for the given threshold ϵ_{term}, and its results are also formatted as text files.

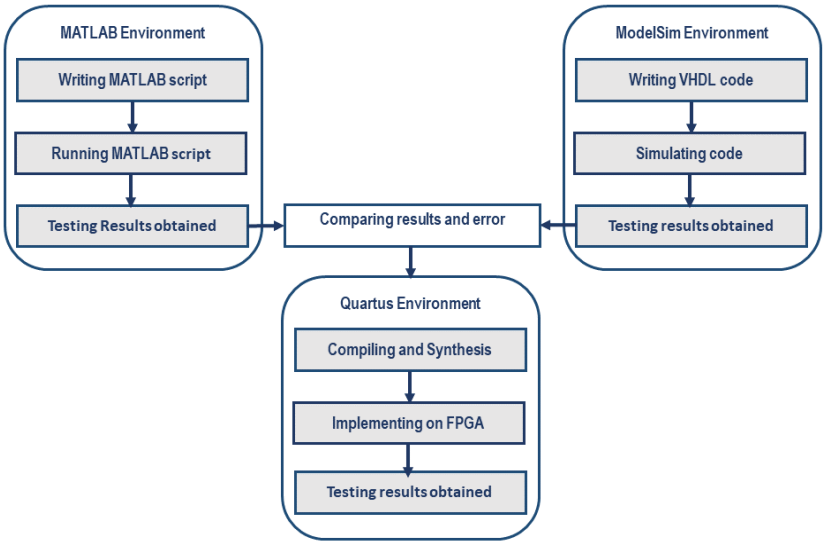

Fig. 1: The sequential steps of the methodology are used

ModelSim Intel Program

The ModelSim Intel program is used to simulate and verify the functionality of digital designs before implementing them at the hardware level (an FPGA device) [22].

Hardware Description Language (HDL) describes the behavior of digital designs using a text-based language. HDLs facilitate the description of large digital systems instead of schematics. This feature enables designers to design and verify the functionality of digital designs at an abstraction level during the design process, allowing for successful implementation on real hardware. VHDL is used in the design process of ASICs, FPGAs, and another digital electronic system [20].

VHDL, which stands for Very High Speed Integrated Circuit Hardware Description Language, is used to model and describe the behavior of the design needed to perform the NR iteration and obtain the estimated solution η for the nonlinear equation. By estimating η at the hardware level, the internal balanced uniaxial Cauchy stress σ can be computed [11, 23, 24].

In VHDL design, the signed fixed-point data type is crucial for representing signed fixed-point values in arithmetic operations instead of floating-point representations [11]. The fixed-point data type requires a simpler hardware implementation than floating-point representations for fractional numbers. Consequently, simpler hardware implementations offer a quick and effective solution for real-time systems, particularly for FPGAs, where resource utilization is limited due to the technology target. Moreover, this data type is significant in applications where division operations are used, as it allows for efficient quantization error compared to floating-point [24–27].

The algorithm requires both the current and new solutions of η for each iterative NR process to compute the relative error. The iterative NR algorithm terminates when the relative error is less than a threshold value of єterm. Once the nonlinear equation η is estimated, the internal balanced uniaxial Cauchy stress σ can be computed. Within the VHDL process, the current solution is maintained as the last reading stored in registers, while the new solution from iterative NR η_{i+1} is compared with the previous one to estimate the relative error. This technique at the hardware level effectively reduces resource utilization. The output data from the digital design consists of two text files: one containing various values of η and β as input, with the final solution of the nonlinear equation η_{i+1} serving as the system output. The second text file includes a ζ value of 0.5 and a β value of 1.0 as input, along with the solution and relative error for each NR iteration until the final solution is reached, following the same procedure as the MATLAB script.

For the ModelSim Intel simulation, it is essential to generate input and control signals, as well as simulate output data. The control signals are clock, start, reset, and ready signals. At the hardware level, ζ and β are read simultaneously during each clock cycle; the start signal is active high to begin the computation process, while the reset signal is active low when needed to set all initial values. Once the relative error is less than the threshold, the ready signal is activated high, indicating that the solution of ηi+1 is reached and the stress can be computed.

The output text files from the ModelSim Intel simulation are compared to the output text file generated using MATLAB; the results must closely match. At this stage, the design's functionality is verified, ensuring accurate and successful performance on an FPGA device.

Synthesis and Implementation on FPGA

The main goal of this research is to implement a digital system entirely based on FPGA technology and to investigate the data obtained at the hardware level. The digital design utilizing FPGA technologies follows a hardware-software co-design methodology. Designers start with software to describe design behavior using a high-level language, independent of the hardware (CMOS technology) devices where it will be implemented [11, 23, 24].

The system is designed to compute internal balanced uniaxial Cauchy stress and to solve the nonlinear equation using the NR algorithm. As previously stated, the inputs for the digital design will include ζ and β as data inputs, along with clock, reset, start, and ready, and as control signals, as shown in Fig. 2.

The DE10-Standard Development Kit serves as the hardware design platform, utilizing the Intel Cyclone V SE5CSXFC6D6F31C6N device [28]. Cyclone V FPGA devices depend on 28 nm semiconductor technology.

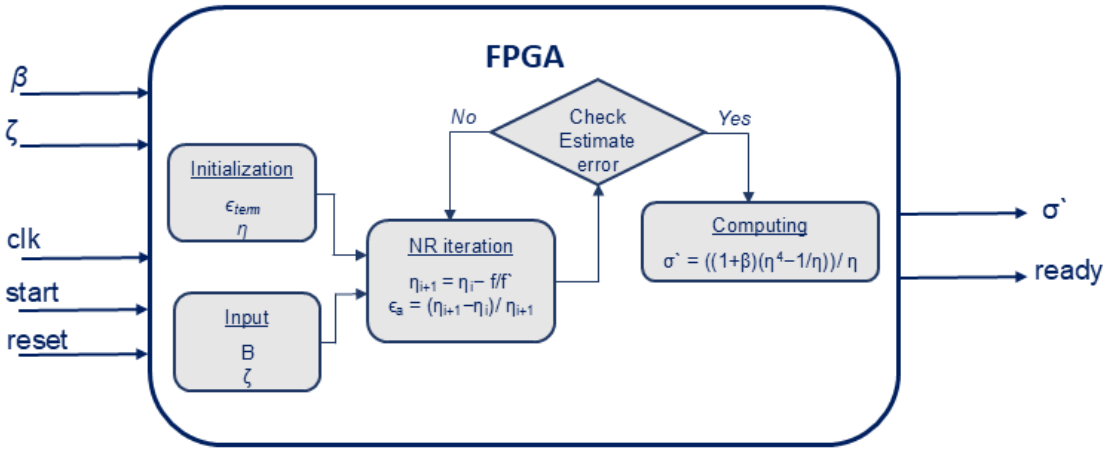

Fig. 2: The NR algorithm on the abstraction level is implemented on an FPGA device.

The DE10-Standard Development Kit is targeted for real applications based on an FPGA as a design platform, where the design is implemented, and actual results are obtained.

The Quartus Intel program is used to synthesize the design described in VHDL [29]. Synthesis involves creating a physical digital design from an abstract representation in VHDL and is also simulated using ModelSim Intel. The resources utilized are determined by the specifications and requirements of the digital system, which are independent of the FPGA technology used at the abstraction level.

Employing the DE10-Standard Development Kit allows for the implementation of the system's reliability targets. Once the synthesis process is successful, a binary file is generated to implement the design on the FPGA technology. As a result, the FPGA resources utilized for the digital design are presented in Table 1 based on the FPGA technology used, which is Cyclone V SE 5CSXFC6D6F31C6N on the DE10-Standard Development Kit.

The FPGA device's resources are determined by the digital design described in VHDL. Signed fixed-point is necessary for the design in order to perform iterative computation and reach the final result within the given threshold. Furthermore, the computing iterations require to implement mathematical operations such as division. Thus, in order to perform mathematically signed fixed point intensive iterations, 42% of the DSP blocks on Cyclone V SE 5CSXFC6D6F31C6N FPGA devices are utilized in the digital design.

Table 1: FPGA resources utilized for Cyclone V SE 5CSXFC6D6F31C6N.

Resources used Cyclone V SE 5CSXFC6D6F31C6N	
Logic utilization (ALMs)	4,913/41,910 (12%)
Register	69
DSP blocks	47/112 (42%)

Results and Discussion

The three sequential steps are conducted to implement and apply the digital design of an internal balanced uniaxial Cauchy stress using Newton-Raphson algorithms. The data obtained from the MATLAB script and ModelSim Intel are tested and compared, followed by synthesizing and implementing the design on the available FPGA device using the DE10-Standard Development kit.

Figures 3 and 4 present the results obtained from running a MATLAB script and from the ModelSim Intel simulation as text files, as previously mentioned.

```
#Output from MATLAB
                    Beta=0.0       Beta=0.5       Beta=1.0
-------------------------------------------------------------------
Zeta = 0.250000   |0.2500000000   0.3771583841   0.5000000052
Zeta = 0.500000   |0.5000000000   0.6073270867   0.7071067812
Zeta = 1.000000   |1.0000000000   1.0000000000   1.0000000000
Zeta = 1.500000   |1.5000000000   1.2873604939   1.2247448729
```

Fig. 3: The output text file from the MATLAB script for different values of β and ζ, and the computed η_{i+1}

```
# Output from Testbench
                         Beta=0.0         Beta=0.5         Beta=1.0
-----------------------------------------------------------------------
Zeta = 2.500000e-01|    2.500005e-01     3.771584e-01     5.000000e-01
Zeta = 5.000000e-01|    5.000000e-01     6.073271e-01     7.071068e-01
Zeta = 1.000000e+00|    1.000000e+00     1.000000e+00     1.000000e+00
Zeta = 1.500000e+00|    1.500000e+00     1.287361e+00     1.224745e+00
```

Fig. 4: The output text file from the ModelSim Intel for different values of β and ζ, and the computed η_{i+1}

Figures 5 and 6 illustrate the relative error ϵ_a and the solutions of the nonlinear equation η_{i+1} for each NR iteration from the MATLAB script and ModelSim Intel for ζ equal to 0.5 and β equal to 1.0, and stops when the relative error is below the specified threshold, which requires a certain number of iterations to achieve the solution of the nonlinear equation. Eleven iterations are required for the MARLAB and ModelSim Intel programs to reach the final solution, which is dependent on the initial value and threshold that are given.

```
   #Output from MATLAB Beta=1.0 and  Zeta=0.5

#       Eta             Errror
-----------------------------------------------
   1   1.7338417235   0.1330791382
   2   1.4987093856   0.1356134962
   3   1.2918560769   0.1380209604
   4   1.1121953122   0.1390718114
   5   0.9607199871   0.1361948962
   6   0.8408741041   0.1247459037
   7   0.7583672055   0.0981203943
   8   0.7172790333   0.0541797851
   9   0.7075921540   0.0135050361
  10   0.7071079442   0.0006843064
  11   0.7071067812   0.0000016448
  12   0.7071067812   0.0000000000
```

Fig. 5: The output text file from the ModelSim Intel for different values of β and ζ, and the computed η_{i+1}

Figure 7 displays the output waveform from the simulation of the ModelSim when ζ equals 0.5 and β equals 1.0, which is also saved as a text file, as shown in Fig.7. The waveform shows the input, output, and control signals. Every clock signal an iterative process is performed and compute η_{i+1} and the relative error, which is compared with the threshold, and this process is repeated until the relative error is less than the threshold, then the solution is reached, and the ready signal is active high, which enables the system to compute the stress σ.

The results obtained from the ModelSim Intel program to verify the functionality of the digital design described in VHDL are compared with the results obtained from the MATLAB script; the results, as observed, are closely matched, which confirms that the algorithm successfully and

```
# Output from Testbench Beta=1.0 and Zeta=0.5

#       Eta             Errror
-----------------------------------------------
1      1.733842e+00      1.330791e-01
2      1.498709e+00      1.356135e-01
3      1.291856e+00      1.380210e-01
4      1.112195e+00      1.390718e-01
5      9.607200e-01      1.361949e-01
6      8.408741e-01      1.247459e-01
7      7.583672e-01      9.812039e-02
8      7.172790e-01      5.417979e-02
9      7.075922e-01      1.350504e-02
10      7.071079e-01      6.843072e-04
11      7.071068e-01      1.643724e-06
12      7.071068e-01      0.000000e+00
```

Fig. 6: The output text file from the ModelSim Intel with the ϵa and η_{i+1}

accurately finds the solution to the nonlinear equation using the digital design described in VHDL using ModelSim Intel simulation for given $\epsilon_{term} = 1 \times 10^{-6}$ is given.In the ModelSim Intel simulation, the algorithm necessitates the use of the signed fixed-point data type with a wide range to verify the functionality as well as the accuracy of the design. The output ηi+1 in ModelSim Intel and Quartus Intel has a wide length of nineteen bits for the fractional part based on the threshold value and six bits for the integer part based on the range of the system's output (maximum and minimum values). This data type is significant, and the hardware supports it to be implemented on FPGA device after verification.

Figure 8 shows the output results from the DE10-Standard Development after implementing the design on the FPGA device; the output is the solution of the nonlinear equation η. The accuracy and precision can be determined based on the given threshold for relative error, which is also accepted and satisfies the requirement for the design. Moreover, the limitations in precision and accuracy for the obtained results at the hardware level depend on the datatype used and the wide range that represents the fractional numbers, as well as the resources available at the FPGA device used.

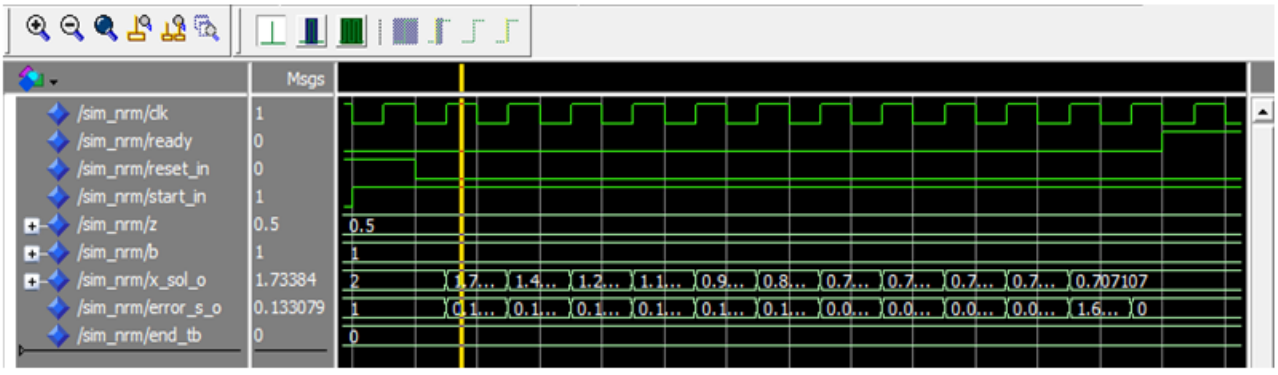

Fig. 7: The output wave from the ModelSim simulation when ζ equals 0.5 and β equals 1.0.

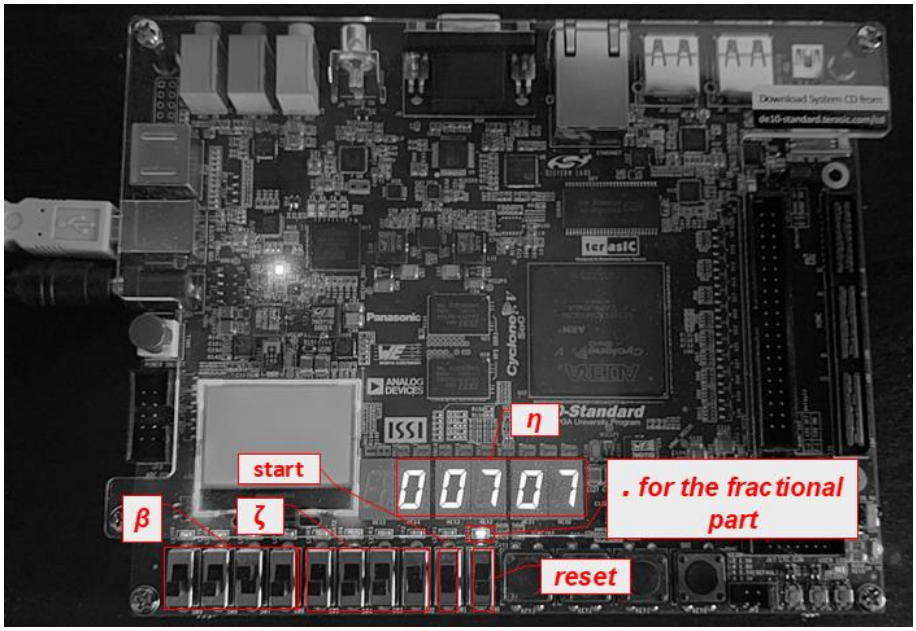

Fig. 8: The output results from the DE10-Standard Development kit when the solution η is reached.

Conclusion

The main idea of this approach is to develop and implement a digital design to compute the uniaxial Cauchy stress of an internally balanced hyperelastic material using Newton-Raphson NR iterative algorithm based on hardware such as an FPGA, which is a semiconductor device. The NR iterative algorithm is used to solve a nonlinear equation in order to computes the internal uniaxial Cauchy stress. VHDL is a hardware description language used to model and describe the behavior of the proposed digital design. VHDL enables verifying the functionality of the digital design at a high level of abstraction before implementing it on real hardware. The ModelSim Intel program simulates the design described in VHDL. After that, the digital design is tested for functionality and accuracy and compared to the output of a MATLAB script. The VHDL language used to describe the design in ModelSim Intel is synthesized using Quartus Intel to generate a binary file, which is then implemented on an FPGA device. The DE10-Standard Development Kit serves as the hardware platform, utilizing the Intel Cyclone V FPGA device. The algorithm requires the use of signed fixed-point data types for division and multiplication operations to represent fractional numbers during iterations. This data type is essential, and the hardware supports it. Additionally, signed fixed-point is faster and provides more efficient quantization errors compared to floating-point data types. The results from simulation and hardware testing show that CMOS technology for real-time systems and parallel processing, such as FPGA, can compute internal balanced uniaxial Cauchy stress using the NR iteration algorithm. Furthermore, the accuracy and precision are acceptable based on the threshold for relative error, considering hardware limitations related to the datatype used for computation and the available hardware resources.

This study investigates the development and hardware implementation of the NR iterative algorithm, which uses signed fixed-point datatype to overcome the difficulty of performing mathematical operations. These results may help future research on another algorithm and find the best iteration method based on hardware resource use, computing time, and performance.

Author Contributions: "All authors have made a substantial and direct intellectual contribution to the work and have read and agreed to the published version of the manuscript."

Funding: "This research received no external funding."

Data Availability Statement: "The data are available at request."

Acknowledgments: "The authors would like to thank Palestine Technical University – Kadoorie for supporting this research."

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] A. Hadoush. "Finite Element Formulation of Internally Balanced Blatz--Ko Material Model," *Jordan Journal of Mechanical & Industrial Engineering*, vol. 14, no. 2, 2020.

[2] A. Hadoush. "Internally Balanced Spatial Elasticity tensor," *Wadi Alshatti University Journal of Pure and Applied Sciences*, pp. 98–101, 2025.

[3] M. Yushkova, A. Sanchez, and A. de Castro. "Strategies for choosing an appropriate numerical method for FPGA-based HIL," *International Journal of Electrical Power & Energy Systems*, vol. 132, p. 107186, 2021, https://doi.org/10.1016/j.ijepes.2021.107186.

[4] H. Yang, J. Zhang, J. Sun, and Le Yu, "Review of advanced FPGA architectures and technologies," *J. Electron.(China)*, vol. 31, no. 5, pp. 371–393, 2014, https://doi.org/10.1007/s11767-014-4090-x.

[5] J. Ruiz-Rosero, G. Ramirez-Gonzalez, and R. Khanna, "Field Programmable Gate Array Applications—A Scientometric Review," *Computation*, vol. 7, no. 4, p. 63, 2019, https://doi.org/10.3390/computation7040063.

[6] P. Babu and E. Parthasarathy, "Reconfigurable FPGA Architectures: A Survey and Applications," *J. Inst. Eng. India Ser. B*, vol. 102, no. 1, pp. 143–156, 2021, https://doi.org/10.1007/s40031-020-00508-y.

[7] A. Boutros and V. Betz, "FPGA Architecture: Principles and Progression," *IEEE Circuits Syst. Mag.*, vol. 21, no. 2, pp. 4–29, 2021, https://doi.org/10.1109/MCAS.2021.3071607.

[8] A. Ebrahimi and M. Zandsalimy, "Evaluation of FPGA Hardware as a New Approach for Accelerating the Numerical Solution of CFD Problems," *IEEE Access*, vol. 5, pp. 9717–9727, 2017, https://doi.org/10.1109/ACCESS.2017.2705434.

[9] Armi, Faouzi, Lazhar Manai and Mongi Besbes., "FPGA Implementation of Selective Harmonic Elimination Controlled Asymmetrical Cascaded Nine Levels Inverter Using Newton Raphson Algorithm," in *3rd International Conference on Automation, Control, Engineering and Computer Science (ACECS'16)*, pp. 377–382.

[10] H. Zhang and J. Sun, "FPGA-based simulation of power electronics using iterative methods," in *2014 International Power Electronics Conference (IPEC-Hiroshima 2014 - ECCE ASIA)*, Hiroshima, Japan, May. 2014 - May. 2014, pp. 2202–2207.

[11] P. J. Ashenden, *The designer's guide to VHDL,* 2nd ed. San Francisco CA: Morgan Kaufmann, 2002.

[12] A. Hadoush, H. Demirkoparan, and T. Pence, "A constitutive model for an internally balanced compressible elastic material," *Mathematics and Mechanics of Solids*, vol. 22, no. 3, pp. 372–400, 2017, https://doi.org/10.1177/1081286515594657.

[13] A. Hadoush, "Effect of Poisson's ratio on internally balanced Blatz-Ko material model," *Acta Mechanica Sinica*, vol. 39, no. 5, p. 422350, 2023.

[14] P. Blatz and W. Ko, "Application of Finite Elastic Theory to the Deformation of Rubbery Materials," *Transactions of the Society of Rheology*, vol. 6, no. 1, pp. 223–252, 1962, https://doi.org/10.1122/1.548937.

[15] M. Beatty, "Topics in Finite Elasticity: Hyperelasticity of Rubber, Elastomers, and Biological Tissues—With Examples," *Applied Mechanics Reviews*, vol. 40, no. 12, pp. 1699–1734, 1987, https://doi.org/10.1115/1.3149545.

[16] A. Hadoush, H. Demirkoparan, and T. J. Pence, "Finite element analysis of internally balanced elastic materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 322, pp. 373–395, 2017, https://doi.org/10.1016/j.cma.2017.04.026.

[17] S. C. Chapra and R. P. Canale, *Numerical methods for engineers*: Singapore, 1998.

[18] L. Al Asadi, H. Bahrani, and L. Al Waeli, "Parametric Study for Design and Analysis of Box Culvert by Using Newton's-Raphson Method and MATLAB Software," *KEM*, vol. 870, pp. 11–19, 2020, https://doi.org/10.4028/www.scientific.net/KEM.870.11.

[19] A. Bernardini, E. Bozzo, F. Fontana, and A. Sarti, "A Wave Digital Newton-Raphson Method for Virtual Analog Modeling of Audio Circuits with Multiple One-Port Nonlinearities," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 2162–2173, 2021, https://doi.org/10.1109/TASLP.2021.3084337.

[20] H. Abbas, T. Musa, and A. Jihad, "Optimum hydraulic investigation of pipe aqueduct by MATLAB software and Newton–Raphson method," *Open Engineering*, vol. 12, no. 1, pp. 808–816, 2022, https://doi.org/10.1515/eng-2022-0369.

[21] S. Adak, H. Cangi, A. Yilmaz, and U. Arifoglu, "Development software program for extraction of photovoltaic cell equivalent circuit model parameters based on the Newton–Raphson method," *J Comput Electron*, 2022, https://doi.org/10.1007/s10825-022-01969-8.

[22] Intel, *ModelSim-Intel® FPGAs Standard Edition Software Version 18.1.* [Online]. Available: https://www.intel.com/content/www/us/en/software-kit/750368/modelsim-intel-fpgas-standard-edition-software-version-18-1.html (accessed: Jun. 4 2025).

[23] J. Lo, *Modern digital designs with EDA, VHDL and FPGA*. Hsinchu City, Taiwan: Terasic Inc, 2015.

[24] B. LaMeres, *Quick Start Guide to VHDL*. Cham: Springer International Publishing, 2024.

[25] E. Tlelo-Cuautle, J. Rangel-Magdaleno, and L. de La Fraga, *Engineering applications of FPGas*. New York NY: Springer Berlin Heidelberg, 2016.

[26] Artur Gramacki, *Nonparametric kernel density estimation and its computational aspects*. New York NY: Springer Berlin Heidelberg, 2018. https://link.springer.com/content/pdf/10.1007/978-3-319-71688-6.pdf

[27] D. Sankar, L. Syamala, B. Chembathu Ayyappan, and M. Kallarackal, "FPGA-Based Cost-Effective and Resource Optimized Solution of Predictive Direct Current Control for Power Converters," *Energies*, vol. 14, no. 22, p. 7669, 2021, https://doi.org/10.3390/en14227669.

[28] T. Technologies, *Terasic - DE Boards - Cyclone - DE10-Standard Development and Education Kit.* [Online]. Available: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=1081 (accessed: Jun. 3 2025).

[29] Intel, *Altera® FPGA Development Tools Design: FPGA Software, Design.* https://www.intel.com/content/www/us/en/products/details/fpga/development-tools.html (accessed: Jun. 4 2025).