International Journal of Electrical and Computer Engineering (IJECE)
Vol. 12, No. 4, August 2022, pp. 3530~3539
ISSN: 2088-8708, DOI: 10.11591/ijece.v12i4.pp3530-3539

A new method for self-organized dynamic delay loop associated pipeline with reconfigurable computing system

Nandigam Suresh Kumar[1], Dasari Vijaya Lakshmi[2], Bejjanki Pooja Sree Prasanna[2], Dodda Venkata Rama Koti Reddy[3]
[1]Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be) University, Visakhapatnam, India
[2]Department of Computer Science and Engineering, GITAM School of Technology, Hyderabad, India
[3]Department of Instrument Technology, College of Engineering, Andhra University, Vishakhapatnam, India

Article Info

Article history:

Received Jul 25, 2021
Revised Mar 17, 2022
Accepted Mar 30, 2022

Keywords:

Dynamic delay
Parallel processing
Pipeline
Propagation delay
Reconfigurable computing-system
Self-organized map

ABSTRACT

The minimization of propagation delay between pipeline stages is very important in wave propagation through pipeline-stages. The propagation delay can be minimized by minimizing the number of stages in a pipeline. In the proposed design a dynamic stage control is imparted in the pipeline. The propagation delay can be optimized in any type of pipeline by controlling number of stages dynamically. The pipeline interpretation helps a lot to overcome the flaws due to not ready sequence (NRS) and synchronization problems. It is observed that, in the pipeline design the basic and actively involved pipeline techniques are concerned with different challenges like clock, throughput, cell area, and sizes. As the data throughput increases the number of stages in pipeline also needs to be increased to meet the desired goal. In the case of unpredictable data speed, the definite number of pipeline stages creates severe problems. In this work a dynamic pipeline is integrated where the number of stages is dynamically changing depending up on data speed. In dynamic pipeline technique the circuit cell area of reconfigurable computing system (RCS) will be reduced dynamically at low-speed data transmission. In the high-speed data communication, the data speed is managed and controlled by dynamic delay loops.

Corresponding Author:

Nandigam Suresh Kumar
Department of Computer Science Engineering, GITAM Institute of Technology, GITAM (Deemed to be) University
Gandhi Nagar, Rushikonda, Visakhapatnam, Andhra Pradesh 530045, India
Email: nskgitam2009@gmail.com

1. INTRODUCTION

With the advent of field programmable gate array (FPGA) it became very easy provides provision to design and develop large circuits and systems on single board. In this paper the required components for the proposed method and the other existing techniques are analyzed, tested, and compared on FPGA Sparton 3E board. The main advantage with FPGA is complex architectures can be easily designed, emulated, and tested for different real time applications. In the present paper the concrete designs of registers, interrupt logic, pipeline, and counters are implemented with Xilinx FPGA.

Flexibility, malleability, training, and adaptability are the characteristics of the present research work. This environment motivated to design dynamic delay loops, which are proportional to the input data rate. In the present system high speed data rates are obtained from real time environment such as digital tachometer which is interfaced with electrical motor speed measurement system. The speed of rotor is measured with wireless sensor (such as tachometer) in terms of standard parameter rotation per minute (RPM).

When the systems are interfaced with sophisticated sensors, the design realizes the importance of trained system for enhancing the count of delay loops and pipeline stages dynamically rather than programming elements [1]. The sensors are sophisticated in measuring wide range of readings. The processing elements in FPGA make feasible to develop prototype of the proposed system.

The trained systems require artificial neural networks (ANN) models [2]. The models like Kohenen self-organizing maps use very simple neuron mode. The interconnections have individual weight. Each weight indicates certain input received from measure end in the current proposed system. In the present self-organizing model, the weights are modified with-out stating any output. Self-organized map (SOM) is method also called as dimensionality reduction. The SOM transforms a multidimensional input space having multiple set of various input samples into low dimensional map [3]. Here in the present work the network is designed to find its own solution in-terms of number of delay loops automatically. The input contains sensor data and the number of stages of pipelines and their inherited loops.

The delay loops are present at individual pipeline stages which are acting intermediate elements between processor and measurement element. The number of stages and number of dynamic loops are detected, and comparisons are done with different methods. The propagation delay time in a pipeline is minimized with proposed clock scheme. The propagation delay time is relatively reduced while number of stages is increased. It is observed that minimum numbers of loops are activated in proposed dynamic method and so cell area and processing time is less. In the present work, the RPM measurement accuracy increased with direct memory access (DMA) controller with dynamic pipeline (DMADP) method. The consequences due to the design problem can be minimized with DMADP method with reconfigurable computing system (RCS). The novelty in dynamically controlled number of stages in pipeline is adjusted and implemented with the help of hardware circuit RCS. In the present work dynamic delay loops are realized on FPGA Sparton 3E family electronic board for controlling number of stages. The dynamic loops provide variable delay between pipeline stages. The propagation delay can be controlled by this dynamic delay loops. The pipelines and other logics are developed in Verilog code.

2. EXISTING TECHNIQUES

The work presents a new wide-range speed measurement method, using the DMADP. As the DMA method is superior to other methods [4], finally the dynamic pipeline is used with DMA method. In this process, a new dynamic pipeline clock scheme is proposed to minimize the propagation delay time through digital measurement system. The new clock scheme is applied to pipeline by applying different input frequencies. An optimized propagation delay time is obtained with new clock scheme, when compared with existing techniques [5]–[7].

In the present research work, first the new dynamic pipeline system is tested with wave-pipeline [8], [9], and mesochronous pipeline alone [10], [11]. Secondly the new method DMADP is implemented on time based method and pulse count method along with dynamic pipeline. As the input device frequency increases, the number of pipeline delay loops dynamically increases and sets or adjusts proportional delay for not ready sequence (NRS) sequence of a DMA processor. The number of loops needs to be increased dynamically and hence a RCS is used, where this is not possible with fixed embedded system design. RCS is more intelligent because it is dynamically controllable and manageable [12]–[15]. RCS also reduces system design complexity, when comparing with synchronous frequency measurement system [6], [15]–[19]. In the present work, the increase in the input frequency of the encoder observed in terms of variations in dynamic loops invoked. It is found that DMADP counter is accurate in activating loops than time based and other Pipeline method.

To study the present method a DC motor is taken into consideration for testing. The new method can also be adoptable to different applications like fiber optics [20], [21], signal processing, and computer architecture [22], and linear model frequency measurement [23]. The prototype of DC motor with Tachometer is interfaced with FPGA board [24]. The prototype is designed to test the speed range between 500 rpm to 3,000 rpm. At different rpm the number of dynamic loops activated is observed in NRS of DMA. Accurate numbers of loops are generated in DMADP method when compared with other methods.

3. PROPOSED METHOD

A DC motor is setup with tachometer. The non-contact disc type tachometer is constructed with 32 slots as shown in Figure 1. The infrared receiver (IR) sensor is place as shown in Figure 2 to detect the white and black slots on tachometer disc. The snapshots of the FPGA board interface with IR sensor are shown in Figures 1 and 2. The IR sensor (optical) output is attenuated with the help of comparator as shown in

Figure 3. The detailed functional diagram of proposed method is presented in Figure 4. In Figure 3 the inherent individual functional units are represented. Each functional unit considered as a node in the neural network. In the proposed method a self-organized neural network plays key role in determining dynamic loops after pre-defined measurement. The system is trained in such a way that the output of the comparator is fed to input pins of Sparton 3E with the help of J-Tag interface which is shown in it can produce or provoke more accurate loops in proportional to the pulses generated as outcome of tachometer. The self-organized neural network representation of proposed system is shown in Figure 5. The Sparton 3E electron board includes for dynamic pipeline and counter. The dynamic pipeline, counter and other components are discussed in the rest of this paper. When high speed devices are interfaced, the dynamic loops are activated appropriately in the proposed DMADP method. Where, this is not happening in simple wave pipeline, mesochronous and any other method because of the difficulty in selecting delay element at real time. As the frequency of the measurement device increases the pre-defined individual loops get cascaded with the previous stages of the pipeline. This is how, the number of stages including pre-defined multiples of loops.

Figure 1. Snapshot of tachometer

Figure 2. The Board I/O pins are interfaced with IR sensor module with the help of J-Tag

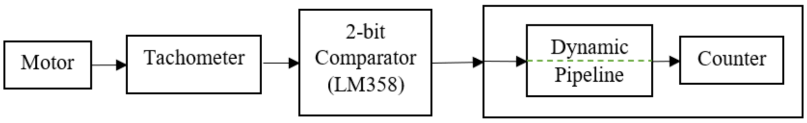


Figure 3. Block diagram of proposed method

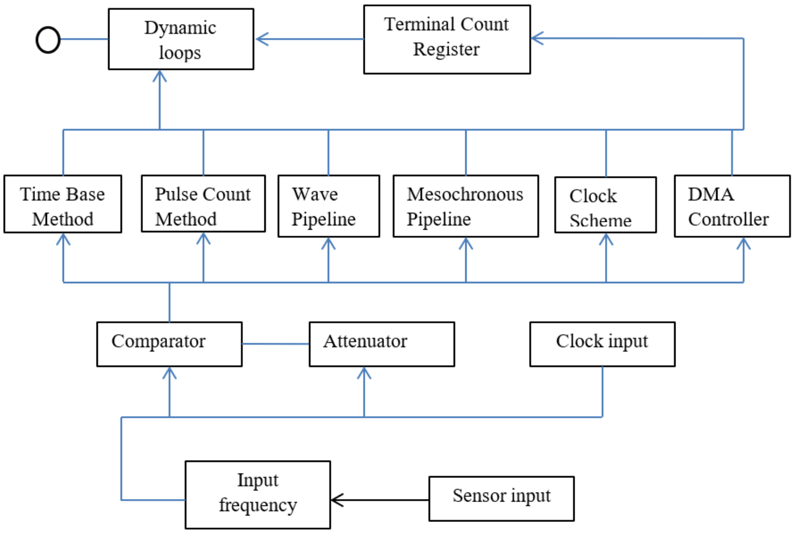


Figure 4. Functional diagram of proposed method

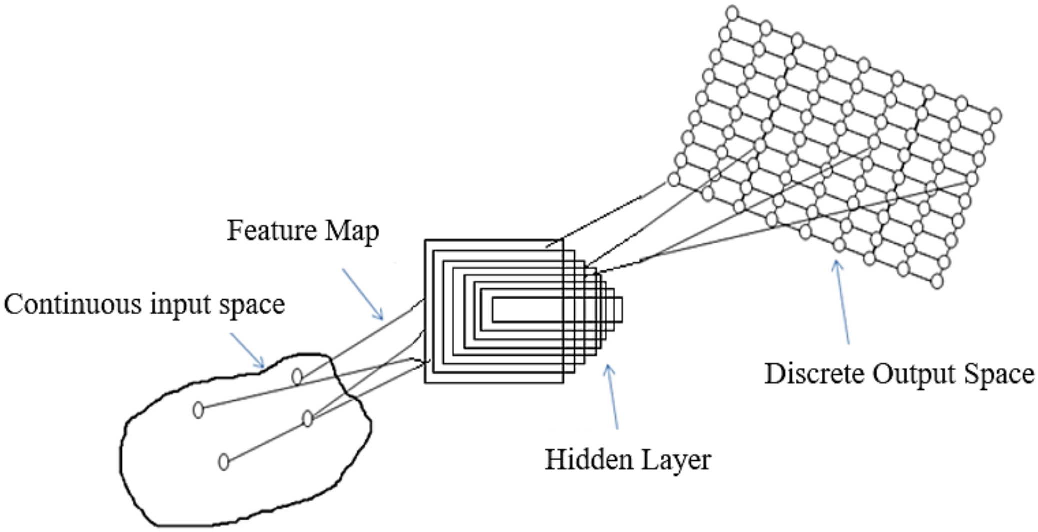


Figure 5. Self-organized neural network

3.1. Self-organized network

The present system contains arbitrary dimensions of input system. The input dimension is framed between sensor data, and number of pipeline stages and their loops dedicated for delay creation. Hence it is very important process of mapping extensive pattern space into feature space. The transformation of arbitrary incoming signal pattern into multi-dimensional pattern i.e., one-or two-dimensional discrete map is done in topological ordered fashion. In the present work during competitive learning, neurons are tuned to take the input signal up to some extent (set-point) of frequency. The winning neurons are mapped in appropriate number of dynamic loop system with respective to the input features. The winning neuron in the hidden layer implements the logic to check the number of pulses and compresses the number of pipeline stages and delay loops. Therefore, it is said that the self-organized map follows the topographic map in association with input frequency.

The Figure 4 depicts the transformation of high dimensional data space into low-dimensional discrete output space. The self-organizing map (SOM) is framed in single forwarded structure. In the output low dimensional layer every node is associated in respect to the input nodes. The low-dimensional layer is arranged in rows and columns.

In the present work single computational layer with rows and columns are arranged as shown in Figure 5. It is observed that, one dimensional or low dimensional layer map is produced with the present architecture. Each neuron in output space has direct connection with the input space through hidden layers. The output layer is a computational layer computes number of loops need to be connected in response to the input frequency at each pipeline stage. In the present output layer, there are eight loops considered across each stage of the pipeline and there will be total 8 stages framed in concern with the increase in the input data frequency. In the output space each node represents a loop.

Each node in the output population is activated with respective to the input signal frequency measurement and their readings and corresponding loops are presented in Table 1 to Table 3. The neighboring node is the low-dimensional output map is dynamically selected at the appropriate and every pre-requisite reckoning of rpm on the tachometer disk.

3.2. Merits and demerits of proposed system

Merits of proposed system are as follows: i) dynamic pipeline is not possible in embedded system hardware, ii) RCS devices are more intelligent, iii) supports self-synchronization [25], iv) modular programming is easy, v) new versions can be easily implemented without hardware up gradation, vi) the clock distribution is easier with internal logic gates, vii) it is very easy to write code in Verilog, viii) dynamic pipeline can be achieved in RCS is easy [26], ix) higher performance can be achieved, x) the proposed pipeline scheme avoids wave collision and minimizes the data propagation delay, and xi) in the proposed system the clock speed mainly affected by interrupt-based clock scheme. It avoids predicting the delay elements in the clock path. While the demerits of the proposed system are as follows: i) cost effective and ii) mostly applicable when high performance is required.

Table 1. RPM in time based method

S NO	Actual RPM	RPM in traditional time based	RPM in traditional pulse count method	Loops invoked in time based DDL method
1	750	750	750	750
2	870	870	870	870
3	960	960	960	960
4	1020	1005	1005	1020
5	1080	1006	1006	1080
6	1140	1006	1006	1140
7	1260	1006	1006	1260
8	1320	1006	1006	1320
9	1380	1006	1006	1380
10	1770	1006	1006	1770
11	2160	1006	1006	2160
12	2460	1006	1006	2460
13	2580	1006	1006	2580

Table 2. RPM in mesochronous dynamic loops

Switch No	RPM	Loops ($T_{measured\_loops}$)	Additional loops invoked per sec (-ve) (T_{error_loops})
1	1080	10	0
2	1170	16	5.25
3	1260	22	10.5
4	1380	29	18.5
5	1470	36	22.75
6	1560	42	28
7	1620	46	31.5

Table 3. RPM in wave pipeline with dynamic loops

S No	RPM	Loops ($T_{measured\_loops}$)	Error in loops (T_{error_loops})
1	1110	24	3.5
2	1170	34	8.5
3	1260	44	21.5
4	1350	56	31.5
5	1470	72	45.5
6	1560	84	56
7	1590	88	59.5

4. EXPERIMENTAL SETUP

4.1. Traditional time base and pulse count methods

The time base method (TBM) [27], [28] and pulse count method (PCM) [29], [30] are two methods which are two basic methods used in all standards. These methods are first implemented with dynamic pipeline and then tested with DMADP. In these traditional methods it is producing exact RPM till the processor set point reaches in the Process element. When it reaches the set point it is unable to read the next pulse. If the i/p speed increases than the processor speed limit (set point) then the processor unable to follow the input data. In this case processor produces a constant reading in all the cases.

In the present case the processor speed is set to measure 1,000 rpm. When the RPM increases than 1,000 rpm the processor is unable to read next pulse from tachometer. The measurement readings are shown in Table 1. A pipeline with dynamic pipeline is cascaded with time base and pulse count method to synchronize the speed between high-speed device and slow speed processor. As the speed increases beyond the processor speed the pipeline stages and dynamic loops increases with respect to the additional pulses generated. The number of stages in the pipeline increases after predefined number of loops is reached in the process.

4.2. Time base methods with dynamic pipeline

Unlike the traditional method, as the number of pulses go beyond the set point it automatically increases the number loops and further it increases the pipeline stages with respect to the number of loops. Time base method with dynamic pipeline is not much suitable because it is activating one loop for each RPM. So, the time complexity is high here. In this method it is assumed as 1 RPM is equal to 64 pulses. The readings at different frequencies are shown in Table 1.

1 rpm=1 dynamic loop

Hence, after every completion of 64 loops only the system will activate one loop. This shows that the accuracy of pulse measurement is poor in both time base and pulse count with dynamic delay loop (DDL) implementation. Therefore, there is no use of any DDL in time base and pulse count method. But, the performance can be improved with the help of integrating dynamic pipeline.

To compensate the clock delay in mesochronous pipeline, delay loops are produced to carry extra pulses. Here 1loop=8 RPM. Like traditional method the DMA counter is set to count 1,000 rpm. Beyond this it will activate the dynamic loops to synchronize the tachometer pulses and counter speed. Due to the clock delay problem the pipeline is unable to invoke exact number of delay loops. The following Table 2 is showing error in terms of extra delay loops required. The first column represents the switch number used to control motor speed. With the combination of multiple numbers of switches the speed can be controlled between wide ranges up to 3,000 rpm.

In this technique there is no requirement of activating dynamic loops. After the set point reached in DMA counter, for instance at 1,080 rpm the rpm needs to measure beyond 1,000 rpm is 80 rpm. To compensate the extra rpm the dynamic loops are activated after each 8 rpm. 8 rpm is considered to identify prominent errors in the loops. Hence, 80/8 loops will be activated. In this case the error in loop measurement is zero. While the speed increases, the pulse rate also increases, and number of loops activated may not reflect in the same ratio. The total number of loops (Tloops) counted as,

4.3. DMA with mesochronous dynamic pipelined pulse count method

The set-point (SP) of the rpm is set in the DMA counter as 1,000 rpm, MV represents the measured value, and N_{rpm_1loop} is equal to the number of rpm considered for 1 loop. For suppose at switch 5, the measured rpm value (MV) is 1,470. The difference between SP and MV is now 470. Then the actual loops need to be activated must be equal to,

$$T_{actual\_loops}=470/8=58.75 \text{ loops}$$

In the above calculation the fraction part 0.75 loops indicated 6 rpm. The number of loops measured ($T_{measured\_loops}$) are 36 loops. Hence the error in loops (T_{error_loops}) are calculated by,

$$T_{error\_loops} = T_{actual\_loops} - T_{measured\_loops}$$

i.e., 58.75-36=22.75 loops

4.4. DMA with dynamic wave pipelined pulse count method

In wave pipelined pulse count method the clock delay is compensated by distributing the delay loops equally to the maximum propagation delay and minimum propagation delay path. This also leads confusion in providing exact delay loops to compensate clock delay. In this method 4 rpm is considered as 1 loop. The following Table 3 is showing error in terms of extra delay loops required in. The RTL schematic of the existing model and dynamic pipeline are shown in Figures 6 and 7 respectively.

4.5. DMA with dynamic pipelined pulse count method

The pulses are measured with the help of a counter. These counting of pulses will start from time t0 to until 1 sec sampling time duration is completed. Here the device is set to measure 1,020 rpm. So that in 1sec sampling time, 17 rpm can be accessed through the device. So here an accuracy level of 0.05882% can be achieved in phase change material (PCM).

Once this 1 sec sampling period is completed a CLER signal will be generated to clear the count in the counter. Once the counter receives the Clear interrupt it latches the count in a buffer. For instance, if the buffer value is 32 and so, it represents one rotation as one rotation is equals to 32 pulses. If it further multiplied with a factor 60 it indicates speed for 1 minute. That is nothing but RPM.

In DMADP pulse count method the counter is set to count 17 pulses per second. When the count crosses 17 pulses the dynamic loops and number of pipeline stages will be activated automatically. The additional dynamic loops activated are shown in in the Table 4. For suppose at switch 6, the actual number of loops are calculated as,

$$T_{actual\_loops}=560/4=140$$

$$T_{error\_loops}=140-84=56$$

For example, when RPM is 1,080, it produces 18 pulses per sec. It means it is producing 1 excess pulse at each second to satisfy the total count. Although the counter is programmed to count 17 pulses, the counter is able to count 18 pulses with the help of dynamic loops at each pipeline. Here the readings of RPM, in terms of dynamic loops and measured RPM are shown in Table 4. As the count increases the delay loops at pipeline also increases dynamically. These dynamic loops will compensate the extra pulses generate at the tachometer. These dynamic loops will carry the extra pulse to the DMA counter. After certain count the DMA will clear the count in its counter. During CLEAR operation the DMA enter NRS state. The pulses during NRS state were also hold in pipeline loops.

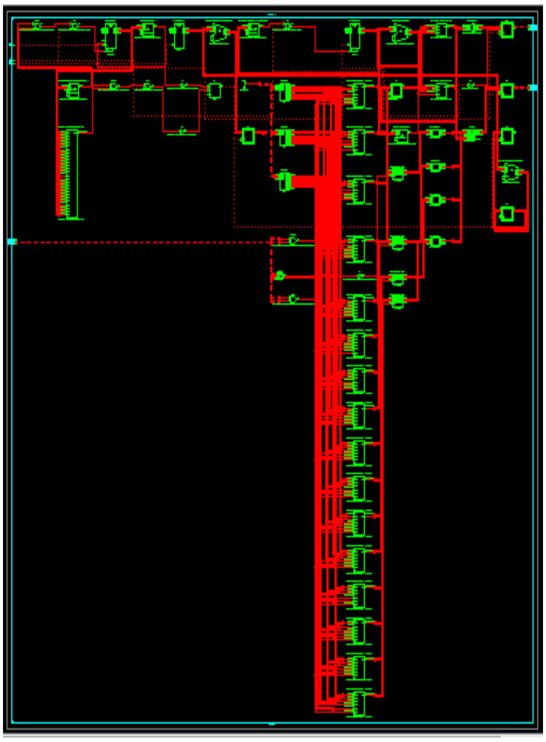


Figure 6. Layout of dynamic mesochronous pipeline

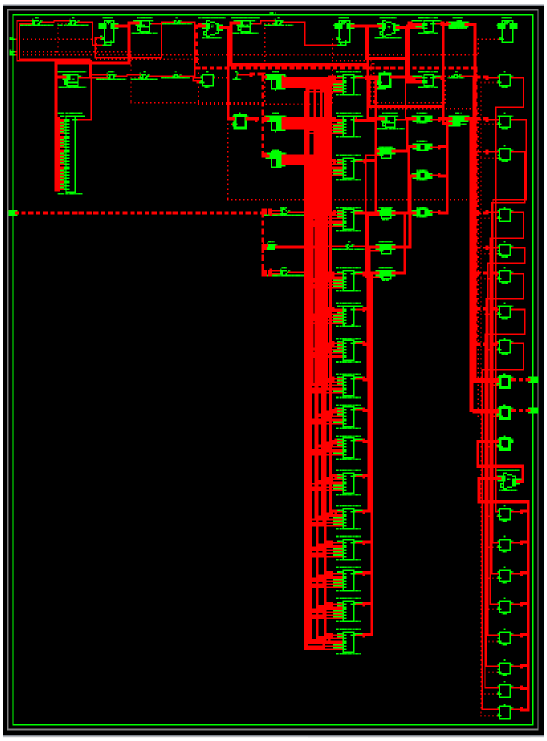


Figure 7. Layout of pulse count method with dynamic pipeline

Table 4. RPM DMADP with pulse count method

S No	RPM	Loops Invoked in DMADP Method ($T_{measured\_loops}$)	Additional loops invoked per sec ($T_{additional\_loops}$)
1	1080	18	1
2	1140	19	2
3	1260	21	4
4	1320	22	5
5	1410	23	6
6	1560	26	9
7	1650	27	10
8	1770	29	12
9	1890	31	14
10	1950	32	15
11	2070	34	17
12	2190	36	19
13	2220	37	20
14	2340	39	22

The CLEAR signal took almost 20 ms to clear the counter, which is producing a not ready sequence in counter. So, during this 20 msec the counter may not receive the next input data. If the width of the NRS due to CLEAR signal could be reduced, then NRS can be minimized. In this state the counter undergoes to not ready state (NRS). As the input data is coming at 960 microsec, the 20 msec NRS can create serious problem, because per one rotation it takes a delay time of (32×960 µsec) 30 msec (approx.). In embedded systems and anorectal malformation (ARM) processors this NRS value is fixed, but this can be manageable in reconfigurable computing. High tenacity, high accuracy, short detecting time, minimum relative error, and wide range of measurement are the characteristics of proposed system. Comparative results with different pipelines are shown in Figure 8.

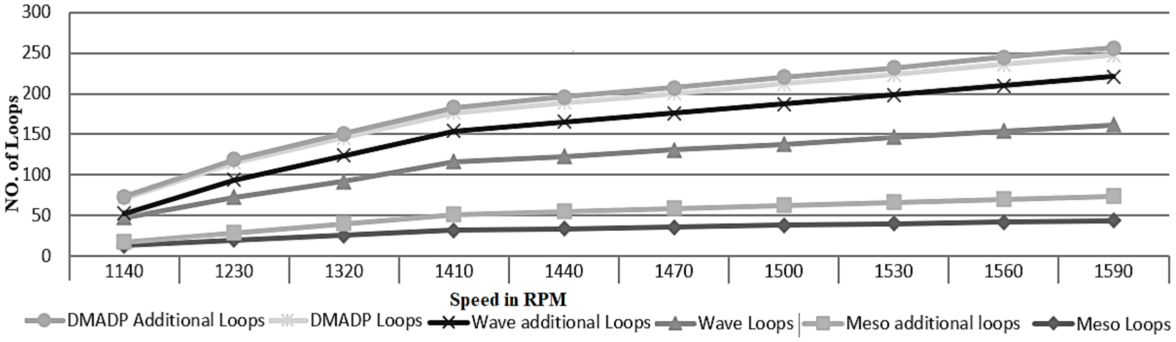


Figure 8. Analysis of loops and additive loops executed at different types of pipelines

5. CONCLUSION

Reconfigurable computing system (RCS) is used to control and generate the clock signals. RCS is used to produce dynamic pipeline and loops. When measure end frequency is very high than the DMA counter, the pipeline will take part to synchronize the speed between DMA and measure end and vice versa. The pipeline stages are dynamically increasable with FPGA programming. The propagation delay is minimized with new clock scheme. The accuracy in the frequency measurement is improved with the help of dynamic loops when compare with other measurements. The proposed system is able to fetch input data in Not Ready Sequence and parameters like loops, pipeline stages, counters, and pipelines are all dynamically controllable. The proposed pipeline enables exact number of delay loops as per the input data rate.

REFERENCES

[1] M. A. Arbib, “Schemas and neural networks for sixth generation computing invited survey,” *Journal of Parallel and Distributed Computing*, vol. 6, no. 2, pp. 185–216, Apr. 1989, doi: 10.1016/0743-7315(89)90059-2.

[2] J. Hertz, A. Krogh, R. G. Palmer, and H. Horner, “Introduction to the theory of neural computation,” *Physics Today*, vol. 44, no. 12, Dec. 1991, doi: 10.1063/1.2810360.

[3] K. M. Patwardhan, “Making sense of a service-oriented architecture (SOA) governance framework in the digital ecosystem utilizing a sociotechnical approach,” in *2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)*, Nov. 2019, pp. 49–54, doi: 10.1109/SOCA.2019.00015.

[4] N. S. Kumar and D. V. R. KotiReddy, “A new method to enhance performance of digital frequency measurement and minimize the clock skew,” *IEEE Sensors Journal*, vol. 11, no. 10, pp. 2421–2425, Oct. 2011, doi: 10.1109/JSEN.2011.2119371.

[5] N. S. Kumar and D V R. K. Reddy, “Optimizing technique to improve the speed of data throughput through pipeline,” in *Advances in Intelligent Systems and Computing*, Springer India, 2016, pp. 237–247.

[6] G. N. Hoyer, G. Yee, and C. Sechen, “Locally clocked pipelines and dynamic logic,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 1, pp. 58–62, Feb. 2002, doi: 10.1109/92.988731.

[7] Y. Yong-Sheng, D. Gao-Ming, and S. Yu-Kun, “Study on the multi-pipeline reconfigurable computing system,” in *2008 International Conference on Computer Science and Software Engineering*, 2008, pp. 122–125, doi: 10.1109/CSSE.2008.1068.

[8] M. Anbuselvi, A. S. Salivahanan, and P. Saravanan, “Analysis of WavePipelined architecture of ARALDPC codes,” *International Journal of Computer Applications*, vol. 24, no. 3, pp. 43–47, Jun. 2011, doi: 10.5120/2927-3872.

[9] J. Nyathi, J. G. Delgado-Frias, and J. Lowe, “A high performance, hybrid wave-pipelined linear feedback shift register with skew tolerant clocks,” in *2003 46th Midwest Symposium on Circuits and Systems*, vol. 3, pp. 1384–1387, doi: 10.1109/MWSCAS.2003.1562553.

[10] S. B. Tatapudi and J. G. Delgado-Frias, “A mesochronous pipelining scheme for high-performance digital systems,” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 53, no. 5, pp. 1078–1088, May 2006, doi: 10.1109/TCSI.2006.870221.

[11] D. E. Duarte, N. Vijaykrishnan, and M. J. Irwin, “A clock power model to evaluate impact of architectural and technology optimizations,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 6, pp. 844–855, Dec. 2002, doi: 10.1109/TVLSI.2002.808433.

[12] F. Grassert and D. Timmermann, “Dynamic single phase logic with self-timed stages for power reduction in pipeline circuit designs,” in *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196)*, vol. 4, pp. 144–147, doi: 10.1109/ISCAS.2001.922191.

[13] W. Jin Kim and Y.-B. Kim, “Clocking for correct functionality on wave pipelined circuits,” in *IEEE International [Systems-on-Chip] SOC Conference, 2003. Proceedings.*, pp. 161–164, doi: 10.1109/SOC.2003.1241484.

[14] G. Scutari, S. Barbarossa, and L. Pescosolido, “Distributed decision through self-synchronizing sensor networks in the presence of propagation delays and nonreciprocal channels,” in *2007 IEEE 8th Workshop on Signal Processing Advances in Wireless*

Communications, Jun. 2007, pp. 1–5, doi: 10.1109/SPAWC.2007.4401363.

[15] M. Singh and S. M. Nowick, "The design of high-performance dynamic asynchronous pipelines: high-capacity style," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 11, pp. 1270–1283, Nov. 2007, doi: 10.1109/TVLSI.2007.902206.

[16] S. Rong Kuang, J. Min Jou, R. Der Chen, and Y. Horng Shiau, "Dynamic pipeline design of an adaptive binary arithmetic coder," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 9, pp. 813–825, 2001, doi: 10.1109/82.964994.

[17] M. M. Bassiri and H. S. Shahhoseini, "On-line HW/SW partitioning and co-scheduling in reconfigurable computing systems," in *2nd IEEE Int. Conf. on Computer Science and Information Technology*, 2009, pp. 557–562, doi: 10.1109/ICCSIT.2009.5234664.

[18] K. Chae and S. Mukhopadhyay, "A dynamic timing error prevention technique in pipelines with time borrowing and clock stretching," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 1, pp. 74–83, Jan. 2014, doi: 10.1109/TCSI.2013.2268272.

[19] T. Li, Y. Ren, D. Yu, and S. Jin, "RAMSYS: resource-aware asynchronous data transfer with multicore systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 5, pp. 1430–1444, May 2017, doi: 10.1109/TPDS.2016.2619344.

[20] N. Suresh, R. Sridevi, D. D.V., and V. Sridevi, "Studies and a method to minimize and control the jitter in optical based communication system," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 4, 2013, doi: 10.14569/IJACSA.2013.040408.

[21] D. Maraval, R. Gabet, Y. Jaouen, and V. Lamour, "Dynamic optical fiber sensing with brillouin optical time domain reflectometry: application to pipeline vibration monitoring," *Journal of Lightwave Technology*, vol. 35, no. 16, pp. 3296–3302, Aug. 2017, doi: 10.1109/JLT.2016.2614835.

[22] N. Suresh Kumar and D. V Rama Koti Reddy, "Effect of interrupt logic on delay balancing circuit," *International Journal of Computer Applications*, vol. 27, no. 4, pp. 26–30, Aug. 2011, doi: 10.5120/3288-4479.

[23] C. Tan, Z. Yue, and J. Wang, "Frequency measurement approach based on linear model for increasing low SNR sinusoidal signal frequency measurement precision," *IET Science, Measurement and Technology*, vol. 13, no. 9, pp. 1268–1276, Nov. 2019, doi: 10.1049/iet-smt.2018.5643.

[24] C. K. Midhun, J. Joy, and R. K. Kavitha, "High-speed dynamic asynchronous pipeline: self-precharging style," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 10, pp. 2233–2237, Oct. 2014, doi: 10.1109/TVLSI.2013.2282834.

[25] M. Campusano, K. Jensen, and U. P. Schultz, "Towards a service-oriented u-space architecture for autonomous drone operations," in *2021 IEEE/ACM 3rd International Workshop on Robotics Software Engineering (RoSE)*, Jun. 2021, pp. 63–66, doi: 10.1109/RoSE52553.2021.00017.

[26] I. Mavroidis *et al.*, "ECOSCALE: reconfigurable computing and runtime system for future exascale systems," in *Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2016, pp. 696–701, doi: 10.3850/9783981537079_1021.

[27] C. D. DiCenzo, B. Szabados, and N. K. Sinha, "Digital measurement of angular velocity for instrumentation and control," *IEEE Transactions on Industrial Electronics and Control Instrumentation*, vol. IECI-23, no. 1, pp. 83–86, Feb. 1976, doi: 10.1109/TIECI.1976.351354.

[28] D. W. Huber, O. P. Malik, and G. S. Hope, "A digital device to measure angular speed and torque angle," *IEEE Trans. on Industrial Electronics and Control Instrum.*, vol. IECI-22, no. 2, pp. 186–188, May 1975, doi: 10.1109/TIECI.1975.351250.

[29] T. J. Maloney and F. L. Alvarado, "A digital method for DC motor speed control," *IEEE Transactions on Industrial Electronics and Control Instrumentation*, vol. IECI-23, no. 1, pp. 44–46, Feb. 1976, doi: 10.1109/TIECI.1976.351346.

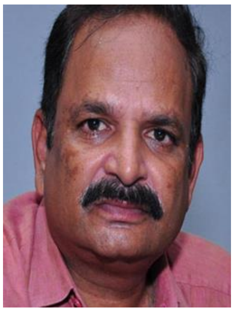
[30] S.-C. Lin and S.-J. Tsai, "A microprocessor-based incremental servo system with variable structure," *IEEE Transactions on Industrial Electronics*, vol. IE-31, no. 4, pp. 313–316, Nov. 1984, doi: 10.1109/TIE.1984.350083.

BIOGRAPHIES OF AUTHORS

Nandigam Suresh Kumar is currently working with the department of Computer Science and Engineering, GITAM Institute of Technology, Visakhapatnam. He has more than 17 years of academic experience. His research interests are machine learning, internet of things, and parallel processing. He can be contacted at email: snandiga@gitam.edu. https://vspgitcse.gitam.edu/faculty/profile/1922

Dasari Vijaya Lakshmi is currently working with the Department of Computer Science and Engineering, GITAM School of Technology, Hyderabad. She has more than 10 years of academic experience. Her research interests are machine learning, internet of things, and Adhoc networks, machine learning, virtual reality, and cyber security. She can be contacted at email: dasarivijayalakshmi@gmail.com. https://hydgstcse.gitam.edu/faculty/profile/10218

Bejjanki Pooja Sree Prasanna is currently working with the Department of Computer Science and Engineering, GITAM School of Technology, Hyderabad, India. She has more than 5 years of academic experience. Her research interests are machine learning, internet of things, and Adhoc networks. She can be contacted at email: poojasr407@gmail.com. https://hydgstcse.gitam.edu/faculty/profile/10390



Dodda Venkata Rama Koti Reddy is currently working with the department of Instrument Technology, Andhra University, Visakhapatnam. He has more than 30 years of academic experience. His research interests are machine learning, internet of things, and Adhoc networks. sensors and sensor networking, micro-electromechanical systems (MEMS) and nano technology. He can be contacted at email: rkreddy_67@yahoo.co.in. https://www.andhrauniversity.edu.in/college/college-of-engineering/engineering-departments/instrumenttechnology.html