International Journal of Research in Engineering, Science and Management
Volume 5, Issue 6, June 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

High Performance on Truncated MAC Units of Digital Filtering in the Residue Number

P. Chandana[1], Babitha S. Ullal[2], B. L. Gagana[3*], R. Mallikarjun[4], S. Malola[5]
[1]*Professor, Department of Electronics and Communication Engineering, East West Institute of Technology, Bangalore, India*
[2,3,4,5]*Student, Department of Electronics and Communication Engineering, East West Institute of Technology, Bangalore, India*

***Abstract*: The Residue Number System is used to create a finite impulse response filter for this project (RNS). The benefits of using the selected moduli set are the same as using the shift and add method. The developed filter architecture is compared to a traditional version of a reprogrammable Residue Number System FIR filter. The filters and the RTL diagram for the Field programmable gate array execution were both produced using the Vivado Design Suite 2018.1 platform (FPGA). The area (A), delay (T), and power (P) of the filters are all examined (P). The trial discoveries uncover that the proposed method has better implementation as far as deferral, region postpone item, region, and energy power item. The proposed methodology is also functionally proven on the Artix-7 Field Programmable Gate Array to evaluate digital signal processing builder.**

***Keywords*: Reconfigurable FIR filter, Redundant-RNS, Parallel Prefix Adder, QRNS, FPGA.**

1. Introduction

The output voltage is one of the first restrictions in the design of time-ahead Application Specific Integrated Circuits (ASIC). Low power consumption is being progressed in order to increase the ASIC's flexibility while bringing down the gadget's value, intricacy, and mass. One reason for output power in modern ASICs is because the Digital Signal Process restricts unit of measurement diversity. The residue information type has been recommended as a power-saving alternative to the usual 2's complement selection line in DSP applications. It is possible to reduce power usage by using FIR filters inside the

RNS rather than the two's complement range illustration system (TCS). In his Arithmetic Classic of Sun Tzu, the Chinese scholar Sun Tzu employed the residue number illustration approach for the first time in the III century AD. Filters in the infrared range are simple DSP elements. Multiplication is a crucial fundamental skill. Multiplication is a very crucial elementary act in arithmetic operations. Multiply and Accumulate (MAC) and real number unit of measurement are unit Multiplication-based operations currently enforced in convolution, fast Fourier Transform (FFT), filtering, and in microprocessors in their arithmetic and logic unit, as well as much of Digital Signal Processing (DSP) applications. The execution time of most DSP algorithms is dominated by the need for high-speed number multiplication. Recently, the instruction cycle time of a DSP chip's multiplication time is the most important factor. The interest for rapid methods has been rising as a result of the rapid growth of portable computer and signal method applications. Higher output arithmetic operations are required to enjoy the given performance over a long period of time for signal and picture methods. Multiplication is one of the most important number-crunching activities in such applications, and the occurrence of a fast number loop has been a topic of discussion for decades. Decreased time postponement and power usage are unit key needs for some applications. This achievement demonstrates a variety of multiplier factor architectures. In VLSI modules, modulo multipliers are implemented with the help of Read-only memory, which is memoryless and look-up-table. Also, this concept was first used to a short period of time, but it will subsequently be applied to a longer period of time. , To increase the potency of memoryless multipliers, alternatives of residue number such as equal and standard are used, and to accelerate multipliers, a Booth-encoding algorithm rule is applied. A 2n-1 modulo multiplier factor based on the redundant residue number representation scheme has been chosen for a very high variance produced 2n-1 modulo multiplier factor (RRNS). With the utilization of the summary plan compiler apparatus, the standards of the 2n-1 modulo multiplier factor have been really examined and investigated for the latest modulo multiplier factor improvement. [15] The 2n-1 modulo multiplier factor, which depends on the Residue number portrayal framework, is generally utilized as a quicker and more reasonable math circle for different sign handling applications, including picture handling, limited motivation reaction (FIR) channels, correspondence, cryptography, separate geometrical capacity redesign, and different computerized signal handling applications (DSP). [16]

2. Residue Number System

The RNS numbers are unit delineate at intervals the variability of pairwise relatively prime numbers, known as moduli (β) = $\{m_1, \ldots, m_n\}$, and the GND(m_i,m_j) = 1 for i≠j. The multiplication of all RNS moduli is denoted by M.

$$M = \prod_{i=1}^{n} m_i$$

*Corresponding author: gaganabl2000@gmail.com

Where, m_i is termed the powerful vary of the system.

An integer $0 \leq X < M$ are often unambiguously described in RNS as a tuple $\{x_1, x_2, \ldots, x_n\}$

$$\text{where } x_i = |X|_{m_i}$$

The actively vary of the RNS is split into two roughly identical components positive numbers and negative numbers.

Any integer assuring one among the two relations shown below:

$$-\frac{M-1}{2} \leq X \leq \frac{M-1}{2}, \text{ for odd M}, \quad (1)$$

$$-\frac{M}{2} \leq X \leq \frac{M}{2} - 1, \text{ for even M}, \quad (2)$$

can be presented in Residue number system.

The arithmetic operations like addition, subtraction and multiplication operations in RNS are unit outlined by the formulas below:

$$A \pm B = (|a_1 \pm b_1|_{m_1} \ldots, |a_n \pm b_n|_{m_n}), \quad (3)$$

$$A \times B = (|a_1 \times b_1|_{m_1} \ldots, |a_n \times b_n|_{m_n}), \quad (4)$$

The number are often reconstructed by the tuple that relies Chinese remainder theorem (CRT),

$$X = \left|\sum_{i=1}^{n} \left|\left|M_i^{-1}\right|_{m_i} xi\right|_{m_i} M_i\right|_M, \quad (5)$$

where $M_i = \frac{M}{m_i}$. The term $|M_i^{-1}|_{m_i}$ means multiplicative inverse of M_i.

A. Chinese Remainder Theorem

Given a combine of considerably main moduli (m1, m2,...mn) and a residue illustration (r1,r2,...rn) there in order of some range X, i.e .ri= |X|mi, the quantity and its residues are unit connected by the equation:

$$|X|_M = \left|\sum_{i=1}^{n} ri |M_i^{-1}|_{mi} M_i\right| \quad (6)$$

Where M is that the product The standard reduction on the left-hand aspect are often skipped if the worth's concerned area restricted specified the ultimate value X is inside the dynamic vary.

Equation we have a tendency to revise X as:

$$X \triangleq \{r1, r2, \ldots, rn\} \triangleq \{r1, 0, \ldots 0\} + \{0, r2, \ldots, 0\} + \{0, 0, \ldots, rn\}$$
$$\triangleq X1 + X2 + \cdots + Xn \quad (7)$$

As a result, the reverse conversion procedure necessitates the invention of Xi's. Getting every Xi may be a reverse conversion procedure in and of itself. Getting X, on the opposite hand, is considerably harder.

Consider the case once we would like to induce Xi from 0,0....,ri,....,0,0. Aside from ri, all of Xi's residues are unit zeros. As a result, Xi should be a multiple of mj, where ji. As a result, Xi are often written as,

$$X_i \triangleq r1 * \{0, 0, \ldots 1, \ldots, 0, 0\} \triangleq ri * Xi \quad (8)$$

$$(r_i \times r_i^{-1}) \bmod m_i = 1 \quad (9)$$

We outline M_i as M/M_i, where $M = \prod_{i=1}^{k} pi$.

Then:

$$||M_i^{-1}|_{mi} M_i|_{mi} = 1 \quad (10)$$

After all m_i's are unit comparatively prime, the inverses manage:

$$X_i = |M_i^{-1}|_{mi} M_i \quad (11)$$

$$X_i = r_i X_i = r_i |M_i^{-1}|_{mi} M_i \quad (12)$$

$$X = \sum_{i=1}^{n} Xi = \sum_{i=1}^{n} r_i |M_i^{-1}|_{mi} M_i \quad (13)$$

Modulo reduction must be applied to both sides of the equation to guarantee that the ultimate result is within the powerful range. The term finite impulse response is used because the filter output is enumerated as a weighted finite term sum of past, present, and perhaps future values of the filter input.

$$y = \sum_{n=0}^{N} xn h_n$$

Where, y = output, x_n= input and h_n= coefficient.

Where y is the output, xn is the input, and hn is the coefficient.

A feed-forward difference equation is used to create a FIR filter. The phrase "feed-forward" refers to the fact that there is no response from previous or year after outputs in order to create the current output, only input-related words. Three main building blocks are required for the execution of a FIR. They really are

- multiplication
- addition
- signal delay

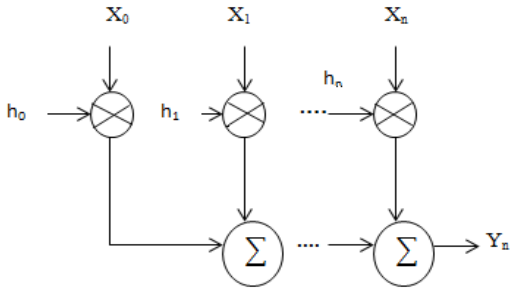

Fig. 1. Direct form of fir filter realization

The FIR Filter's direct form realisation is depicted in the diagram above. The coefficients of an order N FIR filter are N+1, and in general, N+1 multipliers and N two-input adders are required. The multiplier coefficients are the same as the transfer function coefficients in a direct form organisation. FIR filters have a fundamental drawback in that they take more processing to process a signal.

3. Related Work

A. Speed of Multiplication operation

Vedic mathematics is utilised as a low-power multiplier in Vedic mathematics. Design optimization at all stages is required to reduce power consumption in digital systems. This enhancement incorporates the innovation used to create computerized circuits, the circuit point and geography, the design used to execute the circuits, and, at a significant level, the calculations utilized. Advanced multipliers are the most generally involved parts in other computerized circuit plans. The elements that are used to complete any job quickly, in a linked manner, and depending on how the components are stacked, many types of multipliers are present. A positive multiplier design is put away contingent upon the application. In numerous DSP calculations, the multiplier is situated in the basic postpone way, and it is at last what decides the calculation's way of behaving.

The speed of augmentation activities is essential in both DSP and conventional processors. A progression of expansion, deduction, and shift tasks are utilized to play out the numerical cycle. In the writing, a plenty of additon calculations have been introduced, each with its own arrangement of advantages and hindrances concerning speed, circuit intricacy, region, and power utilization. The multiplier in a PC framework is a critical part. A multiplier with n pieces of goal has n doors, and the worth of the hardware included is relative to the square of the goal.

The two most significant variables for duplication calculations utilized in DSP applications as far as big time are dormancy and throughput. Inert is an estimation of how long contributions to a gadget are prepared before the eventual outcome is open on yields, as well as how long contributions to a gadget are steady before the end-product is available on yields. The multiplier isn't simply a huge defer block, yet it likewise consumes a ton of force; throughput is the quantity of duplications that should be possible in a given timeframe. Thus, limiting defer through different postpone enhancements is basic on the off chance that power utilization is likewise a worry.

B. Booth Multiplier

Numerous strong multipliers have been developed previously. Shen and Chen utilized the Booth way to deal with limit the incomplete item's exchanging action, yielding a low-power multiplier. Chen.et.al fostered a low-power 2's supplement multiplier. To decrease the incomplete item's exchanging action, the Radix-4 Booth calculation is applied. To make a decent width multiplier, Wang et al embraced a left-right methodology. It lessens the halfway item, which speeds up the cycle, however at the expense of a great deal of force. Zhijun and Ercegovac proposed a direct cluster multiplier that utilized various procedures. The structure saves energy however expands the region and time it takes to finish. Chen and Chu proposed a low-power fast multiplier utilizing false power concealment technique (SPST) on the Booth decoder and multiplier pressure tree, in spite of the way that these designs occupy a ton of room. Krad and Taie8 performed a execution investigation of two 32-cycle multipliers utilizing a convey look-ahead viper and a wave convey snake for incomplete item expansion. Dastjerdi et al. created BZ-FAD for shift-add multipliers. The multiplier's switching activities are reduced as a result of this configuration, which saves power. The structure consumes less energy and has a shorter critical path than the previous one. Mohanty and Tiwari introduced a modified probabilistic estimation bias (PEB) radix-4 Booth multiplier for DSP applications. In addition, the PEB multiplier is given an efficient adder. To give the optimum critical path latency, this adder uses only a few logic resources. Shao and Li developed an energy and region effective 16-digit fixed width Booth multiplier and squarer. A general model of cluster-based estimate number-crunching processing (AAAC) has been introduced to direct the multiplier and squarer plan. He et partners made a high velocity fixed-width Booth multiplier in view of the contingent likelihood of info series (CPIS). This arrangement decreases circuit above while speeding up the cycle. Shabbir.et.al fostered a low-power, high velocity multiplier utilizing the Dadda calculation. The multiplier is made to work with complete adders. To lessen yield messes up, the multipliers are developed with back-peddles.

C. Digital Multipliers

Digital multipliers are urgent parts in all computerized signal processors (DSPs), and the presentation of the DSP still up in the air by the speed of its multipliers. In advanced hardware, the exhibit duplication calculation and the Booth augmentation calculation are the two most normal augmentation calculations. The exhibit multiplier demands less investment to process on the grounds that the fractional items are determined autonomously in equal. The postponement related with the exhibit multiplier is the time it takes for signs to miss through the doors that make the duplication cluster.

Stall increase is another notable augmentation technique. Enormous corner clusters are expected for fast duplication and remarkable estimations, which requires huge incomplete total and fractional convey registers. The operands of two n-digit operands are increased. While utilizing a radix-4 stall recording multiplier, the most un-huge portion of the end result takes about n/(2m) clock cycles, where m is the quantity of Booth recorder snake stages.

Thus, there is an extensive engendering defer in this situation. Computerized multipliers have forever been a famous subject of exploration, with various novel duplication calculations delivered because of their significance in DSP.

In this undertaking, the Urdhva tiryakbhyam Sutra is applied to the parallel number framework first, and afterward used to foster a computerized multiplier design. This is demonstrated to be very near the ordinary exhibit multiplier engineering. The value of changing a NXN multiplier structure into a 4X4 multiplier structure is likewise exhibited in this Sutra. The Nikhilam Sutra is then checked out, and it is demonstrated to be significantly more productive in enormous number augmentation since it separates two huge numbers into two more modest ones.

The recommended increase strategy's processing

effectiveness is represented by diminishing a 4X4-bit duplication to a solitary 2X2-digit increase activity. This work gives an orderly plan interaction to a quick and region productive digit multiplier in light of Vedic science. The Multiplier Architecture is based on the Vertical and Crosswise technique of ancient Indian Vedic Mathematics.

D. Finite Impulse Response

The limited drive reaction (FIR) channel is the most regular sort of channel utilized in programming. This layout will assist you with figuring out them on a hypothetical as well as a down to earth level. Signal conditioners are channels. Every one works by taking an information signal, obstructing pre-determined recurrence parts, and afterward communicating the first sign short those parts to the result. For instance, a traditional telephone line goes about as a channel, limiting frequencies to a reach that is considerably more modest than the scope of frequencies that people can hear. That is the reason paying attention to CD-quality music on the telephone isn't so fulfilling as paying attention to it face to face.

A computerized channel is comprised of advanced parts that take an advanced information and result a computerized signal. In a regular computerized sifting application, an advanced sign processor (DSP) gathers input tests through A/D converter, plays out the numerical tasks suggested by hypothesis for the ideal channel type, and results the outcome by means of a D/A converter. A simple channel, then again, is made totally out of simple parts like resistors, capacitors, and inductors and works with simple data sources straightforwardly.

The most well-known channel types are lowpass, highpass, bandpass, and band stop. A lowpass channel can be utilized to sift through high frequencies by just permitting low recurrence signals (under a predetermined end recurrence) to go through to its result.

A lowpass channel, like a telephone line, is useful for restricting the highest scope of frequencies in a sound stream in this situation. Conversely, a high pass channel dismisses just recurrence parts under a given limit. as an illustration A high pass channel eliminates the undeniable 60Hz AC power "murmur," which might be gotten as clamour following essentially every sign in the United States.

- A simple band pass channel is generally utilized in the result RF phase of a PDA or other remote transmitter to guarantee that main result signals inside the wireless' little, government-endorsed recurrence range are conveyed.

4. Proposed Method

A. Invert Conversion from RNS to Binary Representation

The method is invert change, in which the buildup arithmetic operation is transferred from residue representations to regular writing. This is the RNS's difficult operation. There are two primary approaches for switch transformation in view of the Chinese Remainder Theorem (CRT) and the Mixed-Radix Conversion (MRC).

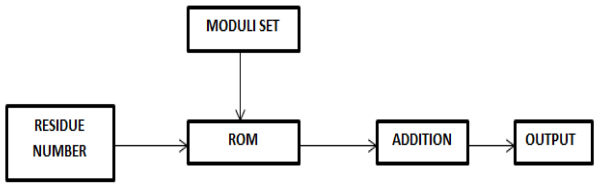

Fig. 2. Block diagram of a reverse conversion

B. FIR Filter based on RNS

Due to their parallelism, modularity, risk toleration, and local carry propagation features, they are ideal for high-speed computerized signal handling. The Residue Number System (RNS) is a good choice for filter realisations. Multiplication and addition are two arithmetic operations that can be performed more quickly. Math activities like increase and expansion can be performed all the more proficiently on account of RNS's restricted convey spread resources. RNS is particularly well suited to the implementation of FIR filters where multiplications and adds are the primary operations.

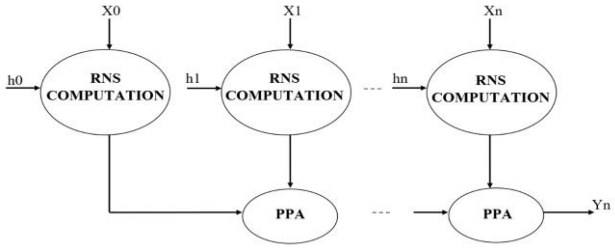

Fig. 3. Basic diagram of FIR filter using RNS system

The above-mentioned features makes RNS more beneficial for digital signal process (DSP) applications, distinctly, when large extent period and high throughput rate are essential.

In this paper, we put forward Residue number system for Multiply Accumulate Units for FIR filer design which has Objectives as like:

- To optimize the propagation delay which is multiplying the producing of partial products in the multipliers.
- To optimize the area and power efficient, the modulo sets are proposed and those sets are $\{2^n\text{-}1, 2^n, 2^n\text{-}1\}$ and all these three sets are part of the Residue Number representation System (RNS).
- To optimize the critical path in the filtering process of Software Defined Radio (SDR) applications.
- To optimize mean square error ratio (SMR).

C. FIR Filter Design

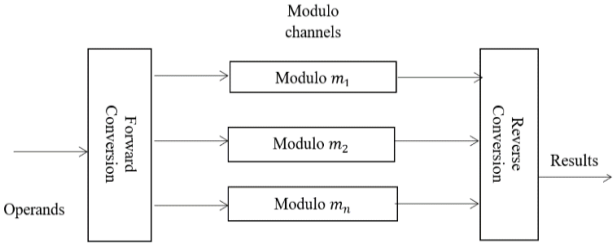

Fig. 4. Block diagram of proposed RNS based FIR filter

In the residue number framework, math calculation is performed utilizing a predetermined moduli set, which comprises of prime whole numbers as moduli sets. The info operands range taken in by the RNS number framework in the output are numerically formulated based on moduli set values without creating any truncation. As moduli, the entire components are employed. RNS structure for a powerful variable M that will accept the endpoints in the range [0, M-1] that are unit regardless of the arithmetic applied. Every moduli and related computations are unit administered as insulating channels in the L parallel technique during RNS computation, which significantly reduces trail propagation delays. [13] Moreover, exploitation changed equal prefix snake geography collection among the RNS system the trail delay is additional optimized is that the projected steps in current styles and also the careful information's and its style half Moduli conversion is shown below:

Considered moduli's (modulo set) {m1, m2,…mp} and its link residue {r1,r2,…rp} whenever the residues are unit associated to the input operands X as given below:

$ri=|X|mi$ and

$$|X|M=|\Sigma ri|Mi-1|mini=1\ Mi|M \quad (14)$$

Where M is that the product of all moduli's, and

Mi=M/mi, this will be rewritten as:

$X=\{r1,r2,r3\ldots..rn\}=\{r10,\ldots\ldots..,0\}+\{0,r2,\ldots\ldots..,0\}+\{0,0\ldots..,rn\}$

the $X=X1+X2+Xn+\cdots...+Xn$ (15)

The post-processing unit referred to as the reverse conversion Process computes Xi's are unit as follows:

$$Xi=ri*\{0,0,\ldots1,\ldots.0,0\}=ri*Xi \quad (16)$$

Where, Xi is computed as |Xi |mi =1.

The Equation that relates ri and its inverse ri-1 is as follows:

$$(ri*ri-1)mod(mi)=1 \quad (17)$$

$Mi\ is\ defined\ as\ M(Mi)$

$where,\ M=\Pi Pi\ ki=1,$

$$then\ ||M1-i||miMi|mi=1 \quad (18)$$

All mi's are unit comparatively prime, thus the inverses exist:

$$Xi=|Mi-1|miMi, \quad (19)$$

$X=\Sigma Xini=1$

$$=\Sigma ri|Mi-1|miMini=1 \quad (20)$$

Because the powerful vary is determined before the computation, each input operand and moduli is unit hand-picked from the specified vary, and the modulo reduction is performed on both sides of the Equation. (20).

The simple FIR filter perform is given by:

$$\Sigma\ M-1j=0\ x(j)h-j=\Sigma\ M-1j=0x(n)h(n-k) \quad (21)$$

Where, k is that the length of filter, here k[0,63].

In equation, a FIR filter operation with a direct structure is proposed, which requires fewer registers and has a simpler internal design (21).

The FIR filter is commonly used in DSP applications for adders and multipliers. The conventional FIR filter style uses a binary number representation method, resulting in longer propagation and delays. To address these issues, the projected RNS-based FIR filter presented in equation (21) uses a faster altered Parallel Prefix Adder (PPA) to avoid carry bit propagation. The results of the existing PPA and the changed PPA are shown in the example below.

Algorithm: The Projected Ternary-RNS based FIR filter

Input: Sample Audio Signal within the kind of Moduli sets ($m1, m2, m3, \ldots\ldots mn$)

Output: Filtered sample audio signal[y(n-k)]

Process 1:

Considered Module set be (7,8,9)

The Input operands be A=14; and B=20;

Let the Input to FIR filter be the Sampled Audio Signal (x)

Process 2: Forward Conversion

Let Ar1 & Br1 be the modulo operation between the quantity and moduli set

Ar1=(A) mod (m1); Ar1=(14)mod(7)=0;

Ar2=(A) mod (m2); Ar2=(14)mod(8)=6;

Ar3=(A) mod (m3); Ar3=(14)mod(9)=5;

Br1=(B) mod (m1); Br1=(20)mod(7)=6;

Br2=(B) mod (m2); Br2=(20)mod(8)=4;

Br3=(B) mod (m3); Br3=(20)mod(9)=2;

Input Residual 1= (0, 6, 5) &

Input Residual 2 = (6, 4, 2).

Process 3: Residue Computation

Compute $r1 = (Ar1 * Br1\ )mod(m1)$=(0*6)mod(7)=0;

Compute $r2 = (Ar2 * Br2\ )mod(m2)$= (6*4)mod(8)=0;

Compute $r3 = (Ar3 * Br3\ )mod(m3)$= (5*2)mod(9)=1;

Residue parts are (0,0,1)

Process 4: Reverse Conversion

Here the inputs be the Moduli set and invM1, invM2, invM3

And the Outputs are unit ROM1, ROM2 and ROM3

Where ROM1 stores computation of

$(M1 * invM1)mod\ (m1)$=1,

Where ROM2 stores computation of

$(M2 * invM2) mod\ (m2)$=1,
Where ROM3 stores computation of
$(M3 * invM3) mod\ (m3)$=1
Compute $M1 = m2 * m3$=8*9=72;
Compute $M2 = m1 * m3$=7*9=63;
Compute $M3 = m1 * m2$=7*8=56;
Compute repetitious method until to urge
$(Mn * invMn) mod\ (mn)$=1
$|M1 * invM1| mod\ (m1)$=|72*(72)^-1|mod(7)=(1)mod(7)=1
$|M2 * invM2| mod\ (m2)$=|63*(63)^-1|mod(8)=(1)mod(8)=1
$|M3 * invM3| mod\ (m3)$= 56*(56)^-1|mod(9)=(1)mod(9)=1

At $M1$ = 72, $M2$ = 63 and $M3$ = 56 we will get output as 1

Process 5: Final Reverse Conversion
Output= $(|M1 * invM1 * r1| + |M2 * invM2 * r2| + |M3 * invM3 * r3|) mod (m1 * m2 * m3)$

=|72*(72)^-1*6|+|63*(63)^-1*2)+|56*(56^-1)*7|mod(7*8*9);
=|280|mod(504)=280
i.e., 14*20=280 (Hence proved)

5. Results

A. ModelSim Output

Initially the Audio text file is been extracted from MATLAB and will be read through $readmemb function. On fetching the Audio coefficients in Xilinx software, the audio filter is simulated and the filtered analog signal is obtained in ModelSim.

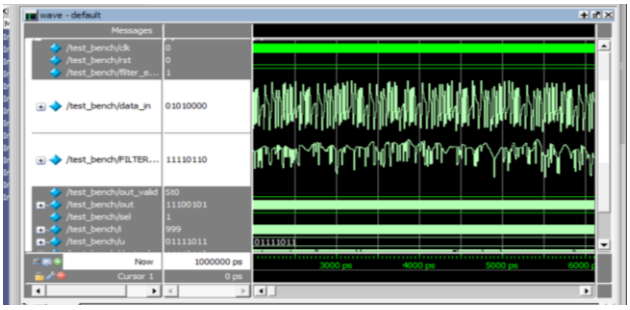

Fig. 5. Modelsim simulation waveform

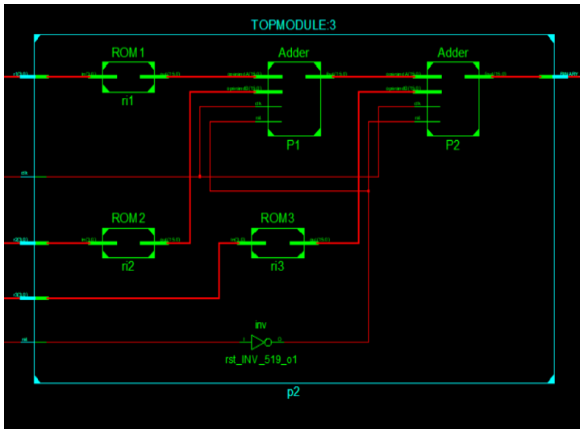

Fig. 6. Portion of RTL schematic view

B. RTL Viewer

From Synthesis, the audio filter is implemented and Register Transfer Level (RTL) Schematic view is shown Fig. 6.

C. Area Utilization Report

The estimated parameters i.e., Area, LUT's, Delay, Frequency report are shown in Fig. 7. & 8.

Audio_Filter Project Status (05/08/2022 - 19:12:27)			
Project File:	topfile.xise	Parser Errors:	No Errors
Module Name:	Audio_Filter	Implementation State:	Synthesized
Target Device:	xc6slx4-3tqg144	• Errors:	
Product Version:	ISE 14.7	• Warnings:	
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	589	4800	12%
Number of Slice LUTs	1985	2400	82%
Number of fully used LUT-FF pairs	375	2199	17%
Number of bonded IOBs	28	102	27%
Number of BUFG/BUFGCTRLs	1	16	6%

Fig. 7. Area summary output

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 9.756ns (Maximum Frequency: 102.499MHz)
   Minimum input arrival time before clock: 3.737ns
   Maximum output required time after clock: 3.634ns
   Maximum combinational path delay: No path found

Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 9.756ns (frequency: 102.499MHz)
  Total number of paths / destination ports: 3707828 / 332
-------------------------------------------------------------------------
Delay:               9.756ns (Levels of Logic = 12)
  Source:            lins/tab/p1/P1/A_9 (FF)
  Destination:       lins/tab/p1/P1/fout_6 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising
```

Fig. 8. Fmax report

D. FPGA Hardware Implementation Output

Hardware simulation and implementation is performed on FPGA Spartan 6 XC6SLX9 in Xilinx ISE plan suite 14.7 by utilization of VHDL equipment portrayal language. The achievement of the hardware implementation is to verify the simulation of analog filter output signal to hardware synthesis in digitized form in hexadecimal values.

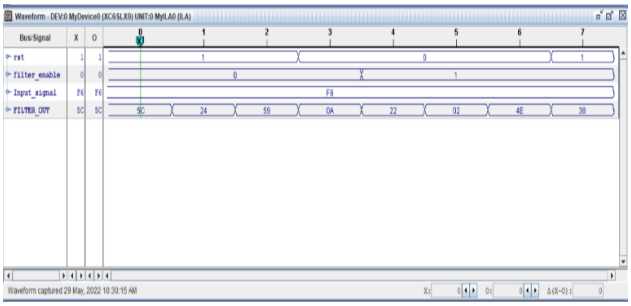

Fig. 9. Digitized form of simulated audio sample

6. Conclusion

In this study, we show that utilising a deconstructed LUT model for FIR design, a high-speed RNS multiplier unit outperforms standard FIR filters. First, we compare the

suggested low-cost RNS-based technique with the maximum memory depth, then we compare the hierarchical assemble of FIR coefficients. To improve the suggested RNS-based model by incorporating approaches for FIR filtering in DSP applications. The FIR implementation unit is used to test the effectiveness of our upgraded RNS coded MAC. Finally, utilising FPGA hardware synthesis, the whole exchange of metrics of the RNS encoded multiplier unit is validated.

References

[1] Oscal T C. Chen, Sandy Wang and Yi-Wen Wu, "Minimization of switching activities of partial products for designing low-power multipliers", IEEE Trans. Very Large Scale Integration (VLSI) Systems. 11 (2003) 418–433.

[2] Jinn-Shyan Wang, Chien-Nan Kuo and Tsung-Han Yang, "Low-power fixed-width array multipliers", in Proc. IEEE Int. Symp. Low Power Electronics and Design (ISLPED04), 2004, pp. 307–312.

[3] Huang Zhijun and Milos D. Ercegovac, "High-performance low-power left-to-right array multiplier design", IEEE Trans. Computers. 54 (2005) 272–283.

[4] Kuan Hung Chen and Yuan Sun Chu, "A low-power multiplier with the spurious power suppression technique", IEEE Trans. Very Large Scale Integration (VLSI) Systems. 15 (2007) 846–850.

[5] Hasan Krad and Aws Yousif Al Taie, "Performance analysis of a 32-bit multiplier with a carry look ahead adder and a 32-bit multiplier with a ripple adder using VHDL", J. Computer Science. 4(4) (2008) 305–308.

[6] M. Mottaghi-Dastjerdi, A. Afzali-Kusha, and M. Pedram, BZ-FAD: "A low-power low area multiplier based on shift-and-add architecture", IEEE Trans. Very Large Scale Integration (VLSI) Systems. 17 (2009) 302–306.

[7] Aloke Saha, Dipankar Pal and Mahesh Chandra, "Low-power 6-GHz wave-pipelined 8b × 8b multiplier", IET Circuits, Devices & Systems. 7(3) (2013) 124–140.

[8] Cong Liu, Jie Han and Fabrizio Lombardi, "A low-lower, high-performance approximate multiplier with configurable partial error recovery", in Proc. IEEE Design, Automation and Test in Europe Conf. and Exhibition (DATE), (2014), pp. 1–4.

[9] Basant K. Mohanty and Vikas Tiwari, "Modified PEB formulation for hardware efficient fixed-width Booth multiplier", J. Circuits Syst. Signal Process. 33 (2014) 3981– 3994.

[10] Botang Shao and Peng Li, "Array-based approximate arithmetic computing: A general model and applications to multiplier and squarer design", IEEE Trans. Circuits and Systems-I: Regular Papers. 62 (2015) 1081–1090.

[11] When-Quan He, Yuan-Ho Chen and Shyh-Jye Jou, "Dynamic error compensated fixed width Booth multiplier based on conditional-probability of input series", J. Circuits Syst. Signal Process. 35 (2016) 2972–2991.

[12] Zain Shabbir, Anas Razzaq Ghumman and Shabbir Majeed Chaudhry, "A reduced-sp D3Lsum adder based high frequency 4 × 4 bit multiplier using Dadda algorithm", J. Circuits Syst. Signal Process. 35 (2016) 3113–3134.

[13] Ahmad Hiasat.et.al, "A Scaler Design for the RNS Three-Moduli Set {2n+1-1, 2n, 2n-1} Based on Mixed-Radix Conversion", Journal of Circuits, Systems, and Computers, Vol. 29, No. 3, 2020.

[14] Raj Kumar.et.al, "Perspective and Opportunities of Modulo 2n-1 Multipliers in Residue Number System: A Review", Journal of Circuits, Systems, and Computers, Vol. 29, No. 11, 2020.

[15] Grande Naga Jyothi.et.al, "ASIC implementation of distributed arithmetic-based FIR filter using RNS for high speed DSP systems", International Journal of Speech Technology, Springer, 2020.

[16] Rami Akeela.at.al, "Software-defined Radios: Architecture, State-of-the-art, and Challenges", Internet of Things Research Lab, Department of Computer Engineering, Santa Clara University, USA, March 2018.