Low Power and High Speed Implementation for Symmetric Convolutions Based FIR Digital Filter Structures

K. SOUJANYA¹, B. VENU GOPAL²

¹Research Scholar, Dept of ECE, Kaushik College of Engineering, Visakhapatnam, AP-India, E-mail: soujanya.k2@gmail.com.
²Asst Prof, Dept of ECE, Kaushik College of Engineering, Visakhapatnam, AP-India, E-mail: basanigopal2@gmail.com.

Abstract: The main challenging areas in VLSI are performance, cost, testing, area, reliability, delay and power. The demand for portable computing devices and communications system are increasing rapidly. These applications require low power dissipation and low area with high speed for VLSI circuits. Hence it is important aspect to optimize power, and area(number of resources) and speed. So these constraints optimization became one of the main challenges. In this a Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm are designed with area and power efficient. To optimize these structures a data flow HDL model is preferred because of consuming less resources when compare with other modeling schemes. In this paper Xilinx-ISE tool is used for logical verification and further synthesizing it on Xilinx-ISE tool using target technology.

Keywords: Filter, Symmetric,Xilinx, Verilog.

I. INTRODUCTION

Filter is a frequency selective network. It passes a band of frequencies while attenuating the others. Filters are classified as analog and digital depending on nature of inputs and outputs. Filters are further classified as finite impulse response and infinite impulse response filters depending on impulse response. Filters of some sort are essential to the operation of most electronic circuits. It is therefore in the interest of anyone involved in electronic circuit design to have the ability to develop filter circuits capable of meeting a given set of specifications. In circuit theory, a filter is an electrical network that alters the amplitude and/or phase characteristics of a signal with respect to frequency. Ideally, a filter will not add new frequencies to the input signal, nor will it change the component frequencies of that signal, but it will change the relative amplitudes of the various frequency components and/or their phase relationships. Filters are often used in electronic systems to emphasize signals in certain frequency ranges and reject signals in other frequency ranges. Such a filter has again which is dependent on signal frequency. This chapter gives a brief about the types of filters.

A. Analog Filters

Very simple analog low pass or high pass filters can be constructed from resistor and capacitor (RC) networks. In the low pass case, a potential divider is formed from a series resistor followed by a shunt capacitor. The filter input is at one end of the resistor and the output is at the point where the resistor and capacitor join. The RC filter works because the capacitor reactance reduces as the frequency increases. It should be remembered that the reactance is 90° out of phase with resistance. At low frequencies the reactance of the capacitor is very high and the output voltage is almost equal to the input, with virtually no phase difference. At the cutoff frequency, the resistance and the capacitive reactance are equal and the filter's output is 1/2 of the input voltage, or 3 dB. At this frequency the output will not be in phase with the input: it will lag by 45° due to the influence of the capacitive reactance. At frequencies above the 3 dB attenuation point, the output voltage will reduce further. The rate of attenuation will be 6 dB per doubling of frequency (per octave). As the frequency rises, the capacitive reactance falls and the phase shift lag approaches 90°. Analogue filters are a basic building block of signal processing much used in electronics. Amongst their many applications are the separation of an audio signal before application to bass, mid-range and tweeter loudspeakers; the combining and later separation of multiple telephone conversations onto a single channel; the selection of a chosen radio station in a radio receiver and rejection of others.

B. Digital Filters

Digital filters are used extensively in all areas of electronic industry. This is because digital filters have the potential to attain much better signal to noise ratios than analog filters and at each intermediate stage the analog filter adds more noise to the signal, the digital filter performs noiseless mathematical operations at each intermediate step in the transform. The digital filters have emerged as a strong...
option for removing noise, shaping spectrum, and minimizing inter-symbol interference in communication architectures. These filters have become popular because their precise reproducibility allows design engineers to achieve performance levels that are difficult to obtain with analog filters. Filters are widely employed in signal processing and communication systems in applications such as channel equalization, noise reduction, radar, audio processing, video processing, biomedical signal processing, and analysis of economic and financial data. For example in a radio receiver band-pass filters, or tuners, are used to extract the signals from a radio channel. In an audio graphic equalizer the input signal is filtered into a number of subband signals and the gain for each sub-band can be varied manually with a set of controls to change the perceived audio sensation. In a Dolby system pre-filtering and post filtering are used to minimize the effect of noise. In hi-fi audio a compensating filter may be included in the preamplifier to compensate for the non-ideal frequency-response characteristics of the speakers. Figure 1 shows a graphical means of describing a digital filter whereby the behavior of the filter is described by using the mathematical operations mentioned above.

The Impulse Response of a digital filter, $h(n)$ is the response of the filter to an input consisting of the unit impulse function, $\delta(n)$. If the impulse response of a system is known, it is possible to calculate the system response for any input sequence $x(n)$. By definition, the unit impulse is applied to a system at sample index $n=0$. So, the impulse response is non-zero only for values of $n$ greater than or equal to zero i.e $h(n)$ is zero for $n<0$. This impulse response is said to be causal otherwise the system would be producing a response before an input has been applied. It is known from the time-invariance property of a Linear Time Invariant System that the response of a system to a delayed unit impulse $\delta(n-k)$ will be a delayed version of the unit impulse, i.e $h(n-k)$. It is also known from the linearity property that the response of a system to a weighted sum of inputs will be a weighted sum of responses of the system to each of the individual inputs. Therefore, the response of a system to an arbitrary input $x(n)$ can be written as follows:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k) h(n-k)$$

(1)

II. DIGITAL FIR FILTER

Linear phase refers to the fact that the phase response of the filter is a straight-line function of frequency. This means that the delay through the filter will be the same at all frequencies. This is the major advantage of FIR filters. This chapter discusses the concepts and various methods for the design and realization of FIR filters. The design specifications of the implemented filter were also presented. Filters are signal conditioners. Each functions by accepting an input signal, blocking pre specified frequency components, and passing the original signal minus those components to the output. For example, a typical phone line acts as a filter that limits frequencies to a range considerably smaller than the range of frequencies human beings can hear. That's why listening to CD-quality music over the phone is not as pleasing to the ear as listening to it directly.

A digital filter takes a digital input, gives a digital output, and consists of digital components. In a typical digital filtering application, software running on a digital signal processor (DSP) reads input samples from an A/D converter, performs the mathematical manipulations dictated by theory for the required filter type, and outputs the result via a D/A converter. An analog filter, by contrast, operates directly on the analog inputs and is built entirely with analog components, such as resistors, capacitors, and inductors. There are many filter types, but the most common are low pass, high pass, band pass, and band stop. A low pass filter allows only low frequency signals (below some specified cutoff) through to its output, so it can be used to eliminate high frequencies. A low pass filter is handy, in that regard, for limiting the uppermost range of frequencies in an audio signal; it's the type of filter that a phone line resembles.

A. Finite Impulse Response Filter

Finite Impulse Response (FIR) filters are one of two primary types of filters used in DSP, the other type being Infinite Impulse Response Filters (IIR) filters. The impulse response of an FIR filter is "finite" because there is no feedback in the filter. Compared to IIR filters, FIR filters offer the following advantages:

- They can easily be designed to be "Linear Phase". Linear-Phase filters delay the input signal, but don’t distort its phase
- They are simple to implement. On most DSP microprocessors, looping a single instruction can do the FIR calculation.
- FIR filters are suited to multi-rate applications. i.e: reducing the sampling rate (decimation) or increasing the sampling rate (interpolation), or both. Whether decimating or interpolating, the use of FIR filters allows some of the calculations to be omitted, thus providing an important computational efficiency. In
Low Power and High Speed Implementation for Symmetric Convolutions Based FIR Digital Filter Structures

contrast, if IIR filters are used, each output must be individually calculated, even if that output will be discarded (so the feedback will be incorporated into the filter).

- FIR filters have desirable numeric properties. In practice, all DSP filters must be implemented using "finite-precision" arithmetic, that is, a limited number of bits. The use of finite-precision arithmetic in IIR filters can cause significant problems due to the use of feedback, but FIR filters have no feedback, so they can usually be implemented using fewer bits, and the designer has fewer practical problems to solve related to non-ideal arithmetic.

- FIR filters can be implemented using fractional arithmetic. Unlike IIR filters, it is always possible to implement a FIR filter using coefficients with magnitude of less than 1.0. (The overall gain of the FIR filter can be adjusted at its output, if desired.)

A disadvantage of using FIR filters is that they require more co-efficient than an IIR filter in order to implement the same frequency response, therefore needing more memory and more hardware resources to carry out mathematical operations.

**B. Linear Phase**

Linear phase refers to the fact that the phase response of the filter is a straight-line function of frequency. This means that the delay through the filter will be the same at all frequencies. As a result, the filter does not cause phase/delay distortion, which can be a major advantage over IIR or analogue filters in certain applications. An FIR filter is linear phase if and only if its co-efficient are symmetrical around the centre co-efficient i.e. the first co-efficient is the same as the last; the second is the same as the second last etc. It takes a finite time before the input to a filter propagates through the filter and appears at the output. A delay of \( T \) seconds in time domain appears as a phase change of \( e^{-j\omega T} \) in frequency domain. The time delay, or phase change, caused by a filter depends on the type of the filter (e.g. linear/nonlinear, FIR/IIR) and its impulse response. An FIR filter of order \( M \), with \( M+1 \) coefficient, has a time delay of \( M/2 \) samples.

**III. FAST FIR ALGORITHM (FFA)**

We first establish through a simple example a new fast FIR filtering algorithm based on a divide-and-conquer approach. This algorithm does not require the use of overlap techniques as is usual in the approaches based on cyclic or a periodic convolutions. We outline the advantages of the proposed algorithm when implemented both in software and in hardware. Finally, we give a systematic way of deriving these algorithms. Consider an N-tap FIR filter which can be expressed in the general form as

\[
y(n) = \sum_{i=0}^{N-1} h(i)x(n-i), \quad n = 0, 1, 2, \ldots, \infty
\]  

(2)

Where \( \{x(n)\} \) is an infinite length input sequence and \( \{h(i)\} \) are the length –N FIR filter coefficients. Then, the traditional L-parallel FIR filter can be derived using poly phase decomposition as [3]

\[
\sum_{p=0}^{L-1} Y_p(z^L)z^{-r} = \sum_{p=0}^{L-1} X_p(z^L)z^{-r} \sum_{r=0}^{L-1} H_r(z^L)z^{-r}
\]

(3)

Where,

\[
X_p = \sum_{k=0}^{\infty} z^{-k} x(Lk + p),
\]

(5)

Equation (5) shows the traditional two-parallel filter structure, which will require four length-N/2 FIR sub filter blocks, two post processing Adders, and totally 2N multipliers and 2N - 2 adders. However, (5) can be written as

\[
\begin{align*}
Y_0 &= H_0 X_0 + z^{-2} H_1 X_1, \\
Y_1 &= H_0 X_1 + H_1 X_0.
\end{align*}
\]

(5)

Figure 2: Two-parallel FIR filter implementation using FFA.
The implementation of (6) will require three FIR subfilter blocks of length N/2, one preprocessing and three postprocessing adders, and 3N/2 Multipliers and 3(N/2-1) +1 adders, which reduces approximately one fourth over the traditional two-parallel filter hardware cost from (5). The two-parallel (L=2) FIR filter implementation using FFA obtained from (6) is shown in Fig. 2. By the similar approach, a three-parallel FIR filter using FFA can be expressed as

$$Y_0 = H_0 X_0 - z^{-3} H_2 X_2 + z^{-3} \times [(H_1 + H_2)(X_1 + X_2) - H_1 X_1]$$

$$Y_1 = [(H_0 + H_1)(X_0 + X_1) - H_1 X_1]$$

$$Y_2 = [(H_0 + H_1 + H_2)(X_0 + X_1 + X_2)]$$

$$= [(H_0 + H_1)(X_0 + X_1) - H_1 X_1]$$

$$= [(H_1 + H_2)(X_1 + X_2) - H_1 X_1].$$

(7)

The hardware implementation of (7) requires six length-N/3 FIR sub-filter blocks, three preprocessing and seven postprocessing adders, and three N multipliers and 2N+1 adders, which has reduced approximately one third over the traditional three-parallel filter hardware cost. The implementation obtained from (7) is shown in Fig. 3.

B. Proposed FFA Structures for Symmetric Convolutions

To utilize the symmetry of coefficients, the main idea behind the proposed structures is actually pretty intuitive, to manipulate the poly phase decomposition to earn as many subfilter blocks as possible which contain symmetric coefficients so that half the number of multiplications in the single sub filter block can be reused for the multiplications of whole taps, which is similar to the fact that a set of symmetric coefficients would only require half the filter length of multiplications in a single FIR filter. Therefore, for an N-tap L-parallel FIR filter the total amount of saved multipliers would be the number of sub filter blocks that contain symmetric coefficient times half the number of multiplications in a single sub filter block (N/2L).

1. 2×2 Proposed FFA (L=2)

From (5), a two-parallel FIR filter can also be written as:

$$Y_0 = \frac{1}{2}[(H_0 + H_1)(X_0 + X_1)$$

$$+ (H_0 - H_1)(X_0 - X_1)] - H_1 X_1 \right\} + z^{-2} H_1 X_1.$$

$$Y_1 = \frac{1}{2}[(H_0 + H_1)(X_0 + X_1)$$

$$- (H_0 - H_1)(X_0 - X_1)].$$

(8)

When it comes to a set of even symmetric coefficients, (8) can earn one more sub filter block containing symmetric coefficients than (6), the existing FFA parallel FIR filter. Fig. 3 shows implementation of the proposed two-parallel FIR filter based on (8). An example is demonstrated here for a clearer perspective.

Example 1: Consider a 24-tap FIR filter with a set of symmetric coefficients applying to the proposed two-parallel FIR filter.

$$\{h(0), h(1), h(2), h(3), h(4), h(5),$$

$$h(6), h(7), h(8), h(9), \ldots, h(23)\}$$

where

$$h(0) = h(23), h(1) = h(22), h(2) = h(21), h(3) = h(20), h(4) = h(19),$$

$$h(5) = h(18), \ldots, h(11) = h(12),$$

applying to the proposed two-parallel FIR filter structure, and the top two subfilter blocks will be as

$$H_0 \pm H_1 = \{h(0) \pm h(1), h(2) \pm h(3),$$

$$h(4) \pm h(5), h(6) \pm h(7), \ldots, h(18) \pm h(19),$$

$$h(20) \pm h(21), h(22) \pm h(23)\}$$

Figure 3: Three-parallel FIR filter implementation using FFA.

Figure 4: Proposed two-parallel FIR filter implementation.
Low Power and High Speed Implementation for Symmetric Convolutions Based FIR Digital Filter Structures

Where,

\[ h(0) \pm h(1) = \pm (h(22) \pm h(23)) \]
\[ h(2) \pm h(3) = \pm (h(20) \pm h(21)) \]
\[ h(4) \pm h(5) = \pm (h(18) \pm h(19)) \]
\[ h(6) \pm h(7) = \pm (h(16) \pm h(17)) \ldots \] (9)

Figure 5: Sub filter block implementation with symmetric coefficients.

Figure 6: Proposed three-parallel FIR filter implementation.

As can be seen from the example above, two of three sub filter blocks from the proposed two-parallel FIR filter structure, H0-H1 and H0+H1, are with symmetric coefficients now, as (9), which means the sub-filter block can be realized by Fig. 4, with only half the amount of multipliers required. Each output of multipliers responds to two taps. Note that the transposed direct-form FIR filter is employed. Compared to the existing FFA two-parallel FIR filter structure, the proposed FFA structure leads to one more sub filter block which contains symmetric coefficients. However, it comes with the price of the increase of amount of adders in preprocessing and post processing blocks. In this case, two additional adders are required for L=2.

Figure 7: Comparison of sub filter blocks between existing FFA and the proposed

Figure 8: Comparison of sub filter blocks between existing FFA and proposed FFA four-parallel FIR structures.

Figure 9: Proposed four-parallel FIR filter implementation.
Proposed FFA (L=3)

With the similar approach, from (7), a three-parallel FIR filter can also be written as (10). Fig. 5 shows implementation of the proposed three-parallel FIR filter. When the number of symmetric coefficients N is the multiple of 3, the proposed three-parallel FIR filter structure presented in (10) enables four sub filter blocks with symmetric coefficients in total, whereas the existing FFA parallel FIR filter structure has only two ones out of six sub filter blocks. A comparison figure is shown in Fig. 6, where the shadow blocks stand for the sub filter blocks which contain symmetric coefficients. Therefore, for an N-tap three-parallel FIR filter, the proposed structure can save N/3 multipliers from the existing FFA structure. However, again, the proposed three-parallel FIR structure also brings an overhead of seven additional adders in preprocessing and post processing blocks.

C. Proposed Cascading FFA

The proposed cascading process for the larger block-sized proposed parallel FIR filter is similar to that introduced in past. However, a small modification is adopted here for lower hardware consumption. As we can see, the proposed parallel FIR structure enables the reuse of multipliers in parts of the sub filter blocks but it also brings more adder cost in preprocessing and post processing blocks. When cascading the proposed FFA parallel FIR structures for larger parallel block factor L, the increase of adders can become larger. Therefore, other than applying the proposed FFA FIR filter structure to all the decomposed sub filter blocks, the existing FFA structures which have more compact operations in preprocessing and post processing blocks are employed for those sub filter blocks that contain no symmetric coefficients, whereas the proposed FIR filter structures are still applied to the rest of sub filter blocks with symmetric coefficients.

An illustration of the proposed cascading process for a four-parallel FIR filter (L=4) as an example is shown in Fig. 7, and the realization is shown in Fig. 8. From Fig. 7, it is clear to see that the proposed four-parallel FIR structure earns three more sub filter blocks containing symmetric coefficients than the existing FFA one, which means 3N/8 multipliers can be saved for an N-tap FIR filter, at the price of 11 additional adders in preprocessing and post processing blocks. By this cascading approach, parallel FIR filter structures with larger block factor L can be realized. The proposed six-parallel FIR filter will result in 6 more symmetric sub filter blocks, equivalently N/2 multipliers saved for an N-tap FIR filter, than the existing FFA, at the expense of an additional 32 adders. Also, the proposed eight-parallel FIR filter will lead to seven more symmetric sub filter blocks, equivalently 7N/16 multipliers saved for an N-tap filter, than the existing FFA, with the overhead of additional 54 adders.

IV. SYNTHESIS AND SIMULATION RESULTS

A. Schematic and Simulation Result of Two-Parallel FIR Filter using FFA

Figure 10 and 11 shows the schematic and simulation results of two-parallel FIR filter.
Low Power and High Speed Implementation for Symmetric Convolutions Based FIR Digital Filter Structures

B. Schematic and Simulation Result of Three-Parallel FIR Filter using FFA

C. RTL schematic and Simulation Result of Proposed Two-Parallel FIR Filter implementation:

D. RTL schematic and Simulation Result of Proposed Three-Parallel FIR Filter implementation:
Figure 17: Waveform of Proposed Three - Parallel FIR Filter.

E. RTL schematic and Simulation Result of 24 –Tap FFA Filter:

Figure 18: RTL schematic of 24 –Tap FFA Filter

Figure 19: Waveform of 24-Tap FFA Filter

F. RTL schematic and Simulation Result of Proposed 24  
–Tap FIR Filter:

Figure 20 : RTL schematic Of Proposed 24 –Tap FIR Filter
G. RTL schematic and Simulation Result of 72 –Tap FFT Filter:

Figure 21: Waveform of Proposed 24 –Tap FIR Filter

Figure 22: RTL schematic of 72–Tap FFA Filter

Figure 23: Waveform of 72 –Tap FFA Filter
H. RTL schematic and Simulation Result of Proposed 72–Tap FIR Filter:

Figure 24: RTL schematic Of Proposed 72–Tap FIR Filter

I. Final Report

24 & 72-tap FFT and Proposed Output Results
Comparison of Delay, Area, Power & Luts

**TABLE I: COMPARISON OF DELAY**

<table>
<thead>
<tr>
<th>LENGTH</th>
<th>STRUCTURE</th>
<th>L=2</th>
<th>L=3</th>
<th>L=4</th>
<th>L=6</th>
<th>L=8</th>
</tr>
</thead>
<tbody>
<tr>
<td>24-tap</td>
<td>FFA</td>
<td>3.871</td>
<td>4.71</td>
<td>3.653</td>
<td>3.656</td>
<td>3.656</td>
</tr>
<tr>
<td>72-tap</td>
<td>FFA</td>
<td>6.352</td>
<td>6.255</td>
<td>5.953</td>
<td>5.48</td>
<td>5.463</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>6.996</td>
<td>6.438</td>
<td>6.301</td>
<td>5.5</td>
<td>7.181</td>
</tr>
</tbody>
</table>

**TABLE II: COMPARISON OF AREA**

<table>
<thead>
<tr>
<th>LENGTH</th>
<th>STRUCTURE</th>
<th>L=2</th>
<th>L=3</th>
<th>L=4</th>
<th>L=6</th>
<th>L=8</th>
</tr>
</thead>
<tbody>
<tr>
<td>24-tap</td>
<td>FFA</td>
<td>246594</td>
<td>245496</td>
<td>245496</td>
<td>245449</td>
<td>245496</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>244804</td>
<td>244844</td>
<td>247344</td>
<td>245449</td>
<td>245498</td>
</tr>
<tr>
<td>72-tap</td>
<td>FFA</td>
<td>247608</td>
<td>249528</td>
<td>246584</td>
<td>248464</td>
<td>247544</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>249464</td>
<td>249384</td>
<td>247544</td>
<td>247544</td>
<td>250040</td>
</tr>
</tbody>
</table>

**TABLE III: COMPARISON OF POWER**

<table>
<thead>
<tr>
<th>LENGTH</th>
<th>STRUCTURE</th>
<th>L=2</th>
<th>L=3</th>
<th>L=4</th>
<th>L=6</th>
<th>L=8</th>
</tr>
</thead>
<tbody>
<tr>
<td>24-tap</td>
<td>FFA</td>
<td>5</td>
<td>6</td>
<td>13</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>27</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>72-tap</td>
<td>FFA</td>
<td>108</td>
<td>124</td>
<td>171</td>
<td>236</td>
<td>298</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>93</td>
<td>101</td>
<td>127</td>
<td>146</td>
<td>248</td>
</tr>
</tbody>
</table>

**TABLE IV: COMPARISON OF LUTS**

<table>
<thead>
<tr>
<th>LENGTH</th>
<th>STRUCTURE</th>
<th>L=2</th>
<th>L=3</th>
<th>L=4</th>
<th>L=6</th>
<th>L=8</th>
</tr>
</thead>
<tbody>
<tr>
<td>24-tap</td>
<td>FFA</td>
<td>8</td>
<td>13</td>
<td>1</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>72-tap</td>
<td>FFA</td>
<td>17</td>
<td>24</td>
<td>24</td>
<td>43</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>24</td>
<td>28</td>
<td>35</td>
<td>32</td>
<td>37</td>
</tr>
</tbody>
</table>
V. CONCLUSION

In this paper the realization of FFA and proposed FIR structures has been carried out using Verilog language. From the table values it should be concluded that the power consumption, critical path delay and memory usage values should be optimized when compared with the existed one. In this paper, we have presented new parallel FIR filter structures, which are beneficial to symmetric convolutions when the number of taps is the multiple of 2 or 3. Multipliers are the major portions in hardware consumption for the parallel FIR filter implementation. Overall, in this paper, we have provided new parallel FIR structures consisting of advantageous dealing with symmetric convolutions comparatively better than the existing FFA structures in terms of hardware consumption. In future there is a chance to optimize the VLSI constraints (power, memory, speed) somewhat better along with estimating for analog signals also using high end EDA software’s.

VI. REFERENCES


Author’s Profile:

Kuna Soujanya, has completed B.Tech (E.C.E) from SHRI VISHNU Engineering College For Women, pursuing M.Tech in KAUSHIK College of Engineering, affiliated to JNTUK, Andhra Pradesh, India. Her main research interest includes in Electronics, Embedded & VLSI Systems.

B. Venugopal, did B.Tech, M.Tech and currently working as assistant professor in KAUSHIK college of engineering, affiliated to JNTUK, Andhra Pradesh, India. His main research interests include Communications, Signal Processing and VLSI.