A Novel VLSI design and implementation of ultra low power digital FIR Filter based on a new signed shift/add multiplier with multiplexer (2 To 1) by pass transistor logic

Bahram Rashidi¹ and Majid Pourormazd²
¹University of Tabriz, Iran.
²University of Shahid Chamran Kerman, Iran.

ABSTRACT
In this paper, a new ultra low power digital linear phase finite Impulse Response (FIR) filter based on parallel architecture of Multiplication and Accumulation (MAC) with a new logic approach by Pass-Transistor Logic (PTL) is presented. PTL as an alternative logic that can enhance the design performance since PTL can transformation signals using either the source or drain and the gate. This multiplier is include three blocks booth encoder, new signed shifter and addition block. Booth encoder and shifter block are completely based on multiplexers (2-to-1). We tried to use low power design methods both in gate level and transistor level to reach minimum hardware and power consumption other reons for reduce power consumption are use D-Flip-Flop’s aim at glitch and charge sharing free structures, low power energy recovery full adder cells has been used in adders, also for increase speed, two parallel MAC are utilized. HSPICE in 0.18μm bulk technology at 1.8v supply voltage. Power consumption is 55uw at 100MHz.

© 2012 Elixir All rights reserved.

Introduction
In the field of Digital Signal Processing (DSP) filters play an important role. For instance, digital filters used in radio transmitters and receivers operating at a high sampling rate form an interesting class. For these filters, efficiency is crucial. Application of filters with a different behaviour for positive and negative frequencies is beneficial in many cases such as in multirate systems. In such filters some coefficients will be complex. Low power design is equivalent to eliminating redundant operations (to minimize the dynamic power) and minimizing the circuit size (to minimize the static power). Power efficiency and silicon efficiency are more related because the static power is no longer negligible and even becomes dominant. Low power DSP ASICs can be implemented in many ways because there is more freedom or more dimensions for power minimization. In Equation 1, power consumption can be partitioned into dynamic power and static power (V is the supply voltage for full swing logic circuits, α is the activity of a circuit node, C is the total capacitance of the circuit node, and f is the clock frequency of the digital circuit). The static power consumption is the accumulation of leakage power of each circuit path from the power supply to the ground.

\[ P = V^2 \sum_{i} \alpha_i f C_i + P_{static} \]  

(1)

The most significant power savings can be achieved by reducing the supply voltage. Reducing the circuit capacitance and speed enables linear power reduction. However, mandatory constraints such as the required circuit performance must be met during the power minimization. Carefully trading off dynamic and static power consumption is required. For example, the hardware multiplexing technique could be used to reduce power consumption by minimizing leakage paths. Both memory power and interconnect power can be reduced by minimizing the number of memory accesses. The first step of power minimization is to identify which part of the circuit consumes significant power [1]. The power optimization will start from the circuit modules that consume most of the power. The second step is to reduce the supply voltage. As the circuit speed is reduced, the performance of the circuit should be maintained by mapping algorithms and arithmetic functions to parallel hardware. Power efficiency is measured by performance over power consumption. This paper focuses on methods for improving the efficiency of symmetric filters. FIR filters with a symmetric impulse response show a linear phase frequency response.

Related works
In recent years a number of techniques have been proposed for low power implementation of FIR filters. These include the following: In [2] presents the implementation of the decorrelating (DECOR) transformation technique for low power FIR filtering cores. The technique was introduced in the past, but was not fully evaluated for its area, delay and power performance. Early evaluations did not consider the whole implementation and were merely based on either some analytical methods or high level simulation models. They presents the complete VLSI implementation of the technique and a study of its area, delay and power performance with different order of coefficient differences and various multiplier types. A digit reconfigurable FIR filter architecture with a very fine granularity is presented by[3]. It provides a flexible yet compact and low-power solution to FIR filters with a wide range of precision and tap length. In[4] a low power FIR filter using folded direct form (FDF) structure is designed which is a key component in a
hearing transistors is proposed that uses dual edge triggering. A novel approach for implementing power-efficient FIR filters that requires less power consumption than traditional FIR filter implementation in wireless embedded systems. The proposed schemes can be adopted in the direct form FIR filter and achieve a large amount of reduction in the power consumption. By using a combination of proposed methods, balanced-modular techniques with retiming and separated processing data-flow scheme with modified canonical signed digit (CSD) representation by [5]. Asynchronous design is progressively becoming more attractive alternative to synchronous design because of its potential for high-speed and low-power. The pipelining technique is very effective for synchronous digital designs. in [6], proposes the design of pipelined FIR filter using asynchrony quasi-delay-insensitive (QDI) template based on Reduced Slack PreCharged Half Buffer (RSPCHB). A novel FIR filter synthesis technique that allows for aggressive voltage scaling by exploiting the fact that all filter coefficients are not equally important to obtain a “reasonably accurate” filter response. technique implements a level-constrained common sub expression-elimination(CSE) algorithm, where they can constrain the number of adder levels (ALs) required to compute each of the coefficient outputs. By specifying a tighter constraint (in terms of the number of adders in the critical path) on the important coefficients. They have presented a novel synthesis technique for generating FIR filter designs which simultaneously cater to low-energy requirements and tolerance to large process variations while maintaining a reasonably accurate filter response. This is achieved by restricting “important filter coefficients to a less number of computation steps than the maximum allowed in a CSE-based filter implementation by [7]. In[8] they describe the novel algorithm for designing low-power and hardware efficient FIR filter. It is mandatory for any filter designer to propose a low power multiplier as most of the power consumption of the filter occurs in multiplier unit. Hence, they proposed a novel modified Wallace tree multiplier. In this paper, we design and implementation a new approach to the low power design of an FIR filter based on the parallel MAC architecture and new low power signed shift/add multiplier. These blocks are complete based on 2 to 1 multiplexer and also in transistor level we use the dynamic D-flip-flop aiming at glitch and charge free, pass transistor logic(PTL) for implementation of 2 to 1 multiplexer and XOR gate because PTL as an alternative logic that can enhance the design performance since PTL can transformation signals using either the source or drain and the gate, its high functionality can reduce the number of transistors in critical path., implementation of full adder based on low power energy recovery full adder cell. These approaches have excellent results in reduce power consumption and increase performance and speed. The remainder of the paper is organized as follows: Section (3) focuses on theory of FIR filter Section (4) emphasizes on proposed architecture in gate level and transistor level, comparison of proposed implementation with other works done in section(5). Finally section(6) provides the conclusion of this paper.

Theory of Digital FIR Filter

FIR filters constitute a class of digital filters having a Finite-length Impulse Response. An FIR filter can be realized using non-recursive as well as recursive algorithms. However, the latter are not recommended due to potential stability problems while non-recursive FIR filters are always stable. Hence, non-recursive FIR filter algorithms are preferable for implementation. An FIR filter can be described by the difference equation (2).

\[ y(n) = \sum_{k=0}^{N} h(k) * x(n-k) \]  

(2)

Where \(y(n)\) is the filter output, \(x(n)\) is the filter input, \(N\) is the filter order, and \(h(n)\) are the impulse response coefficients of the filter.

FIR Filter Structures

The computational properties of a digital filter algorithm can be described with a fully specified signal-flow graph. In such graphs the ordering of all operations is uniquely specified. A digital filter can often be implemented using different algorithms, i.e., different fully specified signal-flow graphs. A nonrecursive FIR filter can be realized using different structures. Here, two basic FIR filter structures are considered; the direct form and the transposed direct form. Other structures can also be used for realization of FIR filters, such as difference coefficient structures [1]:

Direct Form FIR Filter Structure

The direct form FIR filter structure is directly derived from equation (2). An Nth-order direct form structure is composed by N memory elements (registers) holding the input value for N sample periods, N + 1 multipliers, corresponding to the impulse response coefficients in equation (2), and N additions for summation of the results of the multiplications. The term “direct” indicates that the impulse response values are used as coefficients in the realization.

Linear-Phase FIR Filter Structures

An important property of FIR filters is that they can have an exact linear phase response. To obtain this, the FIR filter must have a symmetric or anti-symmetric impulse response. The impulse response of a linear phase FIR filter is either symmetric around \(n = N/2\), \(h(n) = h(N - n)\), \(n = 0,1,\ldots N\) or anti-symmetric around \(n = N/2\), \(h(n) = -h(N - n)\), \(n = 0,1,\ldots N\) where N is the filter order. For a linear-phase FIR filter the number of multiplications required can be reduced by exploiting the symmetry of the impulse response Figure 1 shows direct form and linear phase of FIR structure. From the Figure 1, it can also be seen that the number of additions remains the same while the number of multiplications is halved, compared to the corresponding direct form implementation [1].

![Figure 1: (a) Direct form FIR filter, (b) Linear-phase filter with reduced number of multipliers.](image-url)
Proposed Architecture

In this paper, we offer the linear phase of dedicated digital FIR filter with parallel MAC FIR filter based on a new low power signed shift/add multiplier. Our multiplier is contain a Booth encoder until we reduce the number of partial product. For implementation of booth encoder and signed shifter blocks in shift/add multiplier, We often have the multiplexers (2-to-1). In transistor-level multiplexer realized with only two transistors are implemented by Pass Transistor Logic, as an alternative logic that can enhance the design performance since PTL can transformation signals using either the source or drain and the gate, its high functionality can reduce the number of transistors in critical path. In continue, we describe our proposed architecture.

Proposed Parallel MAC FIR Filter Architecture

If the phase of the filter is linear, the symmetrical architecture can be used to reduce the multiplier operation. Comparing Figure 1(a) and Figure 1(b), the number of multipliers can be reduced to half after adopting the symmetrical architecture. But number of adders remains constant and it is the basic model to develop the proposed architecture. We implement a low power linear FIR filter that is shown in Figure 2. To decrease power consumption and hardware circuit, Fir filter based on the MAC has been used. Because using a single MAC engine degrades the performance of the design significantly, in order to increase the computing speed and sharing of hardware resources we use two parallel MACs that sharing a single adder based on symmetrical properties of linear phase FIR filter. In proposed MAC architecture adder unit indeed is same adder of multiplier without additional new adder. So this adder play two roles in this architecture we’ve shared it for adder of the MAC and adder of the multiplier. Until we can reduce hardware and thus can reduce power consumption. Final block diagram proposed Parallel MAC FIR filter architecture is shown in Figure 2. Description of the proposed multiplier is provided in the following section.

Proposed Multiplier Architecture

Multiplier is one of the most power consumer components in FIR filters. Power consumption of multiplier block is very high in comparison with the other components of the FIR filter and it contributes to power consumption the most. Signed shift/add multiplier has been used with booth encoding to decrease the number of shift and addition operations, also number of partial products implementation of shifter and booth encoder blocks are only based on multiplexers (2-to-1) which are only a combinational circuits. In next section, a description of each cell will be provided.

Booth encoder multiplier based

A multiplier has two stages. In the first stage, the partial products are generated by the booth encoder and the partial product generator (PPG), and are summed by compressors. In the second stage, the two final products are added to form the final product through a final adder. It employs a booth encoder block, compression blocks, and an adder block. X and Y are the input buffers. Y is the multiplier which is recoded by the booth encoder and X is the multiplicand. PPG module and compressor form the major part of the multiplier. Carry propagation adder (CPA) is the final adder used to merge the sum and carry vector from the compressor module. For radix-4 recoding, the popular algorithm is parallel recoding or Modified Booth recoding. In parallel radix-4 recoding, Y becomes:

\[ Y = \sum_{i=0}^{i=2^i-1} (-2y_{3i+1} + y_{3i} + y_{2i+1}) \cdot 2^i \]

That truth table has been shown in Table I.

Table I. Truth table for Booth encoding

<table>
<thead>
<tr>
<th>Y_{2i+1}</th>
<th>Y_{2i}</th>
<th>Y_{2i-1}</th>
<th>Booth op.</th>
<th>Dir.</th>
<th>Shl.</th>
<th>Add.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>0 x</td>
<td>0 0</td>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>1 x</td>
<td>0 0</td>
<td>0 0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>2 x</td>
<td>0 0</td>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>-2 x</td>
<td>1 1</td>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 1</td>
<td>-1 x</td>
<td>1 1</td>
<td>0 0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>-1 x</td>
<td>1 1</td>
<td>0 0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 1</td>
<td>-0 x</td>
<td>1 0</td>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In our design, we described Booth function as three basic operations, which they called ‘direction’, ‘shift’, and ‘addition’ operation.

Direction determined whether the multiplicand was positive or negative, shift explained whether the multiplication operation involved shifting or not and addition meant whether the multiplicand was added to partial products. The expressions for Booth encoding were stated below as:

Direction, \( D_{2i} = Y_{2i+1} \)

Shift, \( S_{2i} = Y_{2i-1}(Y_{3i} \oplus Y_{3i}) + Y_{2i-1}(Y_{3i} \oplus Y_{3i}) = Y_{3i} \oplus Y_{3i} \)

Addition, \( A_{2i} = Y_{2i+1} \oplus Y_{2i} \)

The Booth encoder was implemented using two XOR gates and the selector using 3MUXes and an inverter careful optimization of the partial-product generation can lead to some substantial delay and hardware reduction [9]. In the normal 8*8 multiplication 8 partial products need to be generated and accumulated. For accumulation seven adders to reduce power are required but in the case of booth multiplier only 4 partial products are required to be generated and for accumulation three adders, reduced delay required to compute partial sum and reduces the power consumption. Figure 3 shows booth encoder.
Shifter block is made only based on multiplexer. The input is an 8-bit vector. The output is a shifted version of the input. It is a shift from 0 to 7 proportional to the weight of select bits (sel[2..0]). If the value of select bits is 0, we have non-shift otherwise we have shift proportional to the weight of select bits. The circuit consists of three individual barrel shifters. Notice that the first barrel has only one '0' connected to one of the multiplexers while the second has two, and the third has four. For larger vectors, we would just keep doubling the number of inputs. If shift ‘001’, for example, then only the first barrel should cause a shift. On the other hand, if shift ‘111’, then all barrels should cause a shift. Figure 4 shown a unsigned shifter block.

Proposed signed shifter block

Shifter block is made only based on multiplexer. The input is an 8-bit vector. The output is a shifted version of the input. It shift input to left from 0 to 7 proportional to the weight of select bits (sel[2..0]). If the value of select bits is 0, we have non-shift otherwise we have shift proportional to the weight of select bits. The circuit consists of three individual barrel shifters. Notice that the first barrel has only one '0' connected to one of the multiplexers while the second has two, and the third has four. For larger vectors, we would just keep doubling the number of inputs. If shift ‘001’, for example, then only the first barrel should cause a shift. On the other hand, if shift ‘111’, then all barrels should cause a shift. Figure 4 shown an unsigned shifter block.

Proposed signed shifter block

Shifter block is made only based on multiplexer. The input is an 8-bit vector. The output is a shifted version of the input. It shift input to left from 0 to 7 proportional to the weight of select bits (sel[2..0]). If the value of select bits is 0, we have non-shift otherwise we have shift proportional to the weight of select bits. The circuit consists of three individual barrel shifters. Notice that the first barrel has only one '0' connected to one of the multiplexers while the second has two, and the third has four. For larger vectors, we would just keep doubling the number of inputs. If shift ‘001’, for example, then only the first barrel should cause a shift. On the other hand, if shift ‘111’, then all barrels should cause a shift. Figure 4 shown a unsigned shifter block.

Proposed signed shifter block

Shifter block is made only based on multiplexer. The input is an 8-bit vector. The output is a shifted version of the input. It shift input to left from 0 to 7 proportional to the weight of select bits (sel[2..0]). If the value of select bits is 0, we have non-shift otherwise we have shift proportional to the weight of select bits. The circuit consists of three individual barrel shifters. Notice that the first barrel has only one '0' connected to one of the multiplexers while the second has two, and the third has four. For larger vectors, we would just keep doubling the number of inputs. If shift ‘001’, for example, then only the first barrel should cause a shift. On the other hand, if shift ‘111’, then all barrels should cause a shift. Figure 4 shown a unsigned shifter block.

Proposed signed shifter block

Shifter block is made only based on multiplexer. The input is an 8-bit vector. The output is a shifted version of the input. It shift input to left from 0 to 7 proportional to the weight of select bits (sel[2..0]). If the value of select bits is 0, we have non-shift otherwise we have shift proportional to the weight of select bits. The circuit consists of three individual barrel shifters. Notice that the first barrel has only one '0' connected to one of the multiplexers while the second has two, and the third has four. For larger vectors, we would just keep doubling the number of inputs. If shift ‘001’, for example, then only the first barrel should cause a shift. On the other hand, if shift ‘111’, then all barrels should cause a shift. Figure 4 shown an unsigned shifter block.

Proposed signed shifter block

Shifter block is made only based on multiplexer. The input is an 8-bit vector. The output is a shifted version of the input. It shift input to left from 0 to 7 proportional to the weight of select bits (sel[2..0]). If the value of select bits is 0, we have non-shift otherwise we have shift proportional to the weight of select bits. The circuit consists of three individual barrel shifters. Notice that the first barrel has only one '0' connected to one of the multiplexers while the second has two, and the third has four. For larger vectors, we would just keep doubling the number of inputs. If shift ‘001’, for example, then only the first barrel should cause a shift. On the other hand, if shift ‘111’, then all barrels should cause a shift. Figure 4 shown an unsigned shifter block.

Proposed signed shifter block

Shifter block is made only based on multiplexer. The input is an 8-bit vector. The output is a shifted version of the input. It shift input to left from 0 to 7 proportional to the weight of select bits (sel[2..0]). If the value of select bits is 0, we have non-shift otherwise we have shift proportional to the weight of select bits. The circuit consists of three individual barrel shifters. Notice that the first barrel has only one '0' connected to one of the multiplexers while the second has two, and the third has four. For larger vectors, we would just keep doubling the number of inputs. If shift ‘001’, for example, then only the first barrel should cause a shift. On the other hand, if shift ‘111’, then all barrels should cause a shift. Figure 4 shown an unsigned shifter block.

Proposed signed shifter block

Shifter block is made only based on multiplexer. The input is an 8-bit vector. The output is a shifted version of the input. It shift input to left from 0 to 7 proportional to the weight of select bits (sel[2..0]). If the value of select bits is 0, we have non-shift otherwise we have shift proportional to the weight of select bits. The circuit consists of three individual barrel shifters. Notice that the first barrel has only one '0' connected to one of the multiplexers while the second has two, and the third has four. For larger vectors, we would just keep doubling the number of inputs. If shift ‘001’, for example, then only the first barrel should cause a shift. On the other hand, if shift ‘111’, then all barrels should cause a shift. Figure 4 shown an unsigned shifter block.
Figure 7: waveform of the results of multiplying two numbers A and B in Quartus software

By implementation of signed shifter block and booth encoder block based on 2-to-1 multiplexer also, 2-to-1 multiplexer in the transistor-level is implemented with two transistors thus multiplication complexity, hardware, and power consumption of FIR filter is reduced. Proposed low power digital FIR filter based on new signed shift/add multiplier shown in Figure 8.

Figure 8: Proposed ultra low power digital FIR filter based on new signed shift/add multiplier for 5 bits

Transistor Level Implementation of FIR Filter

One of the most important issues in VLSI design is power consumption. With the continuously increasing chips’ complexity and number of transistors in a chip, circuits’ power consumption is growing as well. Higher power consumption, raises chips’ temperature and directly affect battery life in portable devices as it causes more current to be withdrawn from the power supply. High temperature afflict circuit operation and reliability so requires more complicated cooling and packaging techniques [10]. The main objective of this paper is to provide new low power solutions for VLSI design of digital FIR filter.

Especially, this work focuses on the reduction of the power dissipation, which is showing an ever-increasing growth with the scaling down of the technologies. Various techniques at the different levels of the design process have been implemented to reduce the power dissipation at the circuit, architectural and system level and transistor level. In continue, we description low power transistor level of each block, that is used in proposed architecture.

Implementation Of Dynamic D-Flip-Flop In Transistor Level

We, now, introduce a new dynamic D-flip-flop eliminating glitches and reducing the number of transistors. It is based on a ratioed logic design technique and transistor merging. Transistor merging is to reduce the number of transistors and thereby save both power and silicon area while suppressing glitch occurrences. Pull-up and pull-down transistors are combined together yielding a circuit having fewer pull-up and pull-down transistors. Where the nodes n1 and Y2 have the same potential of VDD during clk=0. When clk=1, its operation is independent of the n1 level and Y2 may stay high or discharge to low. This observation leads to merge two pull-up transistors of the conventional design. In Figure. 9(a), with clk = 0, Y2 is always high turning on MN3 and causing charge sharing between N1 and Qb resulting in an incorrect value at Qb. It is effective to introduce a transistor MNS2 driven by clk between MP2 and MN3. Figure 9(b) shows the proposed circuit, where MNS2 blocks charge sharing between Qb and N2.

\[
\text{(a) Toggle-flip-flop using the transistor merging technique}
\]

\[
\text{(b) Charge sharing protection between Qb and n1 using MNS2}
\]

Figure 9: Toggle-flip-flop designed with transistor merging technique

Figure 10 shows the proposed D-flip-flop comprising nine transistors, which is free from glitches induced by charge sharing. The MPS2 transistor driven by clk as shown in Figure 10 can effectively reduce charge sharing MPS2 disturbs charge distribution path between Y1 and Y2 preventing MN2 from turning on. This guarantees the correct edge-triggering operation of the flip-flop and enhances its reliability. Unfortunately, the critical path to pull up Y1 node is longer and thereby some speed
degradation is expected. The operation of the proposed D-flip-flop shown in Figure 10 is as follows.

![Figure 10: D-flip-flop for glitch elimination](image)

Consider the circuit of Figure 11(a), where nodes Y1, N1, and Y2 are precharged high with clk = 0 and D = 0. During this phase, MNS1 and MP2 are off, and Qb holds the previous value. Note that both n2 and n3 are weak high because of Y1 and Y2 being high. Assuming that clk changes low to high, MPS1 and MPS2 are turned off and MNS1 and MNS2 are on. Since Y2 cannot discharge instantly, a pull-down path is formed consisting of MNS2, MN3, and MNS1. But N2 and N3 keep weak high from the previous phase resulting in a small glitch due to the voltage drop of Qb. As Y2 becomes low through MN2 and MNS1 path, Qb rises high.

Considering that clk = 0 and D = 1 as shown in Figure 11(b), Y2 is precharged to high but Y1 is low. This makes MN2 be turned off. If clk changes low to high, Qb discharges low through the path consisting of MNS2, MN3, and MNS1. If we change D to low when clk = 1, MP1 is turned on, but the charge sharing between Y1 and Y2 never occurs due to the blocking transistor MPS2. This implies that MN2 remains off and the pull-down path of node Y2 does not exist [11].

![Figure 11: Operations and signal paths of the Dynamic D-flip-flop](image)

- Implementation of Low Power Energy Recovery Full Adder Cell In Transistor Level

As in [12] explained an initial step toward designing low power arithmetic circuit modules, we designed a Static Energy Recovery Full Adder (SERF) cell module illustrated in Figure 12.

![Figure 12: Static Energy-Recovery Full (SERF) Adder](image)

The cell uses only 10 transistors and it does not need inverted inputs. The design was inspired by the XNOR gate full adder design. In non-energy recovery design the charge applied to the load capacitance during logic level high is drained to ground during the logic level low. It should be noted that the new SERF adder has no direct path to the ground. The elimination of a path to the ground reduces power consumption, removing the Psc variable (product of Isc and voltage) from the total power equation. The charge stored at the load capacitance is reapplied to the control gates. The combination of not having a direct path to ground and the reapplication of the load charge to the control gate makes the energy-recovering full adder an energy efficient design. To the best of our knowledge this new design has the lowest transistor count for the complete realization of a full adder. The new SERF design needs only 10 transistors for the adder circuit realization obviously it is the most area efficient design. Thus this adder have minimum number of transistor.

- Implementation of 2 To 1 Multiplexer In Transistor Level

We used pass transistor logic for implementation of multiplexers and XOR gates. Multiplexers are implemented with two transistors PTL efficiently decreases power consumption and area of total system. An NMOS or PMOS pass-transistor, or a CMOS transmission gate, can be used to steer or transfer change from one node of a circuit to another node, under the control of the FETs gate voltage. Pass transistor chains are used in designing regular arrays, such as ROMs, PLAs, and multiplexers. When used in regular arrays, depletion-mode pass transistor created by an ion implant step can be used to remove control from a given FET by short-circuiting its drain to its source. Thus, both enhancement-mode and depletion-mode devices can appear in pass-transistor chains. Pass transistors have several advantages over inverters two major advantages of pass transistor over standard NMOS gate logic are:

- They are not ratioed devices and can be minimum geometry.
- They do not have a path from plus supply to ground, and do not dissipate stand by power.

If the gate and drain of a pass transistor are both high, the source will rise to the lower of the two potentials VDD and Vgs – Vth. If the gate and drain are both at VDD, the source can only rise to one threshold voltage below the gate.
Table I. Truth table for Booth encoding

<table>
<thead>
<tr>
<th>$Y_{2b-1}$</th>
<th>$Y_{2b-1}$</th>
<th>Booth op.</th>
<th>Dir.</th>
<th>Shit.</th>
<th>Add.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>0x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>1x</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>0 1 0</td>
<td>1x</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>0 1 1</td>
<td>2x</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1 0 0</td>
<td>-2x</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1 0 1</td>
<td>-1x</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>1 1 0</td>
<td>-1x</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>1 1 1</td>
<td>-0x</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table II: Summary of data obtained from power consumption of our proposed method and other works

<table>
<thead>
<tr>
<th>FIR filter</th>
<th>Power(mW)</th>
<th>Frequency(MHz)</th>
<th>Process</th>
<th>Tap</th>
<th>Supply voltage</th>
<th>Number of bit multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>[2]</td>
<td>7.48</td>
<td>100</td>
<td>0.18 µm</td>
<td>73</td>
<td>1.62v</td>
<td></td>
</tr>
<tr>
<td>[14]</td>
<td>75</td>
<td>20</td>
<td>0.6 µm</td>
<td>32</td>
<td>3.3v</td>
<td></td>
</tr>
<tr>
<td>[3]</td>
<td>16.5</td>
<td>86</td>
<td>0.35 µm</td>
<td>8-digit</td>
<td>2.5v</td>
<td>8*8</td>
</tr>
<tr>
<td>[15]</td>
<td>238.8</td>
<td>100</td>
<td>0.25 µm</td>
<td>10</td>
<td>2.5v</td>
<td>----</td>
</tr>
<tr>
<td>[4]</td>
<td>1.5471</td>
<td>250</td>
<td>----</td>
<td>MAC</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>[5]</td>
<td>367.6</td>
<td>----</td>
<td>----</td>
<td>33</td>
<td>4v</td>
<td>----</td>
</tr>
<tr>
<td>[16]</td>
<td>146.7</td>
<td>10</td>
<td>0.5 µm</td>
<td>8</td>
<td>----</td>
<td>10*10</td>
</tr>
<tr>
<td>[6]</td>
<td>1.363</td>
<td>20</td>
<td>0.18 µm</td>
<td>4</td>
<td>2.5v</td>
<td>8*8</td>
</tr>
<tr>
<td>[17]</td>
<td>104.3</td>
<td>20</td>
<td>2 µm</td>
<td>----</td>
<td>3.2v</td>
<td></td>
</tr>
<tr>
<td>[18]</td>
<td>340.98</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>[19]</td>
<td>61.8</td>
<td>----</td>
<td>70nm</td>
<td>25</td>
<td>0.8v</td>
<td>----</td>
</tr>
<tr>
<td>[18]</td>
<td>315uw/Mhz</td>
<td>----</td>
<td>----</td>
<td>120</td>
<td>----</td>
<td>16*16</td>
</tr>
<tr>
<td>[20]</td>
<td>0.2883</td>
<td>44.1kHz</td>
<td>0.25 µm</td>
<td>48</td>
<td>2.5v</td>
<td>16*16</td>
</tr>
<tr>
<td>[8]</td>
<td>0.9487</td>
<td>----</td>
<td>0.18 µm</td>
<td>20</td>
<td>----</td>
<td>8*8</td>
</tr>
</tbody>
</table>

Proposed method | 55uw | 100 | 0.18 µm | MAC | 1.8v | 8*8 |

If the source tries to rise higher, the device cuts off. If the gate is at least a threshold voltage higher than the drain, the source will rise to within a few millivolts of the drain potential. Charge sharing is a serious problem which occurs when two or more capacitors at different potentials are tied together. A node of a network must never be driven simultaneously by signals of opposite polarity, as this can leave the node in an erroneous or undefined state. One must beware of sneak paths which allow charge to leak. Pass transistors are bilateral, and charge can flow from output to input also. This is not a problem if all the inputs are designed to connect to the output via mutually disjoint paths. A sneak path is created when two pass transistors are both on at the same time and one is connected to $V_{DD}$ while the other is connected to GND. Pass transistors are usually designed to be of minimum size, $2\lambda^*2\lambda$. If further, the two devices have the same gate-to-source bias, their on resistances will be approximately equal and the output voltage will be about $V_{DD}/2$. Figure 13 shows transistor level of 2 to 1 multiplexer and XOR gate.

Finally Figure 14 shows Proposed transistor level for new signed shift/add block based on pass transistor technique.

![Figure 13: Transistor level implementation of 2 to 1 multiplexer (a) and XOR gate (b).](image1.png)

![Figure 14: Proposed transistor level for new signed shift/add block based on pass transistor technique.](image2.png)
Filters, a conclusion is a ultra low power digital FIR filter based new architecture of the shift/add multiplier by PTL be designed and has power consumption less than other works.

HSPICE simulation results demonstrate that proposed method in implementation of multiplexers (2 to 1) and energy recovery full adder cell, and pass transistor logic is used charge free, implementation of full adder is based on low power flop we use utilization dynamic D-flip-flop aiming at glitch and circuits with combinational logic, for implementation of D-flip-flop we use utilization dynamic D-flip-flop reducing number of shift operation and partial encoder, new signed shifter block, adder block. In this design signed shift/add multiplier with three blocks including booth encoder, new signed shifter block, adder block. In this design Booth encoder reducing number of shift operation and partial product. Blocks of booth encoder and new signed shifter are complete based on 2 to 1 multiplexer these blocks are complete circuits with combinational logic, for implementation of D-flip-flop we use utilization dynamic D-flip-flop aiming at glitch and charge free, implementation of full adder is based on low power energy recovery full adder cell, and pass transistor logic is used in implementation of multiplexers (2 to 1) and XOR gates. The conclusion is a ultra low power digital FIR filter based new architecture of the shift/add multiplier by PTL be designed and implement with 0.18µm technology at 1.8v supply voltage. Table II show the power consumption of our proposed method and other works. The result show that the proposed circuit has power consumption less than recently digital FIR filters in the literature.

Conclusion
In this paper, a new low power digital linear phase FIR filter based on new signed shift/add multiplier is proposed. New signed shift/add multiplier with three blocks including booth encoder, new signed shifter block, adder block. In this design Booth encoder reducing number of shift operation and partial product. Blocks of booth encoder and new signed shifter are complete based on 2 to 1 multiplexer these blocks are complete circuits with combinational logic, for implementation of D-flip-flop we use utilization dynamic D-flip-flop aiming at glitch and charge free, implementation of full adder is based on low power energy recovery full adder cell, and pass transistor logic is used in implementation of multiplexers (2 to 1) and XOR gates. The conclusion is a ultra low power digital FIR filter based new architecture of the shift/add multiplier by PTL be designed and implement with 0.18µm technology at 1.8v supply voltage an HSPICE simulation results demonstrate that proposed method has power consumption less than other works.

Reference
[12] R. Shalem1, E. John2 and L. K. John1, “A Novel Low Power Energy Recovery Full Adder Cell”, research is supported by the National Science Foundation.