Multiplier Accumulator

Project Specifications:

A. Names of all team members:

SAHIL LUTHRA and PIPAT SONGKAITISAK

B. Title of the project:

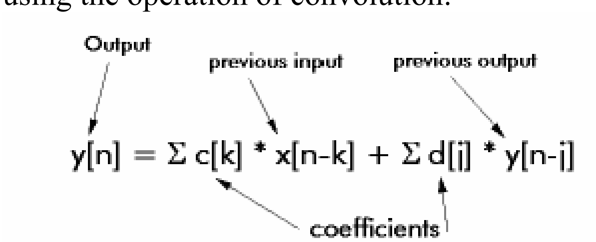
MULTIPLIER ACCUMULATOR.

C. Introduction and Real-life applications:

Multiplication Accumulation is an important part of real-time digital signal processing (DSP) applications ranging from digital filtering to image processing. The motivation for our project is the FIR filter which finds its applications in the field of digital signal processing in the fields of speech, audio, image, and video processing, pulse shaping, correlation and equalization etc.

Filtering is a process of selecting, or suppressing, certain frequency components of a signal.

Output from a digital filter is made up from previous inputs and previous outputs, using the operation of convolution:



Two convolutions are involved: one with the previous inputs, and one with the previous outputs. In each case the convolving function is called the filter coefficient

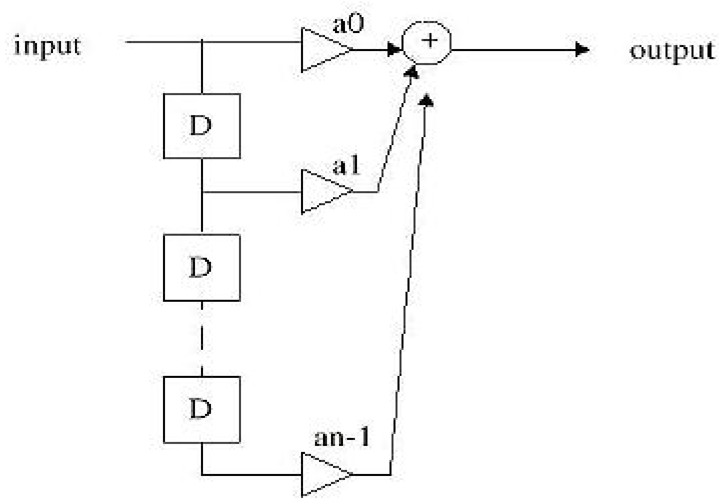
It is much easier to approach the problem of calculating filter coefficient if we simplify the filter equation so that we only have to deal with previous inputs (that is, we exclude the possibility of feedback). The filter equation is then simplified:

$$y[n] = \Sigma\, c[k] * x[n-k] + \Sigma\, d[j] * y[n-j]$$

If such a filter is subjected to an impulse (a signal consisting of one value followed by zeroes) then its output must necessarily become zero after the impulse has run through the summation. So the impulse response of such a filter must necessarily be finite in duration. Such a filter is called a **Finite Impulse Response filter** or **FIR filter**.

Block Diagram for FIR Filter

H(z) = a0 + a1 z^(-1) + a2 z^(-2) + ... + an-1 z^(n-1)



In a FIR context, a "MAC" is the operation of multiplying a coefficient by the corresponding delayed data sample and accumulating the result. FIRs usually require one MAC per tap.
A FIR "tap" is simply a coefficient/delay pair. The number of FIR taps, (often designated as "N") is an indication of 1) the amount of memory required to implement the filter, 2) the number of calculations required, and 3) the amount of "filtering" the filter can do; in effect, more taps means more stopband attenuation, less ripple, narrower filters, etc.)

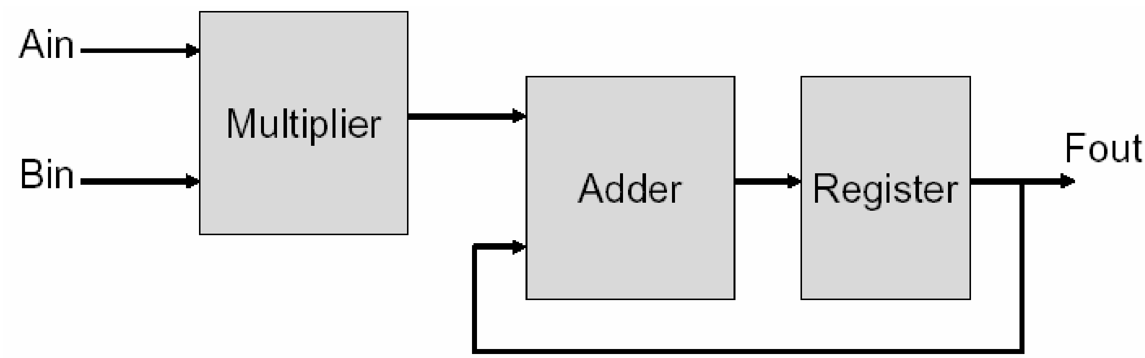
Structurally, FIR filters consist of just two things: a sample delay line and a set of coefficients. To implement the filter:

1. Put the input sample into the delay line.
2. Multiply each sample in the delay line by the corresponding coefficient and accumulate the result.
3. Shift the delay line by one sample to make room for the next input sample.

D. Functional Requirements:

The function of the MAC is given by the following equation:

$$F=\sum_{i=1}^{N} A_i.B_i$$



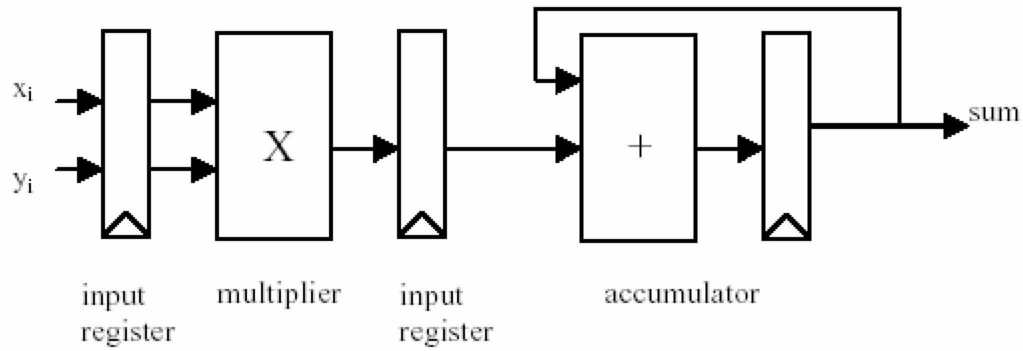
F is the primary output. Different values of Ai and Bi are multiplied together and the result is added to the current result.

D. Types and sizes of all operands:

'Ai' and 'Bi' are 32 bits wide Signed Data inputs. 'N' is the No. of partial products accumulated by the MAC. The maximum value of N is specified as 1024(i.e.2^{10}).So, the size of output F is ($64+\log_2 2^{10}$) =74 bits.

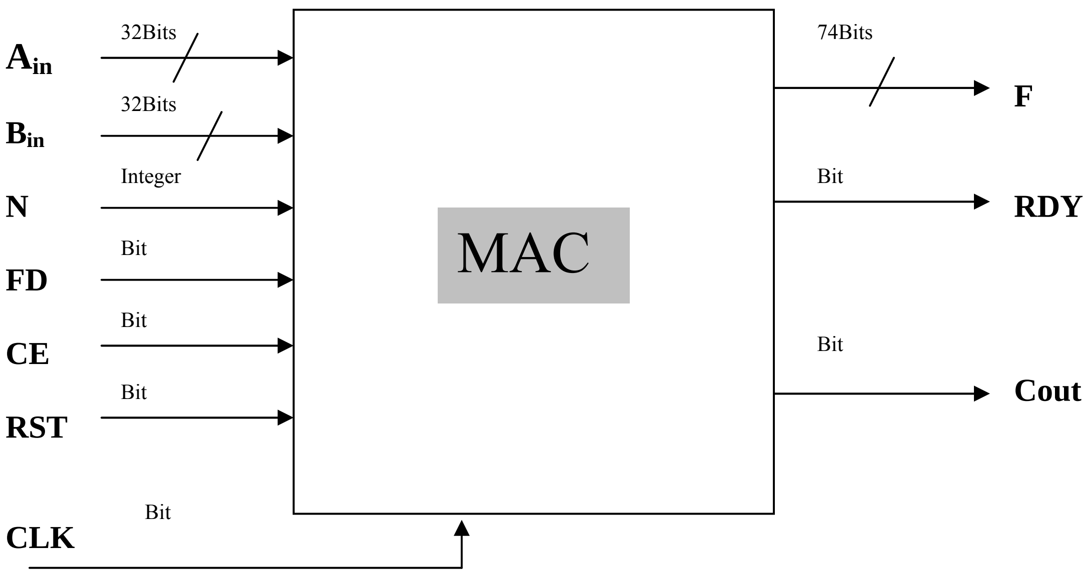
E. Optimization criteria

Since the speed and chip area are important considerations in Microprocessors and various DSP applications, the MAC will be optimized for maximum throughput in a fixed area. We use pipelined architecture to achieve our goal of maximum throughput as shown below.



F. Interface:

Name Of Pin	Mode	Width	Function
A	I/P	32bits	Input A
B	I/P	32bits	Input B
N	I/P	Integer	Input N
RST	I/P	1bit	Control Pin to Reset
CLK	I/P	1bit	CLK Input
CE	I/P	1bit	Control Pin to enable CLK
FD	I/P	1bit	I/P Pin identifies 1st pair of Inputs
RDY	O/P	1bit	Ready Pin to indicate the availability of the Output.
F	O/P	74bits	The final result.
Cout	O/P	1bit	Carry Out



G. Software implementation used to generate test vectors:

We are planning to use C Language to generate test vectors and verify the results. At the moment the algorithm for generation of the test vector has not been decided but most probably random test vectors will be used.

H. Language, Platform, and Tools:

Hardware Description Language: VHDL

Platform : Xilinx FPGA (Spartan 2s200fg256)

HW tools : Aldec Active-HDL, Synplicity Synplify Pro and Xilinx ISE

I. List of references:

1. "The Designer's Guide To VHDL" By Peter J Ashenden.
2. "Computer Arithmetic Algorithms and Hardware Design" By Behrooz Parhami.
3. http://www.bores.com/courses/intro/filters/4_def.htm
4. http://www.xilinx.com/xcell/xl38/xcell38_32.pdf
5. http://www.digitalfilter.com/dfalz/dfafirv.html
6. Dr.Gaj's Lecture notes.
7. Let Us C By Yashwant Kanitkar.