VLSI Design and Implementation of Low Power MAC Unit with Block Enabling Technique

Shanthala S

Asst. Professor, Bangalore Institute of Technology, Bangalore, Research Scholar
EC Research Centre, NMAM Institute of Technology, Nitte-574110, India
E-mail: shanthala_wg@yahoo.com

S. Y. Kulkarni

Principal, NMAM Institute of Technology, Nitte-574110, Karnataka, India
E-mail: sy_kul@yahoo.com

Abstract

In the majority of digital signal processing (DSP) applications the critical operations are the multiplication and accumulation. Real-time signal processing requires high speed and high throughput Multiplier-Accumulator (MAC) unit that consumes low power, which is always a key to achieve a high performance digital signal processing system. The purpose of this work is, design and implementation of a low power MAC unit with block enabling technique to save power. Firstly, a 1-bit MAC unit is designed, with appropriate geometries that gives optimized power, area and delay. The delay in the pipeline stages in the MAC unit is estimated based on which a control unit is designed to control the data flow between the MAC blocks for low power. Similarly, the N-bit MAC unit is designed and controlled for low power using a control logic that enables the pipelined stages at appropriate time. The adder cell designed has advantage of high operational speed, small transistor count and low power. The MAC is implemented on a 0.18um CMOS technology using CADENCE VIRTUOSO tool. This paper also investigates on various architectures of multipliers and adders which are suitable for implementation of high throughput signal processing and at the same time to achieve low power consumption. The whole MAC chip is operated at 125 MHz using 1.8 V power supply. The power is reduced by 27% using the block enabling technique compared to the normal design.

**Keywords:** Low Power, MAC, clock gating, block enable, multiplier.

1. Introduction

In the majority of digital signal processing (DSP) applications the critical operations usually involve many multiplications and/or accumulations. For real-time signal processing, a high speed and high throughput Multiplier-Accumulator (MAC) is always a key to achieve a high performance digital signal processing system. In the last few years, the main consideration of MAC design is to enhance its speed. This is because, speed and throughput rate is always the concern of digital signal processing system. But for the epoch of personal communication, low power design also becomes another main design consideration. This is because, battery energy available for these portable products limits the power consumption of the system. Therefore, the main motivation of this work is to investigate various

pipelined multiplier/accumulator architectures and circuit design techniques which are suitable for implementing high throughput signal processing algorithms and at the same time achieve low power consumption. A conventional MAC unit consists of (fast multiplier) multiplier and an accumulator that contains the sum of the previous consecutive products. The function of the MAC unit is given by the following equation:

$$F = \Sigma \, A_i \, B_i \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (1.1)$$

Figure 1: Basic structure of MAC
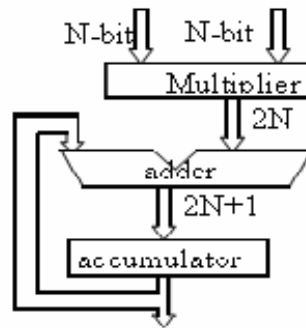
Figure 2: MAC architecture

output
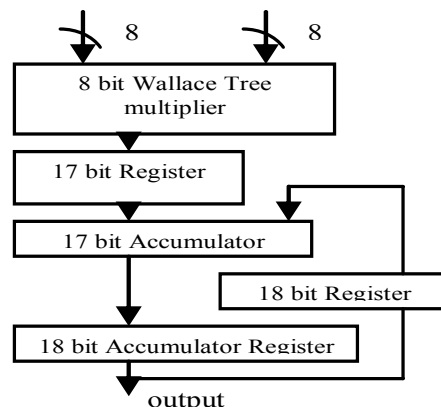
The main goal of a DSP processor design is to enhance the speed of the MAC unit, and at the same time limit the power consumption. In a pipelined MAC circuit, the delay of pipeline stage is the delay of a 1-bit full adder (Jou, Chen, Yang and Su, 1995) . Estimating this delay will assist in identifying the overall delay of the pipelined MAC. In this work, 1-bit full adder is designed. Area, power and delay are calculated for the full adder, based on which the pipelined MAC unit is designed for low power.

2. Multiplier and Accumulator Unit

MAC is composed of an adder, multiplier and an accumulator. Usually adders implemented are Carry-Select or Carry-Save adders, as speed is of utmost importance in DSP (Chandrakasan, Sheng, & Brodersen, 1992 and Weste & Harris, 3rd Ed). One implementation of the multiplier could be as a parallel array multiplier. The inputs for the MAC are to be fetched from memory location and fed to the multiplier block of the MAC, which will perform multiplication and give the result to adder which will accumulate the result and then will store the result into a memory location. This entire process is to be achieved in a single clock cycle (Weste & Harris, 3rd Ed). Figure 2 is the architecture of the MAC unit which had been designed in this work. The design consists of one 17 bit register, one 8-bit Wallace

tree multiplier, 17-bit accumulator using ripple carry and two18-bit accumulator registers. To multiply the values of A and B, Wallace tree multiplier is used instead of conventional multiplier because Wallace tree multiplier can increase the MAC unit design speed. Ripple Carry Adder (RCA) is used as an accumulator in this design. Apparently, together with the utilization of Wallace tree multiplier approach, carry save adder in the final stage of the Wallace tree multiplier and Ripple Carry adder as the accumulator, this MAC unit design is not only reducing the standby power consumption but also can enhance the MAC unit speed so as to gain better system performance. The operation of the designed MAC unit is as in Equation 2.1. The product of Ai X Bi is always fed back into the 17-bit Ripple Carry accumulator and then added again with the next product Ai x Bi. This MAC unit is capable of multiplying and adding with previous product consecutively up to as many as eight times.

Operation: Output $= \Sigma$ Ai Bi   (2.1)

In this paper, the design of 8x8 multiplier unit is carried out that can perform accumulation on 17 bit number. This MAC unit has 18 bit output and its operation is to add repeatedly the multiplication results. The total design area is also being inspected by observing the total count of transistors. Power delay product is calculated by multiplying the power consumption result with the time delay.

2.1. Wallace tree Multiplier

The design analysis starts with the analysis of elementary algorithm for multiplication by Wallace tree multiplier. Figure 3 shows the algorithm for 8 x 8 bits multiplication performed by Wallace tree multiplier. There are five stages to go through, to complete the multiplication process (Weste & Harris, 3rd Ed). Each stage used half adders and full adders that are denoted by the red circle for the 1bit half adder and the blue circle for the 1 bit full adder. Firstly, we had to reduce the partial products using the

Figure 3: Algorithm for 8 bits x 8 bits Wallace tree multiplier (Harun, 2007)
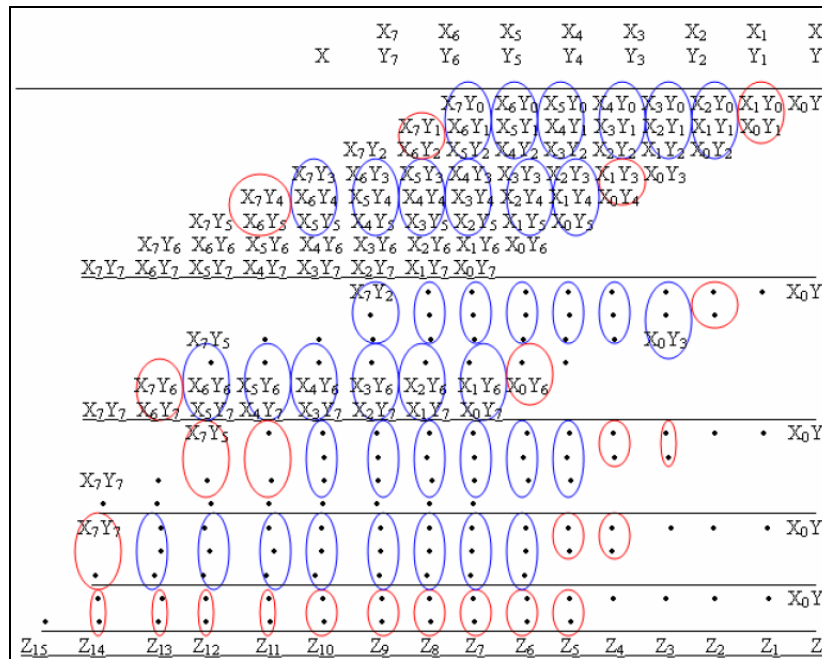
half adders and full adders that are combined to build a carry save adder (CSA) until there were just two rows of partial products left. Next, we add remaining two rows by using a fast carry propagate adder. In this project, CSA (carry save adder) using ripple carry adder is used to get the final product. Secondly, the schematic of the conventional 8 bits x 8bits high speed Wallace tree multiplier is designed by referring to the algorithm.

2.2. Carry Save Adder

When three or more operands are to be added simultaneously using two operand adders, the time consuming carry propagation must be repeated several times. If the number of operands is 'k', then carries have to propagate (k-1) times (Weste & Harris, 3rd Ed). In the carry save addition, we let the carry propagate only in the last step, while in all the other steps we generate the partial sum and sequence of carries separately. A CSA is capable of reducing the number of operands to be added from 3 to 2 without any carry propagation. A CSA can be implemented in different ways. In the simplest implementation, the basic element of carry save adder is the combination of two half adders or 1 bit full adder(Weste & Harris, 3rd Ed).

2.3. Block Enabling Technique

In any MAC unit, data flows from the input register to the output register through multiple stages such as, multiplier stage, adder stage and the accumulator stage as shown in figure 4. Within the multiplier stage, further we find that there are multiple stages of addition. During each operation of multiplication and addition, the blocks in the pipeline may not be required to be on or enabled until the actual data gets in from the previous stage. In block enabling technique, we find the delay of each stage. Every block gets enabled only after the expected delay. For the entire duration until the inputs are available, the successive blocks are disabled, thus saving power. In the next section, we design a 1-bit MAC unit with pipeline structure and find the power consumption.

Figure 4: General Block Diagram of a Pipeline MAC with block enabling Technique cs - control signal
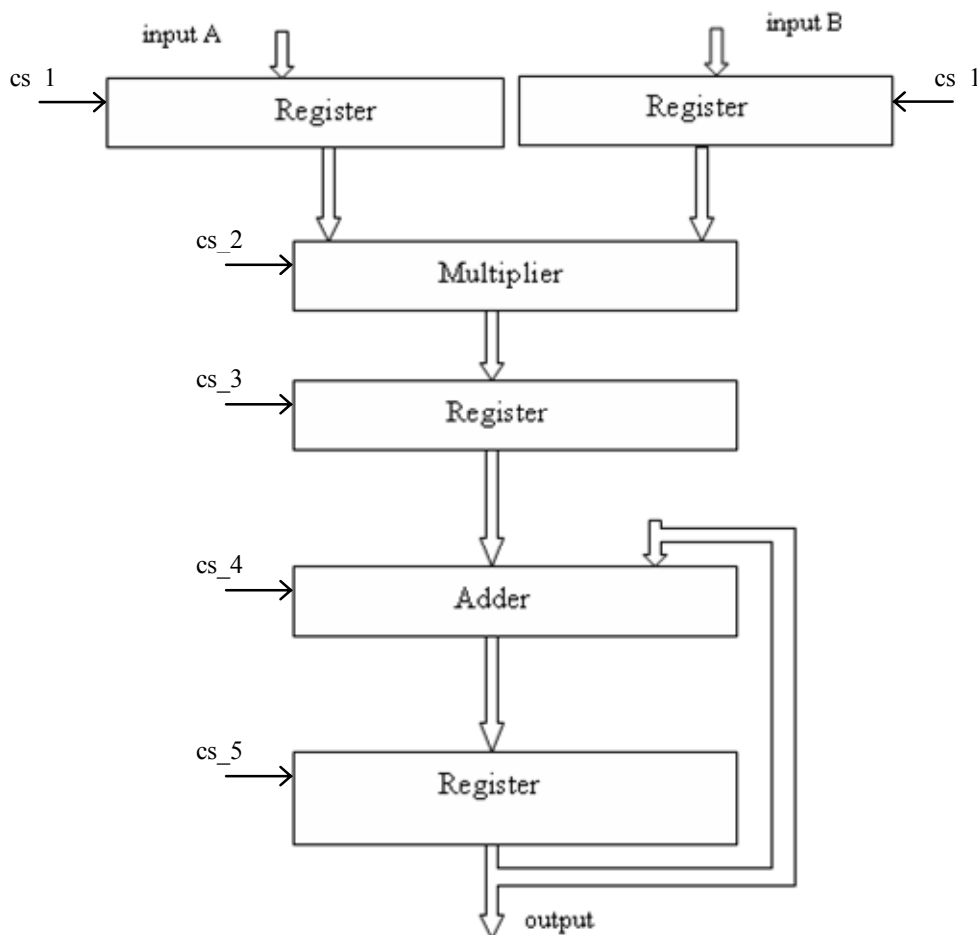
2.4. Pipelined block enabled logic

Figure 5 shows a three stage pipelined MAC with block enable logic. In this logic, depending upon the delay of individual blocks, the control logic enables the clock, power and logic pins of the block, thus saving power. Figure 6 shows the block schematic of the 1 bit full adder circuit with enable. Each of the blocks in the MAC unit has an enable signal to save power.

Figure 5: MAC with control logic
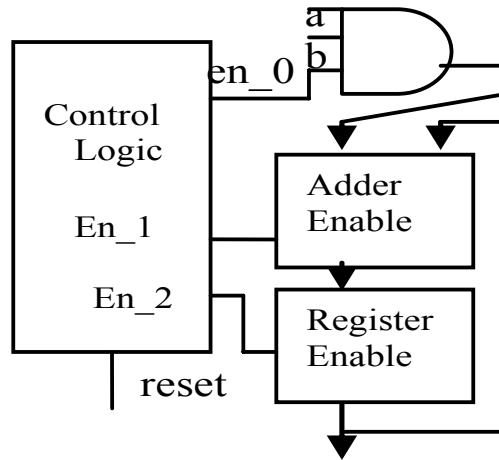
Figure 6: 1bit full adder with enable
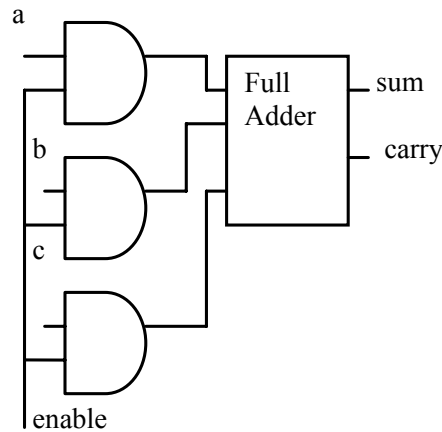
2.5. Accumulator Register

Figure 7 shows the 1 bit register file cell that may be represented by a D-flip flop and two gates. Note that in addition to the clock signal, the cell has 3 inputs and 1 output: write select, read select and D input and Q output signal. With in this cell, the D-flip flop will store the value of the input signal whenever write select is equal to 1, consequently, whenever the read select signal is equal to 1, this D-flip flop will pass its stored value to the output through a tristate buffer.

Figure 7: 1 bit Register cell
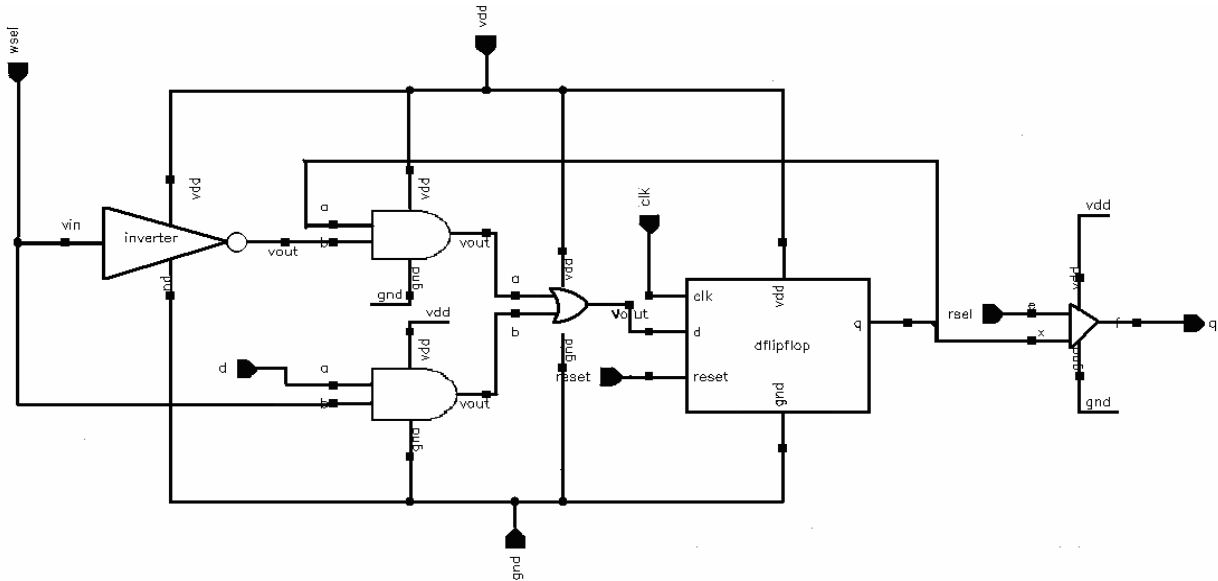
From the observations made, we find that the basic building blocks for any MAC unit are Multiplier, Adder and Register. Multiplier and adder blocks require full adders, and registers require flip-flops or latches. The objective of this work is to find the total area, power and delay of the MAC unit that forms the critical part of any DSP application. At the micro level, the power, delay and area for the basic blocks are calculated based on the experimental setup. Based on the results obtained, the reasons for power and delay are identified at the micro level and remedies are taken to minimize this power. Further this power reduction technique is extended at the macro level. In our design, it is the MAC architecture. Section 3 discusses these results.

2.6. Full adder design

Different ways of realizing the full adder (Jou, Chen,Yang & Su, 1997, Suzuki, Ohkubo, Shinbo, Yamanaka, Shimizu, Sasaki & Nakagome, 1993, Lu & Samulei, 1993) are tabulated in table 1 and the results of the same are also compared. From the table it is very clear that the mux based full adder implementation consumes very less power and also has minimum delay. In this work, the mux based full adder is considered for implementation. The mux based full adder has a delay of 0.0012ns, this implies that, when the input is applied it takes 0.0012ns to produce the outputs. Hence we can disable other blocks connected to the output of full adder, and hence power is saved. Using this design, 1-bit pipelined MAC unit is realized. The basic building blocks for the MAC units are the flip flop to store 1-bit data, 1-bit adder, AND gate for control activity. These basic building blocks are taken independently and analyzed for delay and power. The pipelined MAC is incorporated with an enable pin to reduce power consumption, i.e. at any given point of time only one of the blocks gets enabled to ensure data flow from one stage to the next stage. For example, if the adder block is computing, the register block is disabled to save power or during the loading operation, adder block is disabled to save power. This is controlled by an external signal E, which enables or disables the corresponding block to keep it idle.

Low power techniques as discussed in (Anantha, Samuel & Borderson, 1992) are considered in this work for reducing power.

Table 1:   Full adder design comparison

Full adder using	No. of transistors	Area (um^2)	Power (uw)	Delay (ns)
Only nand	36	507.592	0.01293	0.00987
Only mux	22	324.225	0.0001459	0.0012
Exor, and, or	30	408.127	3.58	0.0065
Conventional cmos logic	28	387.548	0.0328	0.01055
Quasi domino	23	375.124	0.01645	0.00767
Static and dynamic	22	367.721	46.65	0.0109
Exor & AND	30	413.402	3.58	0.0065

2.7. AND Gate

The basic gate that is required to enable or disable the MAC blocks is controlled using an AND gate. The results tabulated in table 2 are by conducting experiments using cadence tools with 180nm technology library. The width of the transistors is varied to find the effect of delay and power. Table 3 lists the effect of width variation on power. It is observed that the delay of AND gate is not constant and it varies as per the input signals. From the table 2, we observe that delay reduces with increase in width, we select 0.4 as the AND gate geometries that gives minimum delay. As the AND gate has delay, the blocks connected to the output of AND gate are disabled until this time, and these blocks are enabled only after the outputs are available, hence saving power. From table 3 we find that the power also varies with input, and the power is maximum for 0.4 geometries.

Table 2:   Delay variations for AND gate

Wn / Wp	Delay 't$_d$' (S)	Delay 't$_d$' (S)
	i/p a = pulse, b = 1	i/p a = 1, b = pulse
0.2	2.187 E-10	2.156 E-10
0.3	2.225 E-10	2.192 E-10
0.4	1.805 E-10	1.776 E-10
0.5	1.745 E-10	1.720 E-10

Table 3:   Power variations for AND gate

Wn / Wp	Power (W)	Power (W)
	i/p a = pulse, b = 1	i/p a = 1, b = pulse
0.2	4.411 E-11	1.275 E-08
0.3	4.457 E-11	2.106E-08
0.4	5.439 E-11	8.919E-09
0.5	2.211 E-11	1.301E-08

2.8.1. Bit Register

Register forms one of the basic unit for the MAC unit, as the register stores data, there is possibility of leakage current and that affects power dissipation. Also the clock connected to the register cell also keeps changing and hence affects the dynamic power dissipation. In this work, the basic register cell is analyzed for its power consumption. The register cell is enabled with clock gating and the power and delay is calculated. Table 4 shows the power and delay of the basic register cell calculated using cadence tools. We find that the power gets reduced with enable. Knowing the delay, we enable the blocks connected at the output of the 1 bit register only after the output is available. This helps in saving power.

Table 4:   Power and delay results for 1-Bit register cell

Wn / Wp	Power (W)		Delay 't$_d$'(S)	
	Data i/p d$_i$ = pulse		Data i/p d$_i$ = pulse	
	With enable	**Without enable**	**With enable**	**Without enable**
0.2	4.029 E-09	4.078 E-09	7.881 E-10	7.882 E-10
0.3	2.485 E-08	3.628 E-09	6.996 E-10	6.985 E-10
0.4	3.684 E-09	3.712 E-09	6.369 E-10	6.371 E-10
0.5	4.636 E-09	4.644 E-09	6.167 E-10	6.152 E-10

2.9.1. Bit Full Adder

Mux based full adder is designed in this work using 180nm technology and the results are obtained using cadence tools. Table 5 depicts the results for power and table 6 depicts the results for the delay of 1-B full adder. We find that the power increases with increase in width and also suddenly reduces. This is due to the fact that as we increase the width ratios of the transistors, due to mobility variations, threshold variations occur and hence the power reduces. Hence in this work Wn / Wp ratio of 0.3 is chosen for better results. The delay of full adder is 0.39ns and 0.4317ns, the maximum delay is selected and based upon this delay the output blocks connected to the full adder are enabled.

Table 5:   1 Bit Full adder – Power

Wn / Wp	Power (W)	Power (W)
	a = pulse, b = 1, Cin = 0	a = 1, b = pulse, Cin = 0
0.2	3.357 E-10	3.107 E-10
0.3	3.400 E-10	3.797 E-10
0.4	4.430 E-10	4.942 E-10
0.5	3.225 E-10	3.602 E-10

Table 6:   1 Bit Full Adder – Delay

Wn / Wp	Delay 't$_d$'(S)	Delay 't$_d$'(S)
	a = pulse, b = 1, Cin = 0	a = 1, b = pulse, Cin = 0
0.2	4.496 E-10	4.912 E-10
0.3	3.973 E-10	4.317 E-10
0.4	3.663 E-10	3.974 E-10
0.5	3.569 E-10	3.866 E-10

2.10.1. Bit MAC

Using the basic building blocks discussed, 1-bit MAC unit is designed with clock gating and enable pin. From the result analysis carried out using experimental setup, we find that the AND gate delay is 0.225ns, Full Adder delay is 0.4317ns and register delay is 0.6996ns. When the input is applied at 0ns, all the blocks are enabled simultaneously, the FA block would compute the results on unknown data until 0.225ns, and the register block would be receiving unknown data for 0.6567ns and hence there is wastage of power as these data's are not the actual ones. Hence in this work, we have incorporated a control signal that enables the blocks only after the outputs are available at their inputs. Hence we call this technique as block enable technique. Based on the delay of each block, a control signal is generated to enable the blocks.

Tables 7 and 8 depict the power and delay results obtained for the MAC unit with and without enable. From table 7, we find that the power varies with the input. MAC unit with all 0's consume less power than all 1's. The power calculation with enable is found to be more than the power without enable. This is due to the fact that extra control logic added for block enable technique consumes additional power. However, if we neglect the power consumed by the control unit, then due to the

enabling technique we find by hand calculation after getting the power report, the power consumption is reduced by 27% of the actual power.

Table 7: 1 Bit MAC – Power calculations with input variations

	Power (W)		Power (W)	
	i/p =all 0s		i/p = all 1s	
Wn / Wp	With enable	Without enable	With enable	Without enable
0.2	7.718 E-09	4.47 E-10	7.795 E-09	2.571E-08
0.3	7.186 E-09	4.562E-10	7.335 E-09	1.241E-04
0.4	5.873 E-09	6.027 E-10	6.006 E-09	8.979E-10
0.5	7.717 E-09	4.576 E-10	7.812 E-09	6.828E-05

Table 8: **1 Bit MAC – Power and delay with pulse input**

	Power (W)		Delay 't_d'(S)	
	i/p a = pulse		i/p a = pulse	
Wn / Wp	With enable	Without enable	With enable	Without enable
0.2	2.803E-5	4.786E-10	2.922E-8	2.928E-9
0.3	5.263E-6	5.061E-10	2.882E-8	3.235E-9
0.4	2.397E-6	6.488E-10	2.868E-8	3.446E-9
0.5	4.445E-6	4.862E-10	2.863E-8	3.492E-9

After analyzing the basic building blocks at the micro level, we find that appropriate widths of the transistors are very important in deciding the power reduction and delay. In this work we have identified that Wn / Wp can be taken to be 0.3 or 0.4. Using the results, we build the macro blocks like the multiplier block, adder block and the register block for constructing the MAC unit. With the analysis carried out for the 1 bit MAC unit, which is also extended to the macro model, the power and delay analysis is discussed in the next section.

2.11. Multiplier (8 X 8)

Table 9 depicts the power dissipation for the multiplier block discussed in this paper. Power consumption of the multiplier block is calculated for all ones and all zeros. One of the operands of the multiplier is set as constant and the other operand is set to a pulse and the power and delay is calculated for variations in the width of the transistors and the results are tabulated. From table 9, we find that the power consumption is less with the width ratio of 0.3. Further increase in width affects the power due to threshold voltage variations.

Table 10 tabulates the power and delay values for a 18 bit register obtained using cadence tools for 180nm technology by varying the width ratios of the transistors used in the schematic.

Table 9: Multiplier – Power and Delay

	Power (W)	Delay 't_d'(S)
Wn / Wp	i/p = pulse	i/p = pulse
0.2	3.950E-7	9.077E-09
0.3	1.339E-9	5.875E-10
0.4	9.400E-8	5.511E-10
0.5	1.379E-7	5.471E-10

Table 10: 18 Bit Register – Power and Delay

Wn / Wp	Power (W)		Delay 't$_d$'(S)	
	i/p =all 0s	i/p = all 1s	i/p = all 0s	i/p = all 1s
0.2	4.337E-9	5.225E-9	1.0737E-8	6.0737E-8
0.3	4.558E-9	4.992E-9	6.149E-10	4.561E-8
0.4	6.158E-9	1.401E-8	5.552E-10	5.551E-10
0.5	4.789E-9	1.766E-8	5.288E-10	5.287E-10

Table 11: Various Blocks – delay, power, speed and power delay product

| Blocks	Power (watt)	Delay(s)	Speed (Hz)	Power delay product(fj)
1bit full adder	0.145n	0.0012n	833.33G	0.000000174
1bit D-flip flop	0.0596n	0.01425n	70.175G	0.000000849
1 bit register cell	0.2804n	0.0405n	24.69G	0.0000113
2x1 mux	0.00434n	0.00458n	218.34G	0.0000000198
17 bit accumulator	0.0122u	0.04438n	22.53G	0.000541
18 bit accumulator register	0.987n	0.0724n	13.81G	0.000714
8x8 Wallace tree multiplier	0.03324u	0.1152n	8.68G	0.0038
MAC unit	0.007698m	0.437n	2.288G	3.364

3. Results
3.1. Power Consumption

Table 11 shows that total power consumption using TSMC 0.18um is about 0.007698mW for the MAC unit with enable. The delay time is observed from time difference between the rise edges of clock input with the rise edge of the output waveform. It also shows the tabulated result of delay value for each part that was used to design MAC unit using TSMC 0.18um. The delay for the MAC unit is 0.437ns. The design speed is calculated from reciprocal of the delay which means 1/Delay time is equal to speed. Total speed for MAC unit using TSMC 0.18um is 2.288GHz.

3.2. Power-Delay Product

The Power-delay product is simply the product of the power consumption and the time delay. The smaller the value of the power-delay product, the better is the performance of the design. Since this MAC unit has almost negligible power-delay product value, it indeed has a better performance in terms of the speed and power dissipation. Based on table 11, total power delay product for MAC unit using TSMC 0.18um is 3.364fj.

4. Conclusion

A 8x8 multiplier-accumulator (MAC) is presented in this work. A full-adder circuit based on mux is used for MAC architecture. Compared to other full-adder circuits, the MUX based full adder has the highest operational speed and less transistor count. The basic building blocks for the MAC unit are identified and each of the blocks is analyzed for its performance. Power and delay is calculated for the blocks. 1-bit MAC unit is designed with enable to reduce the total power consumption based on block enable technique. Using this block, the N-bit MAC unit is constructed and the total power consumption is calculated for the MAC unit. With power reduction techniques adopted in this work, 27% of power is saved. The MAC unit designed in this work can be used in filter realizations for High speed DSP applications. Table 12 summarizes the results obtained. The Full custom design has been carried out for the proposed work and verified using cadence tools. The final GDS II is also generated and is as shown in figure 8.

Figure 8: MAC layout
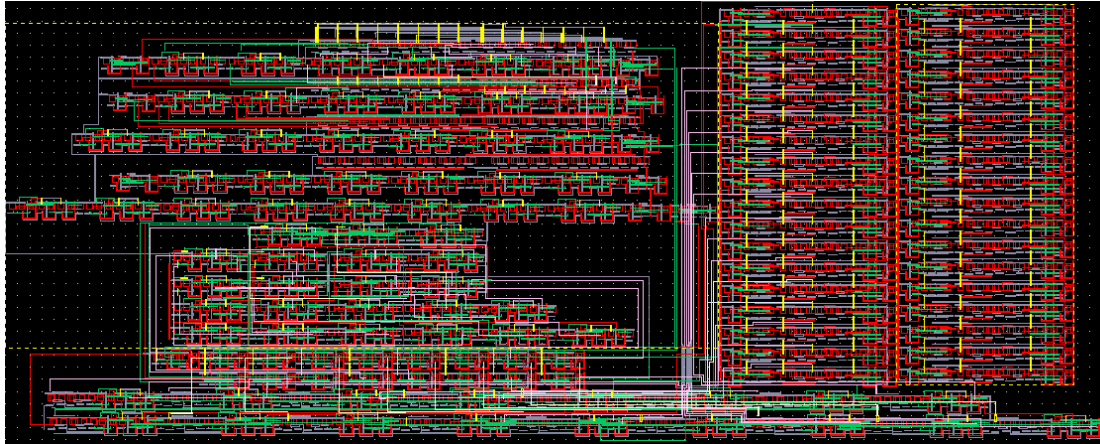
Table 12:   Characteristics of MAC

Supply voltage	1.8V
Power consumption	0.007698mW
Delay	0.437ns
Speed	2.288GHz
Power- Delay Product	3.364fj
IC process	180nm CMOS

References

[1]   S.J. Jou, C.Y.Chen, E.C. Yang, and C.C.Su(1995), "A pipelined Multiplier-accumulator using a high speed, low power static and dynamic full adder design", IEEE Custom Integrated circuit conference, 1995, pp. 593-5961

[2]   Anantha. P. Chandrakasan, Samuel Sheng, Robert W. Brodersen, "Low-Power CMOS digital design(1992)," IEEE Journal of Solid-State Circuits, Vol 27, No. 4, April, 1992

[3]   Neil H.E. Weste ,and David Harris, *CMOS VLSI Design: a circuits and systems perspective*, Addison-Wesley Publishing Company, 3rd ed.

[4]   S.J. Jou, C.Y.Chen, E.C. Yang, and C.C.Su(1997), "A pipelined multiplier-accumulator using a high speed, low power static and dynamic full adder design", IEEE journal of Solid-state circuits, vol.32, no.1, Jan.1997,pp.114-118

[5]   M.Suzuki, N.Ohkubo, T.Shinbo, T.Yamanaka, A.Shimizu, K.Sasaki, and Y. Nakagome(1993), "A 1.5ns 32-bit CMOS ALU in Double pass-transistor logic", IEEE Journal of Solid state circuits, vol.28, no. 11, November 1993, pp.1145-1151

[6]   F. Lu and H. Samulei(1993), "A 200-MHz CMOS pipelined multiplier- accumulator using a quasi-domino dynamic full adder cell design", IEEE J. Solid state circuits, vol.28, pp.123- 132, Feb 1993

[7]   P.C. Anantha, S. Samuel and R.W.Borderson (1992), "Low power CMOS digital design", IEEE J. Solid-state circuits, vol.27, pp.473-483, April 1992

[8]   Tajul Hamimi Harun(2007), "High Speed 8-bits x 8-bits Wallace Tree Multiplier", Chapter 3, *dspace.unimap.edu.my/**bits**tream/123456789/1937/5/Methodology.pdf*, May 2007