MOSVLSIDESIGN

TermProject8-bitWallaceTreeMultiplier

SubmittedBy:JhalakPatel

Contents

1. Overview

In this project, we have implemented 8bit Wallace tree multiplier at scaled supply voltage with adaptive clock stretching. Multipliers are one of the most important parts in the signal processing or other computationally intensive applications. Therefore, we aimed at designing multiplier which operates at high speed, low power and occupies optimal area. We have used 180nm technology for the project. For this project, we have created the library of blocks from scratch. We have used various novel blocks to realize the complete design such as Mirror Adder, True Single Phase Clock Flip Flop, Doubled CMOS Clocked latch. We have done the characterization of the component, the schematic and layout implemented using Cadence tools. For verifying the HSPICE Netlist, we have used NanoSim for functional simulation as well as for Power, Area and timing analysis.

Process Technology	180nm Technology
Final Area of Chip	162x185um-square
Numberof PMOS	1358
Numberof NMOS	1420
Total Numberof Transistors	2778
Power Dissipation*	2.75mW
Voltage	1.8V
Clock Frequency*	333MHz

*All the Power and Timing Calculations are for Extracted Design

2. Description

Our design takes two 8bits binary numbers as inputs for multiplication operation. The inputs to the system as fed with a clock pulse of 3ns for the optimal operation of the design. Firstly, the inputs to the design are latched using Input Flip Flop stage which are clocked by gated clock rising edge (i.e. Coming from the Latency Predictor Block). The clocking is done through the clock coming out of clock tree which ensures that the clock is not skewed or distorted. Latched outputs of the first latch are fed to AND array to generate 64bit partial products. The partial products produced are fed to the Wallace tree which basically has Layers of Adder which compresses the inputs and propagate the carry to the next stage. The output of the Wallace tree are latched using the intermediate stage Flip Flop stage. After Sampling the output at the intermediate stage, the middle bits are decoded using the *Latency Predictor Block (LPB),* which basically generates the enable signal to stall clock. The output of the LPB is gated with the input clock to generate the *enabled_clock* which is fed to all the three latches i.e. input, intermediate and output latch. The outputs sampled from the middle latch is fed to the merge adder. We have implemented merge adder using ripple carry adder which produces the final 16bit output which are latched by the output Flip Flop to generate the final output.

Inourimplementationweareimplementingeachmultiplicationwithlatencyof3or4(withthe stretchedclock)clockcycles.Butduetopipeliningofcomputationalblock(combinationlogic),

weareabletogethigherthroughputi.e.Oneoutputperclockcycle.

AdvantagesofAdaptiveClocking:

Inourdesign,wehavereducedthecriticalpathinthemergeadderstage,sothatwecan operatetheclockathigherfrequencyatlowervoltage.Wealsomadeobservationthatinthe absenceoftheadaptiveclocking,inordertooperatethecircuitatsamefrequencyweneedto applyhighersupplyvoltage.Theadaptiveclockinghelpedustoscaledowntheoperating voltage.

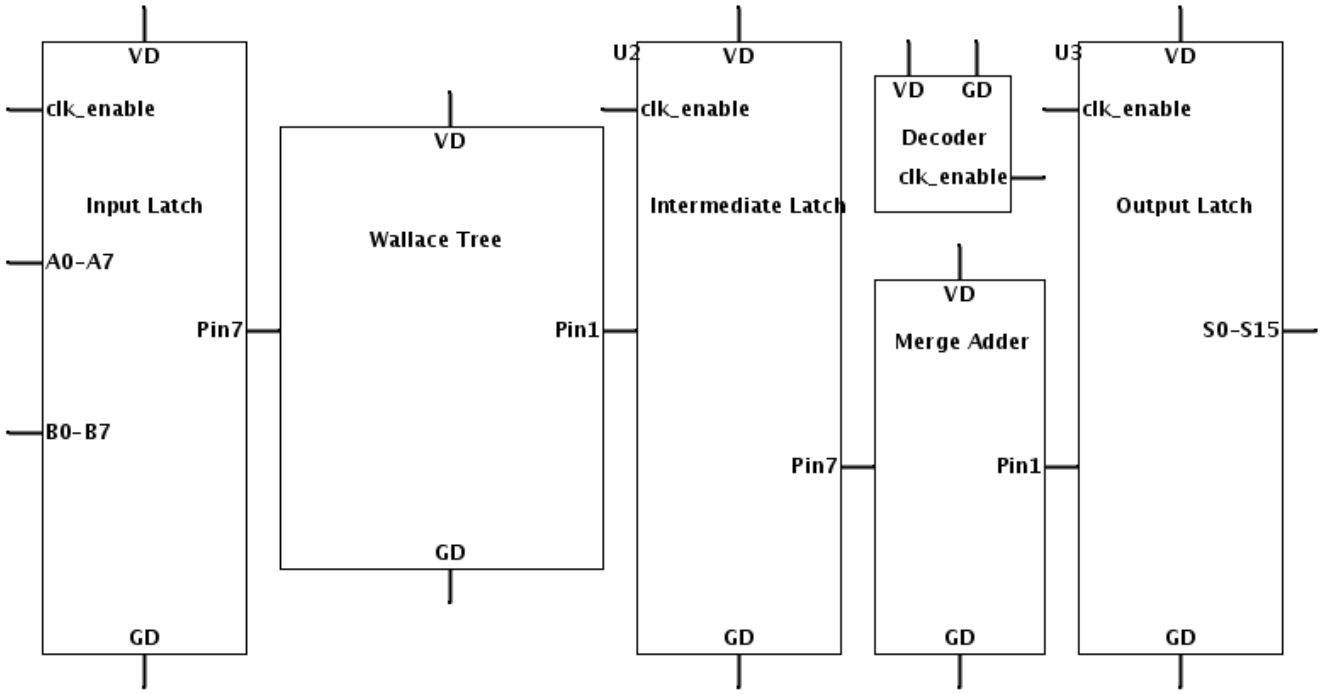

Figure1:WallaceTreeMultiplierwithAdaptiveClocking(BlockDiagram)

3.DesignComponents:

3.1MirrorAdder

WehaveImplementedFulladderasaMirrorAdder.Mirroradderiscomparativelymore fasterthanthenormalfulladdercircuit.Duetoitssymmetricnature,risetimeandfalltimefor themirrorisalmostthesameaswehaveproperlysizedourNMOSandPMOS.

MirrorAdderprovidesitsbenefitsinthecarrygenerationcircuitryastherearemaximumof twotransistorsinseriesforthecarrygenerationpath.Wehaveoptimizedthetransistorsinthe carrystageforoptimaloperationoftheadder.

Duetosymmetricnatureofthecircuit,thePMOSpullupnetworkwasdesignedandthen mirroredacrossthehorizontalaxistofacilitatethecreationofNMOSpulldownnetwork. Whichmerelyrequiredchangingthediffusiontypesandre-sizingtheactiveareastoensurethe propersizeratio.

3.2. TSPC Flip Flop

We have Implemented true single phased clock FF, which acts as a Master Slave FF. We have proposed the use of dynamic circuit based FF which comparatively has less delay than the Static CMOS based D-Flip Flop. The TSPC FF used as Edge Triggered D flip-flop operates at high speed using dynamic logic. The digital output is stored on the parasitic device capacitance while the device is not transitioning. The charge based storage limits the usage of such FF for static or low clock speed circuits. However, we were encouraged to use the dynamic circuit based FF as we planned to operate our circuit at higher clock frequency, such that the FF is clocked fast enough. As by operating a higher frequency we avoided the leakage condition for the parasitic capacitance which could have caused the FF to enter invalid states.

When the clock is low, the master stage is transparent and latches the outputs when the level goes high, the Slave stage reflects the latched output in the previous stage. For error free operation of the FF, clock pulse width should be more than delay of the slave stage.

3.3 Double C2MOS

We have implemented negative latch using Doubled C2MOS. Using Dynamic logic based latch enabled us to use it for faster switching operations. In addition to the advantage of the higher operation speed, it has only 6 transistor count, which helped in area reduction as compared to tradition D-latch formed using Static CMOS logic (i.e. NOR based or NAND based Latch)

4. Design Blocks

4.1 Wallace Tree

Wallace tree is efficient hardware implementation for N-bit multiplication. In our case we have designed Wallace tree multiplier for two 8 bit inputs. First we have generated 64 bit partial products by ANDing the combination of input bits. In the next stage the tree reduces the partial products by compression (addition) using the 4 layers of Full Adder and Half Adders. Full Adder can be thought of as a 3:2 bit compressor which reduces the partial product inputs. At the end of the Wallace Tree stage, we get the 26 bit compressed output which can be used for further computation.

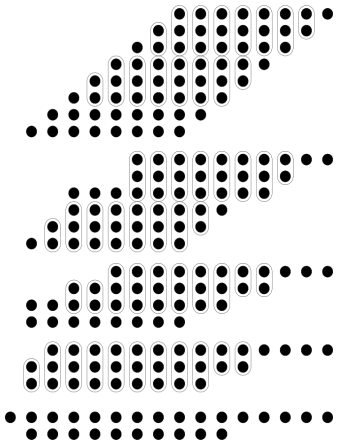

Figure 2: 8x8 Wallace Tree Multiplier (Logical Overview)

4.2LatencyPredictor

ThelatencyPredictorblockdecodestheintermediateinputstopredictifthecurrent computationrequiresclockstretchingornot.WehaveXORedtheintermediateinputsto evaluatethecarrypropagationbit,whichisANDedwiththecorrespondingoutputoftheother XORgates.WehaveusedpositiveedgetriggeredD-FlopFloptofeedbacktheenableoutput bitofthepreviouscycletogeneratethecorrectvalueofenableinthecurrentcycle.Italso ensuresthatforthenextclockcycle,enablebitissetto"1"again,sothatwecangetaccurate stallingforoneadditionalclockcycle.

TheLatencyPredictorblockalsoincludesanegativeD-latch(C2MOSlatch)whichlatchesthe enablebitoutputatthenegativeclocklevel,thusitremainslatcheduringthenextpositive clockcycle.

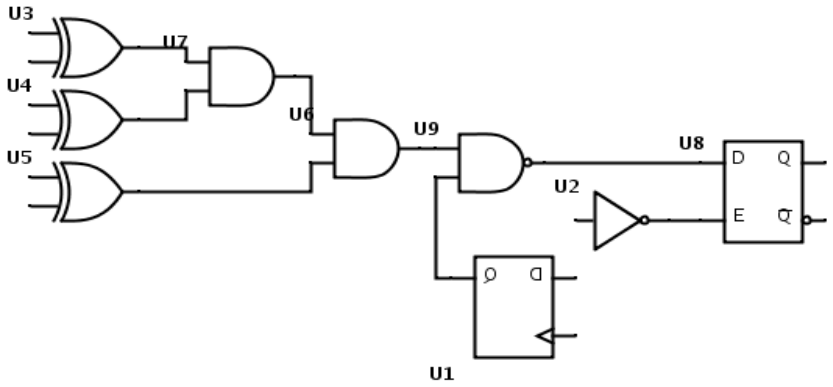

Figure3:LatencyPredictorBlock

Thecircuitallowscomputationoftheenablesignalbeforethenextsetofinputarrivei.e.Atthe nextrisingedgeoftheclock,whichisextremelycriticalforthesuccessofourdesign methodology.Thevalueoftheenablelatchedinthenegativeclockcycleisusedtodetermine whethertheoutputregisterwillbewrittenatthenextrisingedgeoftheclockorbedelayedby oneclockcycletoimplementclockstretching.Wehaveusedclockgatingtogeneratethefinal enabledclocksignaltodisablethewriteoperationtotheinput,outputandintermediatelatch

4.3MergeAdder

WehaveusedRippleCarryaddercircuitryforaddingthelatchedoutputbitofthe Wallacetree.Itactasamergeadderwhichaddsthe25-bitsnumbers.Eachfulladderinputsa *Cin* whichis *Cout* thepreviousaddersuchthateachcarrybit"ripples"tothenextfulladder. WehavefirstadderasaHalfAdderas *Cin*tothefirststageis"0".

Wehavechosenripplecarryadderasitwassimplewhichallowedusfastdesigntime.Alsothe transistorcountincaseofripplecarryadderislessascomparedtootheradderssuchas carry-selectadder.

DelayAnalysisfortheMergeAdder

MergeAdderhas25AddersStageswhichneedstopropagatethecarrytogeneratethe finalstableoutputswhichcanbelatchedatthelaterstage.Inordertoevaluatetheworstcase

time(criticaldelay)forthemergeadderstage,consideringthecarryispropagatedthroughall theadderstages:

tdelay=>2(fromtheinputtocarryinthefirstadder)+19*3(forcarrypropagationinthe lateradders)=> **57 *gatedelays***

Inordertominimizetheworstcasedelayorthecriticalpathdelay,wehaveimplemented "LatencyPredictorBlock"whichbasicallyanalyzesmiddlebits(C8-P9C9-P10C10-P11 C11-P12)forthecarrypropagation,westalltheclockforanextraclockcycleftheMerge Addercomputationifthereiscarrypropagation.Thusthetotaldelayremains57gatedelays, whichtakestwoclockcyclesdelayforcalculations.

Incaseofnocarrypropagation,weconsiderthecriticalpathup-tothemiddleaddersuchthat thetotaldelayis28gatedelay,whichenablesustooperatethecircuitathigherclockrate.

ClockPeriodCalculations

*ProbabilityofCarryPropagation*2clockcycle+ProbabilityofNoCarryPropagation*1 clockcycle* => 0.125*2+0.875*1 =>**1.125CLOCK(EFFECTIVECLOCK CYCLE)**

4.4ClockTree

Theclockenablesignal(*clk_ enable*)generatedaftertheclockgatingisusedto synchronizeallthethreelatchesusedinthedesign.The *clk_ enable* signalisfedtooptimalsize buffersstagestogeneratethefinalclockoutputsforalltheindividuallatches.Thebufferstages arelaiddowninthelayoutsuchthatthereisnodistortionclockstrengthoftheclockdriving differentlatches.Ourlayoutstrategyalsoensuresminimumskewbetweenthedifferentclock drivingthelatches.

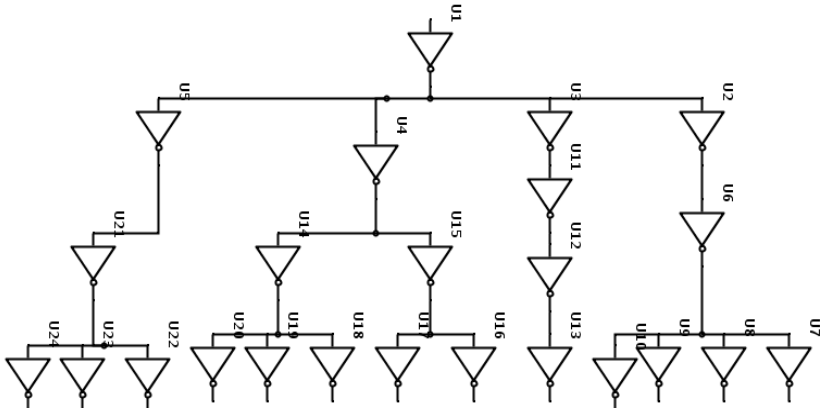

Figure4:ClockTreefortheCompleteDesign

5.AnalysisandConclusion:

5.1AreaAnalysis

DimensionofFullAdder:14.75 x *6.5um-square*

DimensionofTSPCLatch:7.5 x*7.oum-square*

DimensionofCompleteDesign:162 x*185um-square*

LayoutApproach:Inthedesign,wehavehavetriedtooptimizetheperformanceofthecircuit bylayingouttheallthethreelatchesmoreclosertoeachother.Wehaveaccomplisheditby layingouttheAdderTree(compressor)informofa"C"shapedstructure.Byusingsuchlayout technique,wesucceededtoplacetheallthethreelatchesmoreclosertoeachother.Routing andPlacementofthetheclockTreealsoplayedacriticalroleinoverallperformanceofthe design.Duetothecongestionofsignals,wehaveusedupto4metallayerswhilerouting differentcomponents.

PowerandGroundRouting:Wehaveusedmetal1layerforcompletePowerandGround routing.Wehavefollowed"Grid"topologytoroutethosesignals.

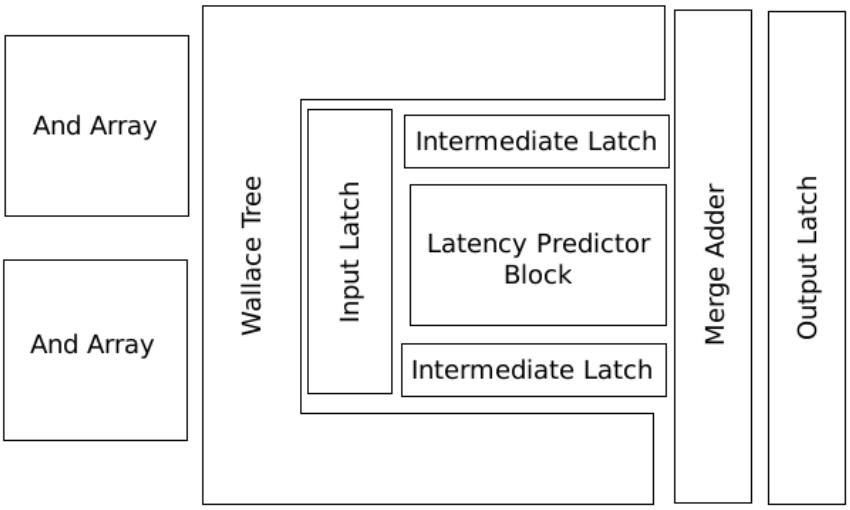

Figure5:FloorPlanning

5.2PowerAnalysis

TSPCMasterSlaveFFPowerConsumption
AverageBlockPower:7.05uW *RMSBlockPower:65.75uW*
MirrorAdderPowerConsumption
AverageBlockPower:8.83uW *RMSBlockPower:76.61uW*

CompleteDesignPowerConsumption
AverageBlockPower:2.750mW *RMSBlockPower:3.755mW*

5.3TimingAnalysis

SchematicOperatingFrequency:500MHz

ExtractedLayoutOperatingFrequency:333MHz

Extractedlayoutoperatesaslowerfrequencyascomparedtotheschematicaswehave accountedfortheparasiticcapacitancebetweenvariousmetallayersandcomponents.The extractedlayoutmodelalsoconsiderstheroutingdelayandtheresistancebetweenvarious "via-s"whiletransitioningfromonemetaltoanother.Italsoaccountsforthedelayinsupply netsi.e.PowerandGroundsignalsduetoroutingtopology.