



Dynamic Power Reduction in Modified Lifting Scheme Based DWT for Image Processing

By Prof. C. Chandrasekhar & Dr. S. Narayana Reddy

S.V. University, Tirupathi.

Abstract - Image compression is one of the major applications in image processing that imposes greater design challenges for VLSI design engineers in design and development of low power and high speed architectures. DWT is used in image compression for transformation of image from spatial to frequency domain. In this paper, DWT architecture based on lifting scheme is considered and dynamic power reduction is achieved with suitable modifications to the architecture and adoption of low power techniques. The interdependency of scaling and dilation coefficients is simplified to single hierarchy and thus reduces latency and increases throughput. Wallace tree multiplier and carry select adder are used in realizing 1D DWT architecture. The hierarchy in the design enables to adopt multi-stage and hierarchical clock gating technique thus reducing dynamic power. Power gating and DVFS techniques are also adopted to optimize power dissipation.

Keywords : Dynamic power dissipation, DWT, Lifting Scheme, Hierarchical design, low power design ASIC implementation.

GJRE-F Classification : FOR Code : 080106



DYNAMIC POWER REDUCTION IN MODIFIED LIFTING SCHEME BASED DWT FOR IMAGE PROCESSING

Strictly as per the compliance and regulations of:



RESEARCH | DIVERSITY | ETHICS

Dynamic Power Reduction in Modified Lifting Scheme Based DWT for Image Processing

Prof. C. Chandrasekhar ^α & Dr. S. Narayana Reddy ^σ

Abstract - Image compression is one of the major applications in image processing that imposes greater design challenges for VLSI design engineers in design and development of low power and high speed architectures. DWT is used in image compression for transformation of image from spatial to frequency domain. In this paper, DWT architecture based on lifting scheme is considered and dynamic power reduction is achieved with suitable modifications to the architecture and adoption of low power techniques. The interdependency of scaling and dilation coefficients is simplified to single hierarchy and thus reduces latency and increases throughput. Wallace tree multiplier and carry select adder are used in realizing 1D DWT architecture. The hierarchy in the design enables to adopt multi-stage and hierarchical clock gating technique thus reducing dynamic power. Power gating and DVFS techniques are also adopted to optimize power dissipation. The modified lifting architecture operates at a maximum frequency of 290MHz, and reduces power by more than 50%. The proposed design is implemented using 65nm TSMC low power library cells and is synthesized using Synopsys DC. The TCL scripts developed optimizes dynamic power dissipation.

Keywords : Dynamic power dissipation, DWT, Lifting Scheme, Hierarchical design, low power design ASIC implementation.

I. INTRODUCTION

DWT is recommended by JPEG2000 standards as it supports features like progressive transmission, higher compression and region of interest encoding schemes. Convolution based DWT or FIR filter bank based DWT architectures occupy large area as they require more number of multipliers and adders, thus making the computations complex and time consuming. Mobile phones and other similar hand held devices that support image/video applications demand high speed and low power architectures with reduced memory size for DWT processing. There are several architectures discussed in literature to perform lifting based DWT. General approach for 2-D DWT is to apply the 1-D DWT row-wise which produces L and H subbands and then process these sub-bands column-wise to get LL, LH, HL and HH coefficients. Several architectures like direct mapped [2], folded [3], and flipping [4] for single level and multi-level DWT have been proposed to implement 1-D lifting DWT. Many architectures that implement the Two-Dimensional

separable Forward (2D-DWT) and Inverse DWT (2D-IDWT) in order to be applied on 2D signals have been presented in the past [5], [6], [7] and [8]. These architectures are consisting of filters for performing the 1D-DWT and memory units for storing the results of the transformation. Due to the fact that streaming multimedia applications - in which the DWT is present - are characterized by high throughput requirements, this imposes the need for optimizing the design of the filters in terms of speed. Moreover, portable multimedia devices require low power consumption for increasing the battery lifetime and this can be achieved by minimizing the storage size and number of memory accesses [9]. Low power DWT architectures based on pipelining and parallel processing has been discussed in [10] and [11], in their work low power is achieved by modifying the architecture to reduce number of computations the design was implemented on FPGA. Many of the low power techniques reported in literature [12],[13], [14] and [15] for DWT propose modifications in the architecture level to reduce power dissipation. Power reduction can be accomplished at various levels of abstraction starting from architecture level to circuit level. Power reduction at the sub system level or at the circuit level can be accomplished when ASIC design of DWT architecture is performed. Many of the work reported in literature have restricted to FPGA implementation. In this paper, in order to demonstrate the dynamic power reduction techniques at various levels of abstraction, DWT architecture is considered as a test case for illustration. ASIC design of DWT architecture optimizing dynamic power reduction using 65nm TSMC libraries is performed.

Section II discusses wavelet transforms, DWT architecture and dynamic low power reduction techniques. Section III discusses proposed low power schemes for design DWT architecture sub systems. Section IV presents ASIC implementation of DWT architecture based on low power schemes. Section V discusses implementation results and performance comparison and section VI presents conclusion.

a) DWT and Low Power Schemes

In this section, DWT architecture and low power schemes are presented. Lifting scheme based DWT architecture is considered as test case for dynamic power reduction and is briefly discussed in this section. Major sources of power dissipation on VLSI circuits are also presented in this section.

Author α : HOD, Dept.of ECE SVCET, CHITTOOR,
E-mail : Umashekar_2000@yahoo.com

Author σ : prof & Head in Dept.of ECE, S.V.University, Tirupathi
E-mail : snreddysvu@yahoo.com

i. DWT architecture

In wavelet analysis, signals are represented using a set of basis functions derived by shifting and scaling a single prototype function, referred to as "mother wavelet", in time [16]. Wavelet transforms are closely related to tree structured digital filter banks and multiresolution analysis. A set of wavelet basis functions can be generated by translating and dilating the mother wavelet. A number of architectures have been proposed for calculation of DWT [2], [3], [4], [5] and [6]. The architectures are mostly folded and can be broadly

classified into serial architectures (where the inputs are supplied to the filters in a serial manner) and parallel architectures (where the inputs are supplied to the filters in a parallel manner). A methodology for implementing lifting-based DWT that reduces the memory requirements and communication between the processors, when the input is broken up into blocks is presented in [17]. Figure 1(a) and 1(b) shows the forward and inverse DWT based on lifting scheme architecture.

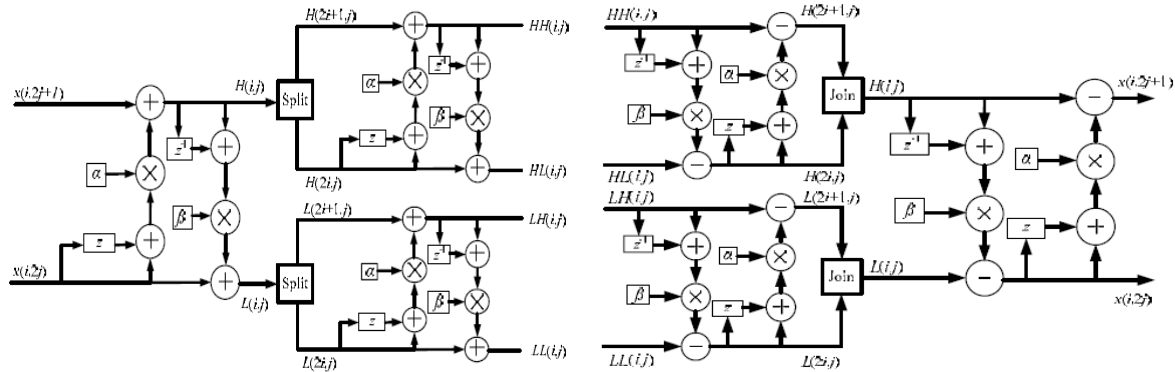


Figure 1 : Lifting based architecture (a) Forward DWT (b) Inverse DWT [17]

The z^{-1} blocks are for delay, $\alpha, \beta, \gamma, \delta, \zeta$ are the lifting coefficients and the shaded blocks are registers. 9/7 filter has been used for implementation which requires four steps for lifting and one step for scaling. The input signal x_i is split into two parts even part x_{2i} and odd part x_{2i+1} then the first step of lifting performed is given by the equations [17].

$$d_i^1 = \alpha (x_{2i} + x_{2i+2}) + x_{2i+1}$$

$$a_i^1 = \beta (d_i^1 + d_{i-1}^1) + x_{2i}$$

Then the second lifting step performed gives:

$$d_i^2 = \gamma (a_i^1 + a_{i+1}^1) + d_i^1$$

$$a_i^2 = \delta (d_i^2 + d_{i-1}^2) + a_i^1$$

Then scaling is performed and the following equations are obtained:

$$a_i = \zeta a_i^2$$

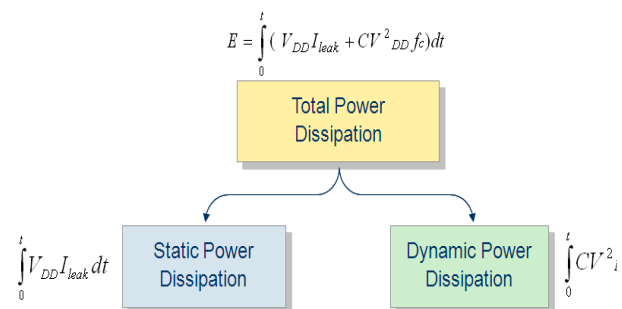
$$d_i = d_i^2 / \zeta$$

The predict step helps determine the correlation between the sets of data and predicts even data samples from odd. These samples are used in the update step for updating the present phase. Some of

the properties of the original input data can be maintained in the reduced set also by construction of a new operator using the update step. The lifting coefficients have constant values of -1.58613, -0.0529, 0.882911, 0.44350, -1.1496 for $\alpha, \beta, \gamma, \delta, \zeta$ respectively. a_i and d_i are DWT outputs after level 1 decomposition.

ii. Sources of power dissipation in CMOS VLSI circuits

Power consumption in CMOS digital circuits is divided into two major components (Static and Dynamic) as shown in Figure 2 (a).



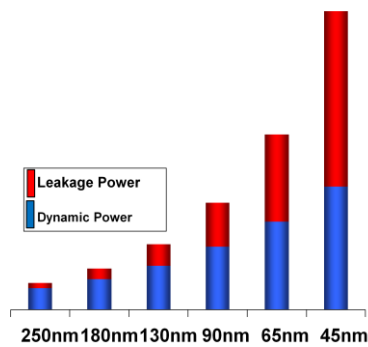


Figure 2 : nPower dissipation in CMOS circuits (a) Types of power dissipation (b) Power dissipation with technology scaling

Static power is due to leakage current and short circuit current, dynamic power is due to switching current. Power dissipation in CMOS is exponentially increased with scaling in transistor size. Figure 2(b) shows the power dissipation in CMOS with technology scaling. Dynamic power dissipation was dominating with 250nm technology, with technology scaling towards lower geometries (65nm and below), leakage power has significantly increased. However, dynamic power has also exponentially increased; this is due to the fact in increase in switching current and frequency of operation of CMOS circuits. There are various low power reduction techniques such as [18]:

- (a) Reducing voltage for lower performance blocks,
- (b) Cut off of power on blocks when they are not required
- (c) Combination of MV and/or Power Gating (shutdown), optionally retaining state on shutdown block
- (d) Lower voltage when blocks not needed, but leave them powered enough to save state w/o extra retention overhead.
- (e) Vary the voltage and/or frequency dynamically, adaptively on-the-fly, depending on immediate performance/power requirement.
- (f) Vary the well voltage to adjust threshold voltage, which in turn increases speed (forward bias) / reduces leakage (backward bias). Also known as Variable V_{th}
- (g) Reduce gate lengths in transistors along the non-critical paths
- (h) Source biasing, push the transistor to operate in cut-off region by increasing the source ground potential
- (i) Isolation/level shifting bugs, control sequencing bugs, retention scheme/control errors, Retention selection errors
- (j) Electrical problems like memory corruption hardware-software deadlock

- (k) Power gating failure/dysfunction, power-on-reset/bring-up problems, power sequencing/voltage scheduling errors

Power reduction techniques mentioned above are to be applied to the DWT architecture to optimize for low power. The major building blocks in DWT and IDWT as shown in Figure 2 are the adders, multipliers, registers and control unit for data flow control. As the focus of this work is to reduce power dissipation at various levels of abstraction, adders and multipliers are designed with low power techniques.

II. SUBSYSTEM DESIGNS FOR DWT ARCHITECTURE

An adder is the most commonly used arithmetic block in the Central Processing Unit (CPU) of a microprocessor, a Digital Signal Processor (DSP), and even in a variety of ASICs. In a DWT processor, adder is one of the important building blocks, required to compute the DWT coefficients of input signal. Multiplier used in a DWT processor also requires adder to add the partial products. Hence, design and analysis of adder is considered in this section. Speed and optimization of power of an adder is significant, to improve the overall performance of the system. But an adder also experiences the power-delay trade off. That is, its power dissipation increases with reduction in delay and vice versa. There are various architectures for adder design. 4-bit adders can be of different types. Some of those are Carry look Ahead Adder, Ripple Carry Adder, Carry Save Adder, Carry Select Adder. In many digital signal processing operations-such as correlations, convolutions, filtering, and frequency analysis-one needs to perform multiplication. Multiplication algorithms will be used to illustrate methods of designing different cells so that they fit into a larger structure. In order to introduce these designs, simple and serial and parallel multipliers will be introduced. High-speed parallel multipliers are becoming one of the keys in RISCs (Reduced Instruction Set Computers), DSPs (Digital Signal Processors), and graphics accelerators and so on. Parallel multipliers are used in data processor as well as in digital signal processors. There are various multiplier architectures reported in literature, Wallace tree, booths multiplier, BZ-FAD multiplier, Shift and Add multiplier and Array multiplier are most popular for DSP applications. In this work, the adders and multipliers are modeled using HDL and is synthesized using TSMC 65nm CMOS libraries using Synopsys DC. The synthesis results generate reports that provide information on area, delay and power dissipation. The results obtained are presented in table 1 and table 2 is without low power techniques. Multipliers are designed using carry save adders.

Table 1 : Full Adder Design Comparisons

Type of adder (16-bit)	No. of transistors	Power - μ W	Delay -ps
Ripple carry adders	286	40.5505	600
Carry save adder	92	18.9241	74
Carry select adder	102	16.897	65
Carry look ahead adder	621	55.1482	62

Table 2 : Power comparison of multipliers

Multipliers	Total Dynamic power (uW)	Cell Leakage power(uW)
BZ-FAD Multiplier	161.27	5.32
Shift and Add Multiplier	241.14	4.71
Booth Multiplier	468.02	12.69
Array Multiplier	298.83	10.24
Wallace Tree Multiplier	341.62	13.81

In order to reduce power dissipation of adder and multiplier, multi VDD technique is adopted. Reducing VDD supply voltage, reduce the power consumption, there will be no effect on area. From the results obtained it is found that power consumption is a quadratic function of voltage ($\text{Power} = f(V_{DD}^2)$). Decrease in supply voltage increases the overall delay ($\text{Delay} = (KV_{DD}/V_{DD} - V_t)^{\alpha}$).

The synthesis results generate reports that provide information on area, delay and power dissipation. The results obtained are presented in table 1 and table 2 is without low power techniques. Multipliers are designed using carry save adders.

Table 3 : Reduction in dynamic power with voltage scaling

Voltage	Dynamic Power (uW)	Delay (ns)
1.8	341.62	1.76
1.5	156.12	1.934
1.2	82.61	2.34
1	41.29	3.99

Forward DWT and inverse DWT architecture are realized using carry save adder and Wallace tree multiplier due to its delay. Voltage scaling significantly

reduce power dissipation, however, delay increases. Thus in this work, a trade-off between power and delay is considered and thus the power supply voltage is chosen to 1.2V, thus the multiplier constitutes a delay of 2.34ns and carry save adder has a delay of 112 ps. The multipliers and adders with scaled operating voltages are adopted in design of a modified DWT architecture, to demonstrate the importance of architectural modifications leading to low power. Next section discusses low power techniques addressed at architectural level.

a) Modified lifting based DWT architecture for low power

Lifting equations presented in (1) – (6) when realized using HDL model is a sequential process, as the scaling factors and are dependent on previous samples, thus introducing latency. In order to increase throughput and latency modified equation are derived. The modified lifting equations eliminate dependency of outputs on previous samples. We have obtained the equations for a_i and d_i by substituting (4) in (3), (3) in (2) and so on. The lifting coefficients were substituted and the results were scaled by multiplying with 256 to avoid decimal and to round off the values. The modified lifting scheme equations are:

- $$a_i = 294 * (8(6 * x_{2i} + 4 * x_{2i-2} + x_{2i} + 4 * x_{2i+4} + x_{2i-4} + 4 * x_{2i+2}) - 5 * (3 * x_{2i+1} + x_{2i+3} + x_{2i+3} + 3 * x_{2i-1}) + 100 * (2 * x_{2i} + x_{2i+2} + x_{2i+2}) - 180 * (2 * x_{2i} + x_{2i+2} + x_{2i+2}) + 113 * (x_{2i+1} + x_{2i+1}) + 21 * (2 * x_{2i} + x_{2i+2} + x_{2i+2}) - 13 * (x_{2i+1} + x_{2i+1} + x_{2i-1}))$$
- $$d_i = 19 * (3 * x_{2i+3} * x_{2i+2} + x_{2i+4} + x_{2i-2}) + (-12) * (2 * x_{2i+1} + x_{2i-1} + x_{2i+3}) + 226 * (x_{2i} + x_{2i+2}) - 406 * (x_{2i} + x_{2i+2}) + x_{2i+1}$$

These equations are obtained by taking coefficients as common. The equations have initial latency, as the input samples need to be stored before DWT a_i and d_i coefficients computations.

The design of low power architecture to reduce dynamic power dissipation is based on equations (7) and (8). From the equation the following are the observations made:

- a_i and d_i coefficients are computed based on input samples and lifting coefficients. Every output sample depends upon x_0 to x_4 input samples. Input samples are multiplied by coefficients as per the equations.
- Common factors are identified between a_i and d_i equations and these common functions are realized once and are reused to reduce the circuit complexity.
- Lifting coefficients are stored in memory and are retrieved only once and used for computation of a_i and d_i components.
- Pipelined architecture is developed to realize a_i and d_i equations.

The proposed architecture shown in Figure 4 takes two inputs and gives two outputs per cycle. Data1 and Data2 are the odd and even input samples given to hardware in single clock for 100 % hardware utilization. This architecture is very simple design as compared to

other architectures suggested in [20] which have complex control path to achieve 100% hardware utilization. The row processor and column processor shown in figure 4 are realized using modified lifting scheme based equations.

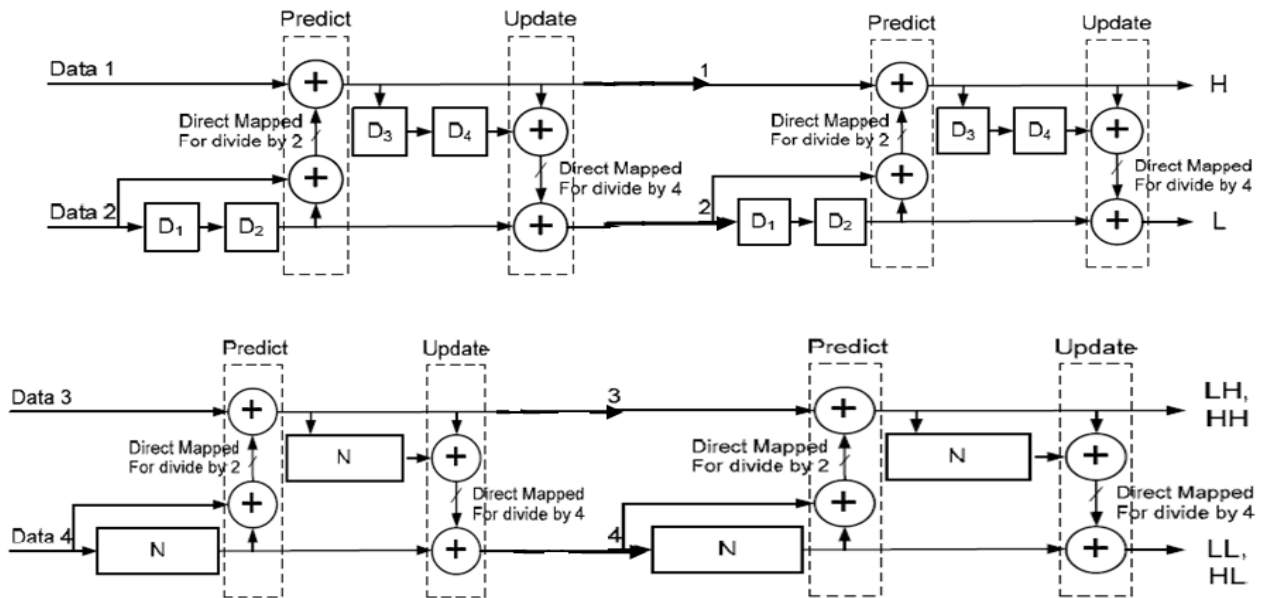


Figure 3 : Row processor and column processor for modified lifting DWT

Based on the architecture shown in figure 3 and equations presented in (7) and (8), the top level model for the architecture is shown in figure 4. A detailed data flow for the proposed architecture is presented in the figure 4.

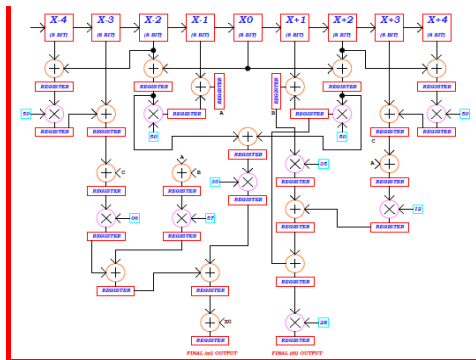


Figure 4 : Modified lifting scheme architecture to reduce dynamic power

The modified architecture derived consists of the following blocks: parallel input and serial output register, serial input and parallel output register, Multiplier and adders and control unit. The HDL model is developed and the design is verified for its functionality using test bench in ModelSim. The functionally correct HDL code is synthesized using Synopsys DC targeting TSMC 65 nm library and technology files. The reports obtained are compiled and presented in table 4.

Table 4 : ASIC synthesis results of modified lifting based DWT

Parameters	Reference [19]	Modified lifting based DWT
Area (sq.mm)	21984.657462	29542.89061
Power (μ W)	962.9536	604.712
Operating frequency (max) MHz	212	278

From the results obtained and tabulated in Table 4, it is found that due to changes in architecture that reduces number of stages in DWT computation, the dynamic power dissipation is reduced by 37%. However, the area is increased due to increase in registers and intermediate storage units, the design is synthesized to obtain minimum delay and zero slack requirement. Due to architectural changes it is demonstrated that dynamic power is reduced by 37%. In order to further reduce power dissipation various other dynamic low power techniques are introduced for optimization.

b) Dynamic low power reduction techniques

There is various dynamic low power techniques that have been recommend by synthesis tools like Design Compiler. In this work, Synopsys DC supporting low power design library is chosen for low power implementation. The low power techniques adopted for ASIC implementation of modified lifting based DWT architecture are: clock gating techniques, power gating technique, device sizing, logic restructuring, balanced delay paths to reduce glitch and Dynamic Voltage and Frequency Scaling (DVS, DFS).

i. Dynamic power reduction techniques for modified lifting based DWT

The simplest, general (or automatic) clock gating inserts a single clock gate for each register bank. Most tools permit the user "split" register banks or to prevent clock gate "sharing" across unrelated register banks. To save even more dynamic power, advanced clock gating styles such as multi-stage and hierarchical can be used, depending on design architecture and design requirements. The modified lifting DWT have common coefficients and thus need to be enabling at different instants of time and hence multi-stage clock gating technique is implemented. The 2D DWT architecture is realized using sub systems (multipliers, adders and registers), 1D DWT and finally 2D DWT, in order to reduce power dissipation hierarchical clock

gating technique is adopted. Figure 5 shows the multi-stage clock gating technique introduced into the row processor. Enable adder enables all adders together, similarly the enable reg enables all intermediate registers, thus saving power.

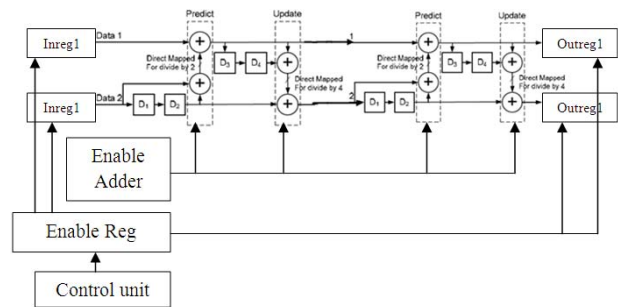


Figure 5 : Multi-stage clock gating technique on modified DWT

Figure 6 shows the block diagram of 2D DWT based on modified lifting scheme. 1D DWT is used in the first stage as well as the second stage. The first stage performs DWT on row and second level performs DWT on column data. Every 1D DWT have internal control logic that executes multi-gate clock gating technique. In the top module, hierarchical clock gating technique is adopted to reduce dynamic power dissipation.

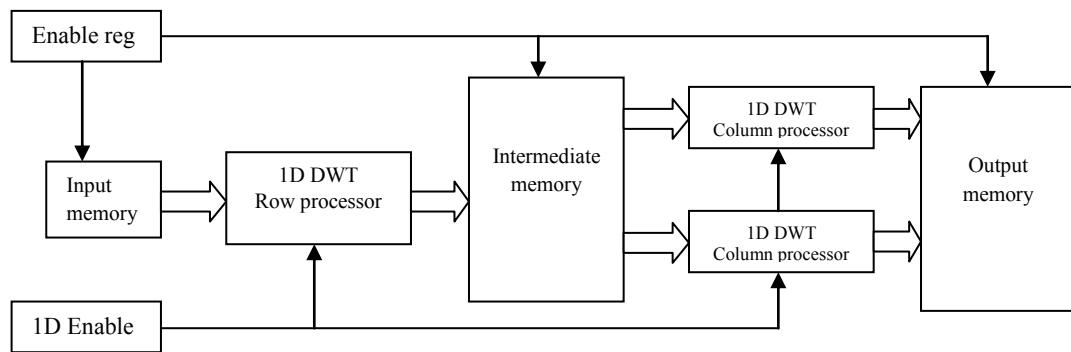


Figure 6 : Hierarchical clock gating for 2D DWT

In order to implement power gating technique power gates and state retention register required. Power gating cells are required for turning blocks on and off. State retention registers in their turn are useful because, if the state of a shut down or "sleeping" block needs to be retained the most automated method to retain the state is the use of retention registers. These registers have a backup power supply connection that remains always on to hold the state of the register via a high voltage threshold latch built into the register. An isolation cell is required to ensure electrical and logical isolation

of logic that is shut down from active logic in a design. The reason this is required is because when a block is shut down the internal signal level will transition to an unknown, floating state. Also always on cells are required between switched and steady state blocks to ensuring interoperability. Figure 7 shows the power gating logic for dynamic power reduction. Multiple voltages are used to drive the cells that are active or in standby. In the hierarchical design shown in Figure 6, 1D DWT are active during computation and inactive during data storage, thus power gating techniques are

inserted. The most common approach to provide state retention during power gating is to replace a standard register with a retention register.

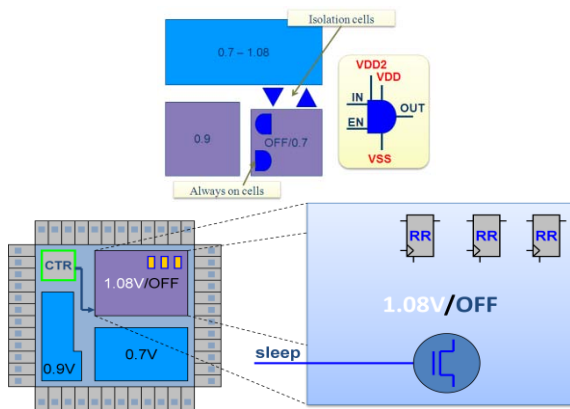


Figure 7 : Power gating techniques for modified DWT

As the modified lifting is hierarchical in nature and consists of multiple parallel data paths, power gating is easily implemented. Glitching is due to a mismatch in the path lengths in the logic network. If all input signals of a gate change simultaneously, no glitching occurs. Critical path is estimated based on synthesis report, the critical paths identified are manually observed, if they introduce any glitches. Based on the observations made, multiple critical paths that are in parallel are identified having mismatch in path lengths, thus intermediate registers are introduced at eh inputs and outputs of DWT architecture to introduce equal delay, thus reducing glitches. The fan-out out constrain is set to 4 to obtain reduced number of critical paths.

To achieve further improvements in power reduction without resorting to custom circuit techniques, Dynamic Voltage and Frequency Scaling can be used. Dynamic Voltage and Frequency Scaling is effective because of the following two facts:

- The amount of energy required to complete a task is proportional to the square of the supply voltage.
- The maximum frequency of any CMOS circuit is proportional to the supply voltage.

So if the supply voltage is decreased there is a square-law reduction in energy to complete a given task. However the task takes longer to complete because of the linear reduction in frequency. Therefore, the principle gain with Dynamic Voltage and Frequency Scaling is with respect to dynamic power consumption.

Dynamic voltage and frequency scaling adjusts performance and energy consumption levels while the logic circuit is active. It is required to reduce processor frequency and voltage to obtain quadratic energy savings. DVFS is an effective way of reducing the CPU energy consumption by providing computation power.

DVFS technique has been proven to be a highly effective technique for power minimization subject to a performance constraint. DVFS should consider not only the CPU power, but also the total system power dissipation. In this work, to realize 2D DWT, multiple 1D DWT architecture is realized using modified lifting scheme logic. Thus DVFS is adopted to minimize power dissipation.

DVFS computation for modified lifting DWT: Workload of a task, W_{task} , is defined as the total number of clock cycles required to compute 1D DWT.

$$W_{task} = \sum_{i=1}^n CPI_i$$

n: total number of iterations in DWT, CPI: clock cycles per DWT coefficient computation. The maximum value of n is 7 as there are 7 different partial factors to be added in computing a_i . Each computation of partial product requires 4 clocks, as there are multipliers and adders. The task execution time, T_{task} , is a function of DWT processor frequency, f_{DWTpf}

$$T_{task} = \frac{W_{task}}{f_{DWTpf}}$$

To save DWTpf energy using DVFS for a given deadline D, choosing a f_{DWTpf} , at which T_{task} can be closest to D.

$$T_{task} = D, \quad f_{DWTpf} = \frac{W_{task}}{D}$$

From the first cut synthesis results obtained, f_{DWTpf} is 290MHz. All the above discussed dynamic low power techniques have been included in the constrains file to minimize power dissipation.

III. ASIC IMPLEMENTATION AND RESULT ANALYSIS

The simulation results for modified DWT are presented in this section. There are sixty four inputs, each having bit width of twenty bits. These inputs are serially sent to the DWT architecture. The DWT consists of registers, multiplexer, adder and multiplier. Whenever the inputs are sent through SIPO (serial input parallel output), the data has been divided into even data and odd data. The even data and odd data are stored in the temporary registers. When the reset is high, the temporary register value consists of zero, whenever the reset is low, the input data is split into the even data and odd data. The input data is read up to sixty four clock cycles, after that the data read according to the lifting scheme. The output data consists of low pass and high pass elements. This is the 1-D discrete wavelet

transform. The two level discrete wavelet transform is that the low pass and the high pass filter outputs are again divided into LL, LH and HH, HL. The output is verified in the VCS. Figure 8 shows the VCS simulation results of DWT.

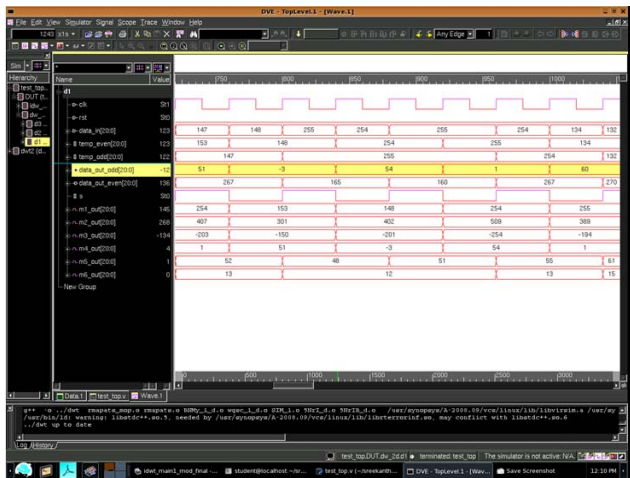


Figure 8 : 1-D Discrete Wavelet Transform output waveform in VCS

From the simulation results obtained the logic correctness is verified and the HDL model is synthesized for low power optimization. The low power design flow adopted in this work is shown in figure 9. Low Power design techniques have their impact on libraries, because in order to implement these techniques special cells (high-V_{th} MTCMOS power switches, isolation cells, level shifters, retention registers and Always-On buffers) are required in addition to the basic cells already included in digital standard cell libraries.

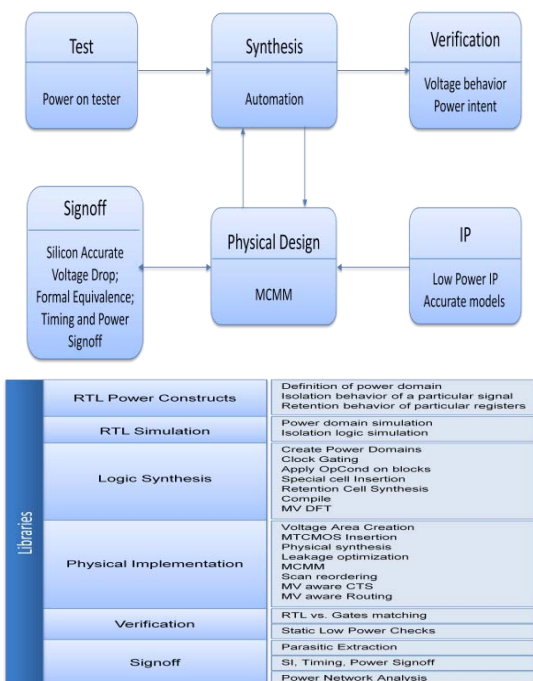


Figure 9 : Low power design flow

IV. IMPLEMENTATION RESULTS AND DISCUSSION

In this work, ASIC design flow is restricted to synthesis only for the modified lifting DWT, thus low power libraries and low power IPS from Synopsys design ware are adopted for synthesis. The synthesis constraint file is set for low power synthesis, the Synopsys DC constraints are:

```
create_power_domain TOP
create_power_domain GPRS -elements GPRS
create_supply_port VDD
create_supply_net VDD -domain TOP
create_supply_net VDD -domain GPRS -reuse
connect_supply_net VDD -ports VDD
create_supply_port VSS
create_supply_net VSS -domain TOP
create_supply_net VSS -domain GPRS -reuse
connect_supply_net VSS -ports VSS
create_supply_net VDDGS -domain GPRS
set_domain_supply_net TOP \
    -primary_power_net VDD \
    -primary_ground_net VSS
set_domain_supply_net GPRS \
    -primary_power_net VDDGS \
    -primary_ground_net VSS
create_power_switch gprs_sw \
    -domain GPRS \
    -input_supply_port {in VDD} \
    -output_supply_port {out VDDGS} \
    -control_port {gprs_sd}
    -applies_to outputs
```

```
set_isolation_control gprs_iso_out \
    -domain GPRS \
    -isolation_signal PwrCtrl/gprs_iso \
    -isolation_sense low \
    -location parent
PwrCtrl/gprs_sd \
    -on_state {state2002 in {gprs_sd}}
set_isolation gprs_iso_out \
    -domain GPRS \
    -isolation_power_net VDD \
    -isolation_ground_net VSS \
    -clamp_value 1 \
set_retention gprs_ret -domain GPRS \
    -retention_power_net VDDGS \
    -retention_ground_net VSS
set_retention_control gprs_ret
    -domain GPRS \
    -save_signal {PwrCtrl/gprs_restore low} \
    -restore_signal {PwrCtrl/gprs_restore high}
map_retention_cell gprs_ret -domain GPRS -lib_cells
RDFFNX1
add_port_state VDD
    -state {HV 1.2}
add_pst_state sleep -pst chiptop_pst -state {HV OFF}
```

The constraints are set according to the command set in the file above. The low power constraints are supported only if the RTL is hierarchical and is parallel in nature. The constraints file is shown in below. The constraints for dynamic power reduction discussed earlier are set in a constraints file and are used for synthesis. The TCL scripts for DWT_TOP_MODULE are presented below and are used for synthesis.

#Reading design

```
analyze -library WORK -format verilog
{./RTL/top_DWTpf.v ./RTL/1ddwt.v
./RTL/power_controller.v}
read_file -format verilog {./RTL/top_DWTpf.v}
name_format \
```

```
-isolation_prefix "ISO_"\
```

```
-level_shift_prefix "LS_"
```

#Reading UPF

```
source ./inputs/chiptop+.upf
```

#Setting voltages and options

```
set_voltage 1.2 -obj {VDD VDDGS}
```

```
set_voltage 0.000 -obj {VSS}
```

```
set auto_insert_level_shifters_on_clocks all
```

```
set_dont_touch [get_nets Ovfl]
```

```
set_dont_use saed65nm_typ_ht/AODFF*
```

```
set_dont_use saed65nm_max/AODFF*
```

```
set_dont_use saed65nm_min/AODFF*
```

#Reading constraints

```
source ./inputs/chiptop+_s0.sdc
```

#Compiling

```
compile
```

#Writing out results

```
change_names -rule verilog -hier
```

```
write -f verilog
```

```
-h -out ./results/compile.v
```

```
write -f ddc
```

```
-h -out ./results/compile.ddc
```

```
save_upf ./results/compile.upf
```

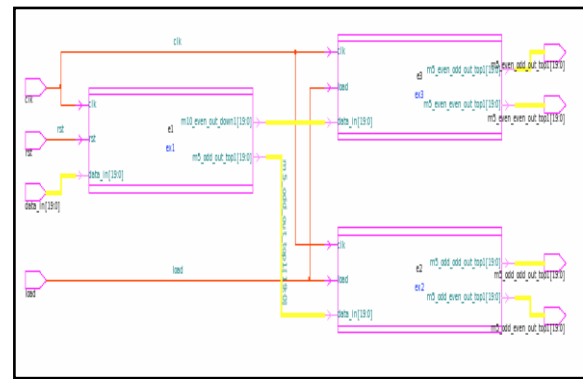


Figure 10 : Modified DWT synthesized netlist

Figure 10 shows the synthesis netlist obtained using 65nm technology and the interconnections used in the design along with clock tree network. Figure 11 shows the synthesized netlist along with clock tree network.

RTL model developed for the modified lifting scheme based DWT architecture is remodeled for ASIC implementation. The design is synthesized using Design Compiler and timing analysis is carried out using Prime Time. The design requires 42 input-output ports and requires 550 cells. The total combinational area is 21527.410 sq umm and non-combinational area is 10256.23 sq umm. Total dynamic power is 498.3µW. Due to the low power techniques adopted the dynamic power dissipation is reduced by 19%. From the results obtained, design of architecture achieves 37% power reduction; low power techniques presented in this section reduces power dissipation by 17%. Thus maximum power dissipation is achieved at the architecture abstract level.

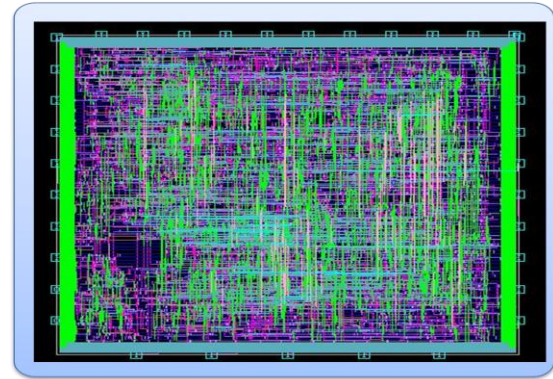
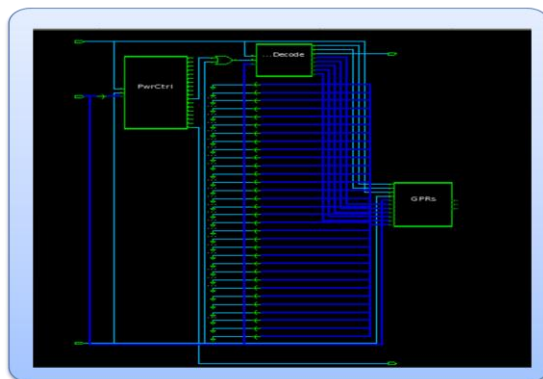


Figure 11 : Synthesized netlist of modified DWT and clock tree network

Table 5 : Dynamic power reduction comparison

Parameters	Modified lifting based DWT	Modified lifting based DWT with low power
Area (sq.mm)	29542.89061	31783.64
Power (μ W)	604.712	489.36
Operating frequency (max) MHz	278	290

DWT architecture is considered as a case study to analyze the power dissipation factor in VLSI circuits. Power saving achieved at various levels of hierarchy is proven in this work. Starting from architecture level to circuit level, power reduction need to be performed and is illustrated in this work.

V. CONCLUSION

In this work, a modified lifting based DWT is proposed, designed and implemented using 64nm TSMC low power design library. Lifting based DWT is considered to illustrate the techniques that can be adopted to reduce dynamic power. Modification in the architecture level as well as at different abstraction levels are considered for power reduction. Low power library cells from Synopsys design ware are considered for synthesis. TCL scripts for constraining the design for various dynamic power dissipation are developed. The RTL model developed is synthesized and performances are estimated. From the results obtained it is found that there is a total of 50% power reduction as compared with direct implementation. The developed low power techniques can be adopted to other complex designs. Further power dissipation can be reduced at the physical design stage.

VI. ACKNOWLEDGEMENT

The authors would like to acknowledge Dr. Cyril Prasanna Raj P, for his valuable support and guidance extended in completion of this work.

REFERENCES RÉFÉRENCES REFERENCIAS

1. I. Daubechies and W. Sweldens, "Factoring Wavelet transforms into Lifting Schemes," The J. of Fourier Analysis and Applications, vol. 4, 1, pp. 247–269, 1998
2. C.C. Liu, Y.H. Shiao, and J.M. Jou, "Design and Implementation of a Progressive Image Coding Chip Based on the Lifted Wavelet Transform," in Proc. of the 11th VLSI Design/CAD Symposium, Taiwan, 2000.
3. C.J Lian, K.F. Chen, H.H. Chen, and L.G. Chen, "Lifting Based Discrete Wavelet Transform Architecture for JPEG 2000," in IEEE International Symposium on Circuits and Systems, Sydney, Australia, pp. 445–448, 2001
4. C.T. Huang, P.C. Tseng, and L.G. Chen, "Flipping Structure: An Efficient VLSI Architecture for Lifting-Based Discrete Wavelet Transform," in IEEE Transactions on Signal Processing, pp. 1080–1089, 2004
5. C. Chakrabarti and M. Vishwanath, "Efficient realizations of the discrete and continuous wavelet transforms: from single chip implementations to SIMD parallel computers," IEEE Trans. Signal Processing, vol. 43, no.3, pp. 759–771, March 1995.
6. C. Chakrabarti and M. Vishwanath and R. M. Owens, "Architectures for wavelet transforms: A survey," Journal of VLSI Signal Processing, vol. 4, no. 2, pp 171–192, 1996.
7. Vishwanath, R. M. Owens, M. J. Irwin "VLSI architectures for the discrete wavelet transform", IEEE Trans. Circuits and Syst. II, vol. 42, no. 5, May 1995.
8. N. D. Zervas, G. P. Anagnostopoulos, V. Spiliotopoulos, Y. Andreopoulos and C.E. Goutis, "Evaluation of design alternatives for the 2-D-discrete wavelet transform", IEEE Trans. Circuits and Syst. Video Technol., vol. 11, no. 2, pp. 1246–1262, December 2001.
9. F. Catthoor, S. Wuytack, E. De Greff, F. Balasa, L. Nachtergale, A. Vandecappelle, "Custom Memory Management Methodology -Exploration of Memory management Organization for Embedded Multimedia System Design", Kluwer Academic Publishers, 1998.
10. Bing-Fei Wu and Chung-Fu Lin, "A High-Performance and Memory-Efficient Pipeline Architecture for the 5/3 and 9/7 Discrete Wavelet Transform of JPEG2000 Codec," IEEE Trans. on circuit and systems for video Technology, vol. 15, no. 12, pp. 1615–1627, December 2005
11. Nagabushnam, Cyril Prasanna Raj P, Ramachandra, Design and FPGA Implementation of Modified Distributive Arithmetic Based DWT-IDWT Processor for Image Compression, European Journal of Scientific Research, Vol. 2, pp. 23–26, 2009
12. Cyril Prasanna Raj P, Low power DWT for image compression, SASTech Journal, Vol.7, pp. 56–61, 2008
13. F. Marino, "Efficient high-speed/low-power pipelined architecture for the direct 2-D discrete wavelet transform," IEEE Trans. Circuits Systems II, Analog Digit. Process., vol.47, no.12, pp.1476–1491, 2000
14. T. Park and S. Jung, "High speed lattice based VLSI architecture of 2D discrete wavelet transform for real-time video signal processing," IEEE Trans. Consumer Elect., vol.48, no.4, pp.1026–1032, 2002

15. Yeong-Kang Lai, Lien-Fei Chen and Yui-Chih shih, "A Highperformance and Memory-Efficient VLSI Architecture with Parallel Scanning method for 2-D Lifting-Based Discrete Wavelet Transform" IEEE Transaction on Consumer Electronics, vol. 55, No. 2, May 2009
16. P. P. Vaidyanathan, Multirate systems and Filter Banks, Englewood Cliffs, Prentice-Hall, 1993
17. Tinku Acharya and Chaitali Chakrabarti, "A Survey on Lifting-based Discrete Wavelet Transform Architectures", Journal of VLSI Signal Processing 42, 321–339, 2006
18. Neil H.E Weste and David Harris, CMOS VLSI Design – A Circuit and System Perspective, 3rd edition, Pearson Education, 2005.
19. Shanthala. S, Cyril Prasanna Raj P and Dr. S. Y. Kulkarni "Design and VLSI implementation of Pipelined Multiply Accumulate Unit" was presented at International Conference on Emerging Trends in Engineering and Technology (ICETET 09) during 16th – 18th December 2009 at G.H. Rasoni College of Engineering , Nagpur (Maharashtra).
20. A.D.Darji, A.N.Chandorkar, and S.N.Merchant, Memory Efficient and Low power VLSI architecture for 2-D Lifting based DWT with Dual data Scan Technique, Recent Researches in Circuits, Systems and Signal Processing, ISBN: 978-1-61804-017-6.



This page is intentionally left blank