WELCOME TO MY DEFENSE

Saraju P. Mohanty

Energy and Transient Power Minimization During Behavioral Synthesis

Saraju P. Mohanty

Dept. of Computer Science and Engineering

University of South Florida

smohanty@csee.usf.edu

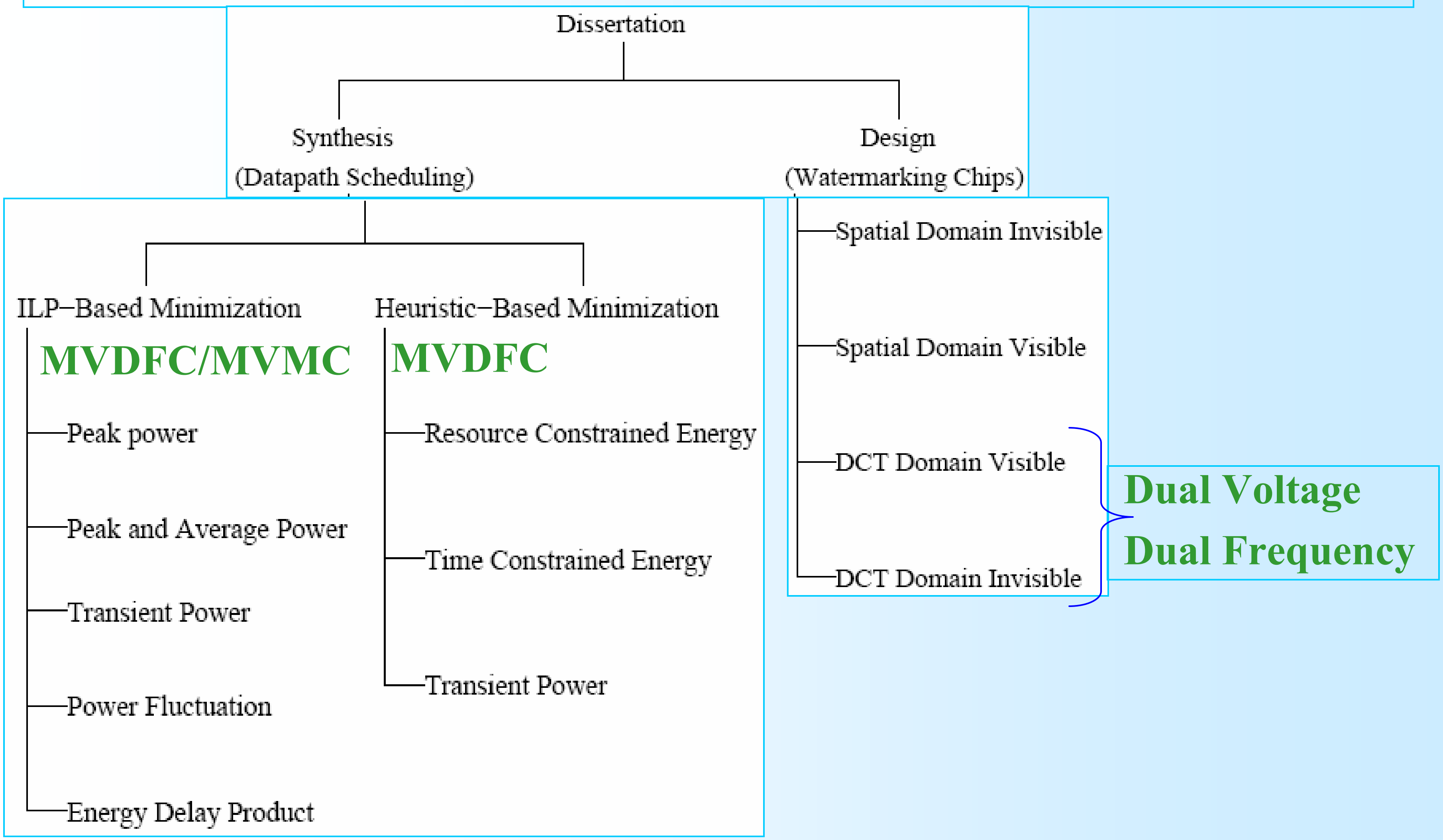
For more details, visit :

http://www.csee.usf.edu/~smohanty/research

Outline of the Talk

- Introduction
- Related Works
- Target Architecture
- Proposed Datapath Scheduling Schemes
- Image Watermarking Chip Design
- Conclusions

Dissertation Overview



- Simultaneous minimization of various powers and energy considered.
- Secure JPEG Encoder and Secure Digital Still Camera

What is High-Level Synthesis ??

McFarland (1990)

"HLS is conversion or translation from an algorithmic level specification of the behavior of a digital system to a RT level structure that implements that behavior."

[Analogous to "compiler" that translates high-level language like C/Pascal to assembly language.]

NOTE: also known as Behavioral Synthesis.

Various Phases of Behavioral Synthesis

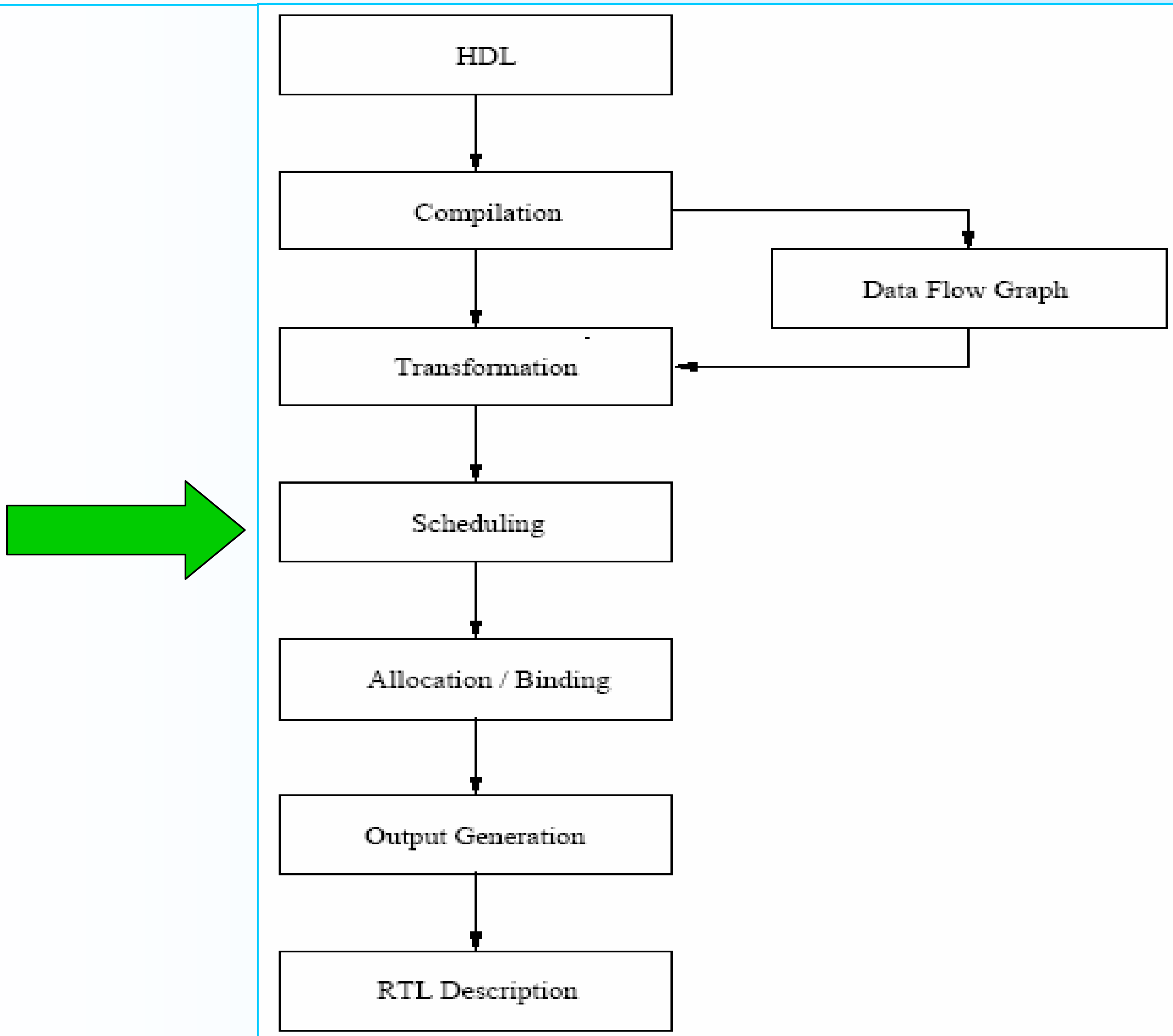


Figure 1.4. Various Phases of High-Level Synthesis

Why Power Reduction ?

- To reduce energy costs
- To increase battery life time
- To increase battery efficiency
- To maintain supply voltage levels
- To reduce power supply noise
- To reduce cross-talk and electromagnetic noise
- To use smaller heat sinks
- To make packaging cheaper
- To increase reliability
- To reduce use of natural resources

Why Dynamic Power Minimization ??

- Veendrick Observation: In a well designed circuit, short-circuit power dissipation is less than 20% of the dynamic power dissipation.
- Sylvester and Kaul: At larger switching activity the static power is negligible compared to the dynamic power.

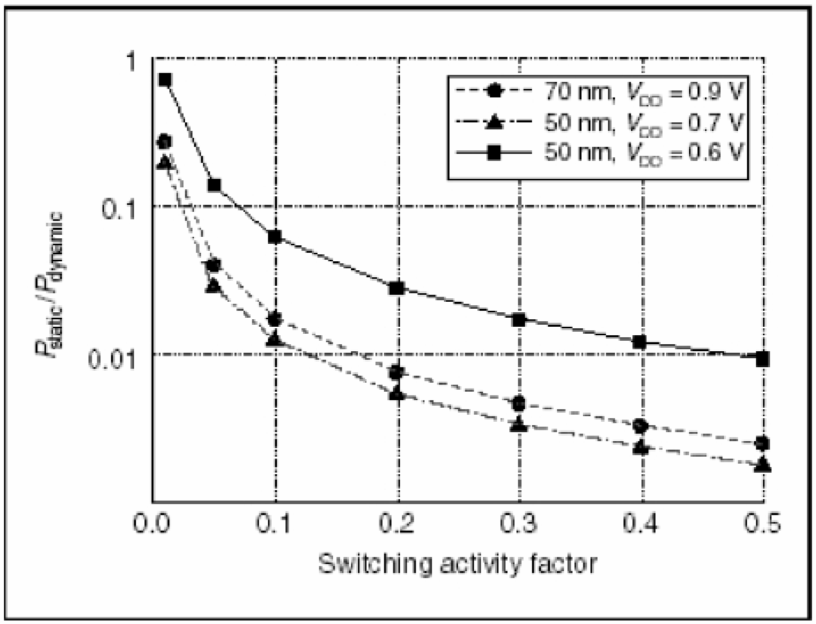


Figure 1.10. Static Vs Dynamic Power Dissipation for different Switching Activity [3, 4]

Dynamic Power: Major one

$$P_{dynamic} = \frac{1}{2} C_L V_{dd}^2 N f$$

C_L = load capacitor, V_{dd} = supply voltage,

N = average number of transitions/clock cycle

= E(sw) = 2 $a_{0\text{->}1}$ = switching activity

f = clock frequency

Note:

1. N*f is transition density
2. C_L* N (= C_{sw} = C_{eff}) is the effective switching capacitance

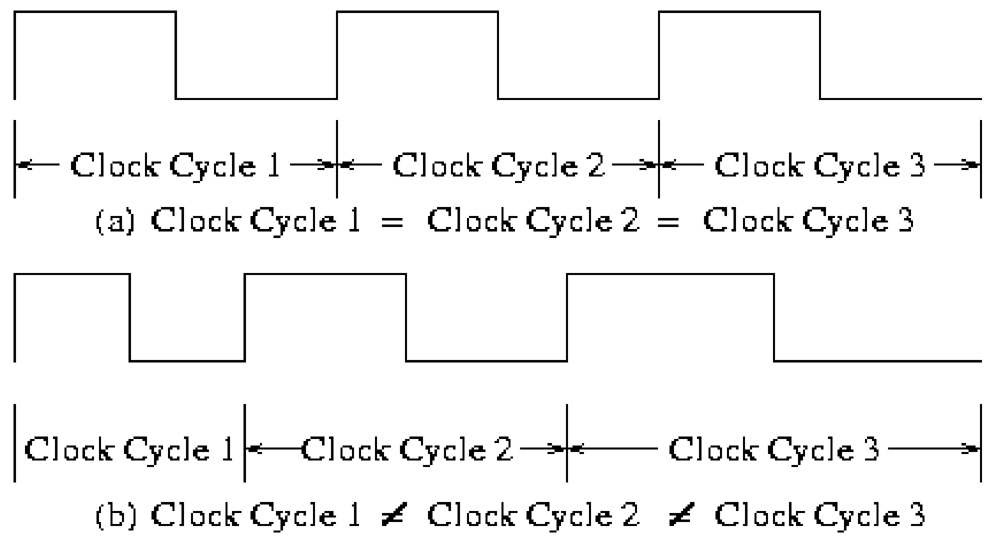
Dynamic Power Reduction: How ??

- Reduce Supply Voltage (V_{dd}): delay increases; performance degradation
- Reduce Clock Frequency (f): only power saving no energy, performance degradation
- Reduce Switching Activity (N or E(sw)): no switching no power loss !!! Not in fully under designers control. Switching activity depends on the logic function. Temporal/and spatial correlations difficult to handle.
- Reduce Physical Capacitance: done by reducing device size reduces the current drive of the transistor making the circuit slow

What is our approach ?

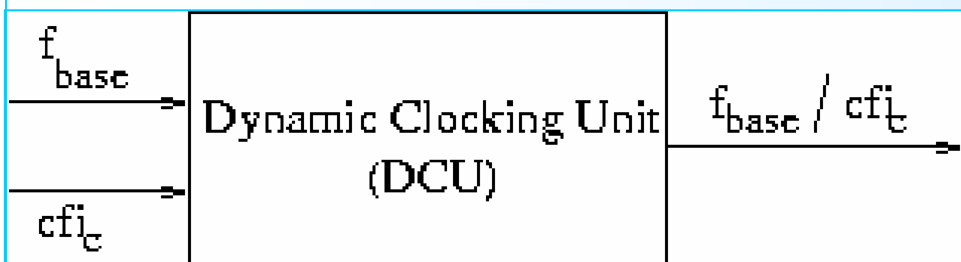
Adjust the frequency and reduce the supply voltage in a co-coordinated manner to reduce various forms dynamic power while maintaining performance, through datapath scheduling during behavioral synthesis.

Dynamic Frequency ??



Single Frequency

Dynamic Frequency

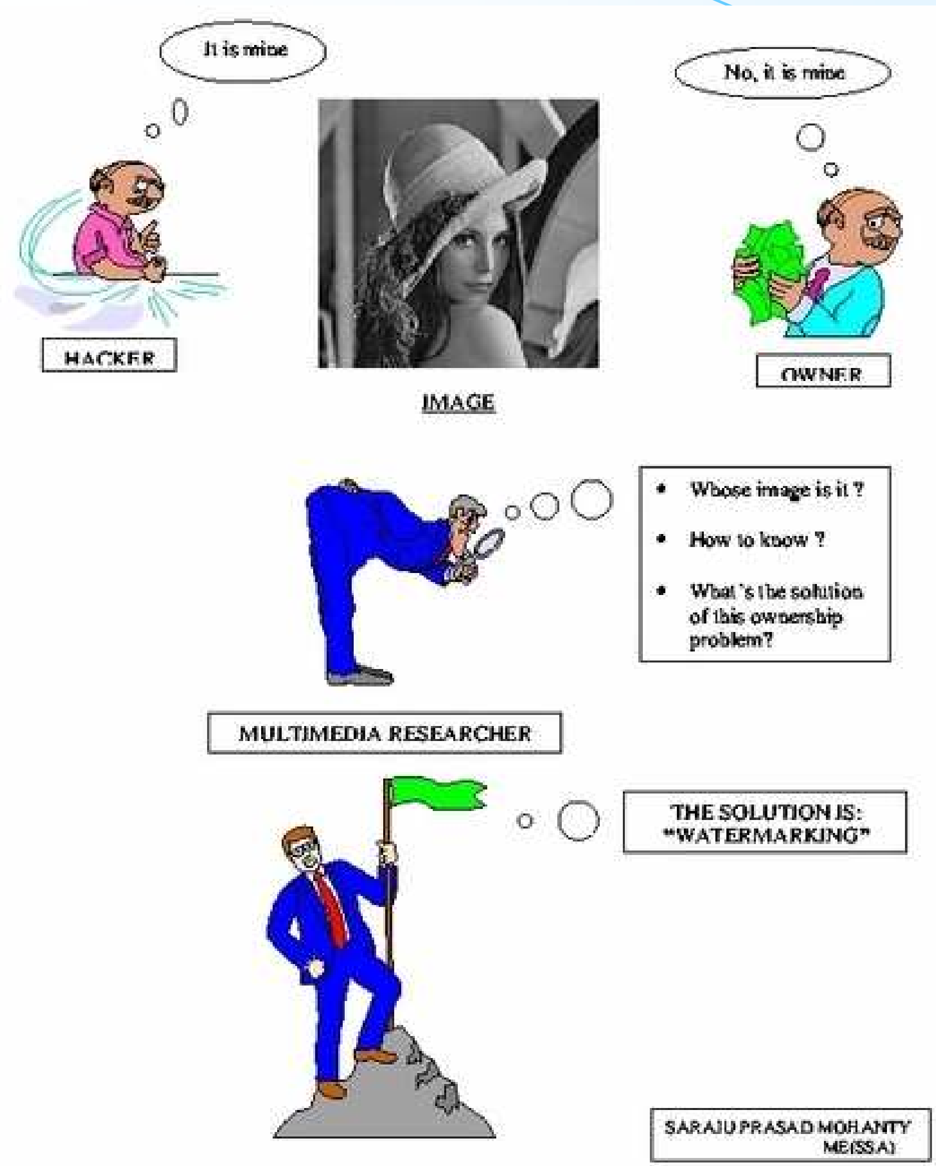


DCU uses clock divider strategy

More details :

- Ranganathan, et.al.
- Byrnjolfson and Zilic

Digital Watermarking ?

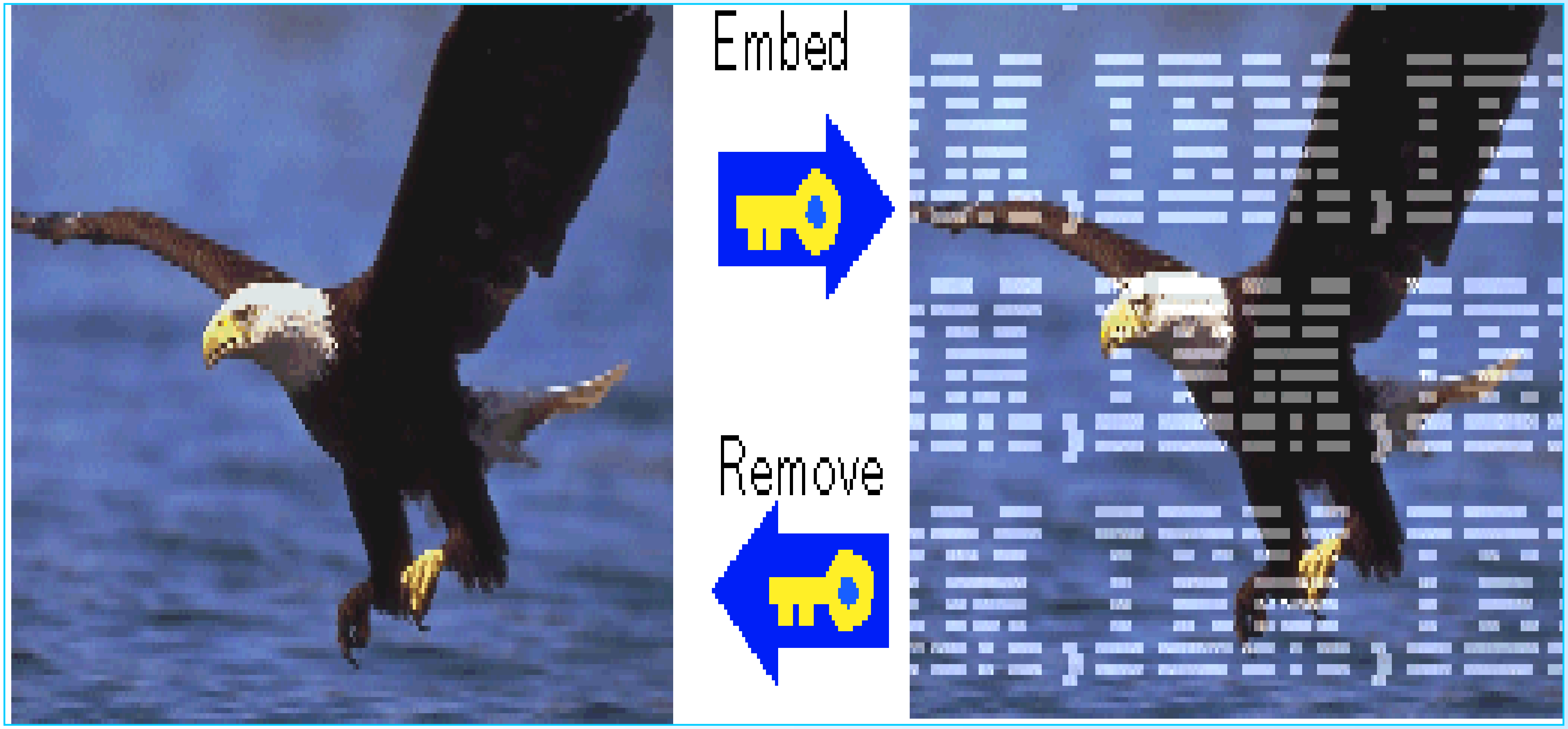


Digital watermarking is defined as a process of embedding data (watermark) into a multimedia object to help to protect the owner's right to that object.

Types

- Visible and Invisible
- Spatial, DCT and Wavelet domain
- Robust and Fragile

Digital Watermarking : Examples



Watermarking : General Framework

- Encoder: Inserts the watermark into the host image
- Decoder: Decodes or extracts the watermark from image
- Comparator: Verifies if extracted watermark matches with the inserted one

The Watermarking Encoders Designed are :

- Spatial domain invisible-robust and invisible-fragile watermarking encoder
- Spatial domain visible watermarking encoder
- DCT domain invisible and visible watermarking encoder (only architecture proposed)

Related Works

Low Power Synthesis / Watermarking

- Scheduling for Energy Minimization
- Switching Activity Reduction at Behavioral Level
- Datapath Scheduling for Peak Power Reduction
- Scheduling for Variable Voltage Processor
- Design and Synthesis of Variable Frequency/Latency and Multiple Voltage based Systems
- Hardware-based Watermarking Systems

Scheduling Schemes using Multiple Voltages

Table 2.1. Datapath Scheduling Schemes Using Multiple Supply Voltages

Proposed Scheme	Optimization Method Used	Constraints Assumed	Operating Voltage Levels	Time Complexity
Johnson and Roy [89, 90]	ILP	Time	$(5.0V \rightarrow 2.0V)$	Expoential
Johnson and Roy [6]	ILP	Time	$(5.0V, 3.3V, 2.4V)$	Expoential
Chang and Pedram [63, 91]	Dynamic Programming	Time	$(5.0V, 3.3V, 2.4V)$	Pseudo-Polynomial
Lin, Hwang and Wu [92]	ILP and Heuristic	Time and Resource	$(5.0V, 3.3V)$	Expoential $O\left(n^3 logn\right)$
Sarrafzadeh and Raje [93]	Dynamic Prog Geometric	Time and Resource	$(5.0V, 3.3V)$	$O\left(n^2 k\beta\|R\|^2\right)$ $O\left(nClognC\right)$
Kumar and Bayoumi [94, 95, 96]	Stochastic Evolution	Resource	$(5.0V, 3.3V, 2.4V)$	$O\left(n^2\right)$
Elgamel and Bayoumi [97]	Genetic Algorithms	Time and Area	$(5.0V, 3.3V, 2.4V)$	NA
Shiue and Chakrabarti [98, 99]	List-Based	Time and Resource	$(5.0V, 3.3V)$ or $(5.0V, 3.3V, 2.4V)$	Polynomial
Manzak and Chakrabarti [100]	Lagrangian Multiplier	Time and Resource	$(5.0V, 3.3V, 2.4V, 1.5V)$	$O\left(n^2\right)$ and $O\left(n^2 logL\right)$
Manzak and Chakrabarti [101]	List-Based	Time and Resource	$(5.0V, 3.3V, 2.4V, 1.5V)$	$O\left(r^2 L^2\right)$

None of these works :

•Handle variable frequency

•Minimize other forms of power

And

Most of the cases, the time penalty and area penalty are high.

Switching Reduction during Behavioral Synthesis

Table 2.2. High-Level Synthesis Schemes using Switching Activity Reduction

Proposed Work	Synthesis Tasks Performed	Methods Used	Time Complexity	% Power Reduction
Kumar, Katkoori, Rader and Vemuri [102, 103]	Scheduling, Register Optimization, etc.	Simulation of DFG	NA	NA
Raghunathan and Jha [104]	Tranformation, Scheduling and Allocation	Iterative Improvement	Polynomial	4.6
Musoll and Cortadella [50]	Scheduling and Resource Binding	List-Based Algorithm	$O(n^2 m)$	6.67
Lundberg, Muhammad, Roy and Wilson [112, 113]	NA	Hierarchical	NA	14.93
Shin and Lin [114]	Resource Allocation	Heuristic	Polynomial	7.84
Monteiro, Devadas, Ashar and Mauskar [116]	Scheduling	HYPER [115]	NA	22.43
Gupta and Katkoori [119]	Scheduling	Force-Directed Heuristic	$O(n^4 t)$	16.4
Murugavel and Ranganathan [120]	Scheduling Binding	Game Theory	Exponential	13.9

- These synthesis works neither handle multiple supply voltages nor variable frequency.
- Minimize average power only.
- Often accompanied by high time penalty.

Peak Power Reduction at Behavioral Level

Table 2.3. Relative Performance of Various Schemes Proposed for Peak Power Minimization

Proposed Work	Synthesis Tasks Performed	Methods Used	Time Complexity	% Power Reduction
Martin and Knight [53, 56]	Scheduling Assignment	Genetic Algorithms	NA	40.3-60.0
Shiue and et. al. [122, 123, 124, 111]	Scheduling	ILP Force Directed	Exponential $O(cn^3)$	50.0 – 75.0
Raghunathan, and et. al. [59]	Scheduling	Data Monitor Operations	NA	17.42-32.46

- Do not handle MV or DFC
- High time penalty
- Do not minimize other forms of power

Scheduling for Variable Frequency Processor

Table 2.4. Scheduling Algorithms for Variable Voltage Processor

Proposed Work	Working Level	Static or Dynamic	Method Used	Running Time	% Power Savings
Ishihara and Yasuura [125]	OS	Static	ILP	Exponential	70
Okuma, Ishihara, and Yasuura [126, 127]	OS	Static Dynamic	ILP Heuristic	Exponential NA	56 58
Hong, Potkonjak, and Srivastava [128]	OS	Dynamic	Heuristic	$O(N+m)$	20
Hong, Kirovski, and et. al. [129]	System	Static	Heuristic	$O(n^3)$	25
Mansour, Mansour, and et. al. [130]	Circuit and Behavioral	Static	List-based Heuristic	$O\left(n^4\right)$	56
Azevedo, Issenin, and Cornea [131, 132]	Compiler	Static	Heuristic	NA	82
Hsu, Kremer, and Hsiao [135, 136]	Compiler	Static	Heuristic	NA	70
Pering, Burd and Brodersen [69]	OS	Static	Heuristic	$O(n)$	80
Lee and [137] Krishna [137]	OS	Static Dynamic	Heuristic Heuristic	$O\left(n^2\left(\frac{T_{max}}{T_{min}}\right)\right)$ NA	54.5 65.6
Pouwelse, Langendoen, and Sips [64]	OS	Dynamic	Heuristic	$O\left(n^3\right)$	50
Yao, Demers, and Shenker [138]	OS and Circuit	Static Dynamic	Heuristic NA	$O\left(nlog^2n\right)$ NA	NA NA

- Handle variable frequency at OS or compiler level.
- Minimize average power or energy only.

Design and Synthesis using Variable Frequency

Table 2.5. Design and Synthesis Works on Variable Frequency or Multiple Frequency

Proposed Work	Design or Synthesis	Power or Performance	Operation Mode	Voltage or Frequency	Result
Usami, Igarashi, and et. al. [7, 75]	Design Synthesis	Low-Power	Multiple Voltage	$(3.3, 1.9)V$	47% (max)
Usami, Igarashi, and et. al. [74]	Design	Low-Power	Variable Voltage	NA	55% (max)
Ranganathan, and et. al. [8, 70]	Design	High Performance	Dynamic Frequency	$50 - 400MHz$	1.79-3.0 (times)
Krishna, and et. al. [144, 145]	Synthesis (Scheduling)	Low-Power	Dynamic Frequency	$(5.0, 3.3, 2.4)V$	$2 - 54\%$
Papachristou, and et. al. [146]	Synthesis (Allocation)	Low-Power	Multiple Frequency	NA	50% (max)
Burd, Brodersen, and et. al. [147, 148]	Design	Low-Power	Variable Voltage	$1.2 - 3.8V$	11% (avg)
Kim and Chae [72]	Design	Low-Power	Frequency Scaling	NA	NA
Pouwelse, and, et. al. [11]	Design	Low-power	Variable Frequency	$0.8 - 2.0V$ $59 - 251MHz$	NA
Acquaviva, Benini, and Riccò [149]	Design	Low-power	Variable Frequency	NA	40% (max)
Benini, and et. al. [150, 151]	Design Synthesis	High Performance	Variable Latency	NA	27%
Raghunathan, and et. al. [152]	Synthesis	High Performance	Variable Latency	NA	$1.6\times$
Nowka and [153, 154]	Design	Low-power	Frequency Scaling	$1.0 - 1.8V$	NA
Lu, Benini, and Michelli [155]	Design	Low-power	Frequency Scaling	$103 - 206MHz$	46% (max)

- Low-power or High-performance synthesis or design works using variable frequency.
- Minimize only average power.

Hardware Systems for Digital Watermarking

Table 2.6. Watermarking Chips Proposed in Current Literature

Proposed Work	Type of Watermark	Target Object	Working Domain	Techno-logy	Chip Area	Chip Power Consumption
Mathai and et. al. [161]	Invisible Robust	Video	Wavelet	0.18μ	NA	NA
Tsai and Lu [162]	Invisible Robust	Image	DCT	0.35μ	3.064×3.064 mm^2	$62.78mW$ $3.3V, 50MHz$
Garimella and et. al. [163]	Invisible Fragile	Image	Spatial	0.13μ	3453×3453 μm^2	$37.6\mu W$ $1.2V$

A lot needs to be done ……. ☺

In this Dissertation …….

Two design options explored

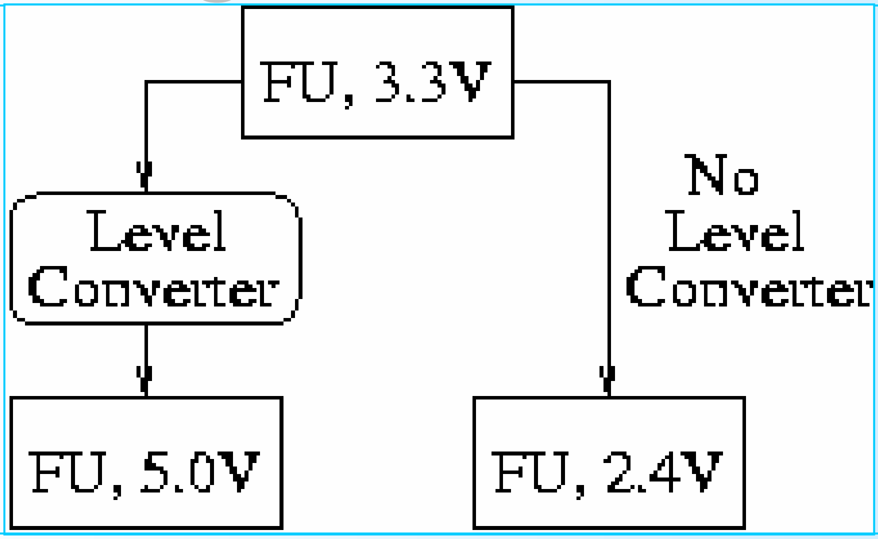
- Multiple Supply Voltages and Dynamic Frequency Clocking (MVDFC)
- Multiple Supply Voltages and Multicycling (MVMC)

Minimization during Behavioral Synthesis

- Energy or Energy-delay-product
- Peak power
- Simultaneous peak and average power
- Transient power
- Power fluctuation
- Framework for simultaneous minimization

Designing various watermarking chips

Target Architecture



- Level converters are used when a low-voltage functional unit is driving a high-voltage functional unit.
- Each functional unit has one register and one multiplexer.
- The register and the multiplexor operate at the same voltage level as that of the functional units.
- Operational delay of a FU : $(d_{FU} + d_{Mux} + d_{Reg} + d_{Conv})$.
- Time for voltage conversion equals to time for frequency change.
- Controller has a storage unit to store the cycle frequency index (cfi_c).
- Datapath is represented as a sequencing DFG.
- Operating frequencies are calculated from the delays.

A Framework for Simultaneous Minimization

CPF Minimization
(Different Power and Energy Parameters)

Aim at simultaneous minimization of:

- Average Power
- Peak power
- Cycle difference power
- Peak power differential
- Total Energy

NOTE: The peak power, the cycle difference power, and the peak power differential drive the transient characteristic of a CMOS circuit.

CPF Minimization: Power Definitions

- Cycle Power (P_c): power consumption of any control step.
- Peak Power (P_{peak}): maximum power consumption of any control step i.e. maximum (P_c).
- Mean Cycle Power (P): mean of the cycle powers (an estimate for the average power consumption of a DFG).
- Cycle Difference Power (DP_c): quantifies variation of power consumption of a cycle c from the mean /average power consumption. This determines the power profile of a DFG over all the control steps.
- Peak power differential (DP_{peak}): the maximum of the cycle difference power for any control step.
- Mean Cycle Difference Power (DP): mean of the cycle difference powers (a measure of overall power fluctuation)

CPF Minimization: Cycle Power Function

- We Define: A new parameter called "cycle power function" (CPF) as an equally weighted sum of the normalized mean cycle power and the normalized mean cycle difference power.

- We claim: The minimization of CPF using multiple supply voltages and dynamic frequency clocking (MVDFC), and multiple supply voltages and multicycling (MVMC) under resource constraints will lead to the reduction of energy and all different forms of power.

CPF Minimization: Power Models (Notations Needed)

Table 6.1. List of notataions and terminology used in CPF modeling

N	: total number of control steps in the DFG
O	: total number of operations in the DFG
c	: a control step or a clock cycle in the DFG
o_i	: any operation i, where $1 \leq i \leq O$,
P_c	: the total power consumption of all functional units active in control step c (cycle power consumption)
P_{peak}	: peak power consumption for the DFG equal to $max(P_c)_{\forall c}$
P	: mean power consumption of the DFG (average P_c over all control steps)
P_{norm}	: normalised mean power consumption of the DFG
DP_c	: cycle difference power (for cycle c; a measure of cycle power fluctuation)
DP_{peak}	: peak differential power consumption for the DFG equal to $max(DP_c)_{\forall c}$
DP	: mean of the cycle difference powers for all control steps in DFG
DP_{norm}	: normalised mean of the mean difference powers for all steps in DFG
CPF	: cycle power function
$FU_{k,v}$	: any functional unit of type k operating at voltage level v
FU_i	: any functional unit $FU_{k,v}$ needed by o_i for its execution ($o_i \in FU_{k,v}$)
$FU_{i,c}$	: any functional unit FU_i active in control step c
R_c	: total number of functional units active in step c (same as the number of operations scheduled in c)
$\alpha_{i,c}$	: switching activity of resource $FU_{i,c}$
$V_{i,c}$	: operating voltage of resource $FU_{i,c}$
$C_{i,c}$	: load capacitance of resource $FU_{i,c}$
f_c	: frequency of control step c

CPF Minimization: Power Model …

- The power consumption for any control step c is given by,

$$\mathbf{P_c = \sum_{i=\{1 \rightarrow Rc\}} \alpha_{i,c}\, C_{i,c}\, V^2_{i,c}\, f_c}$$

- The peak power consumption of the DFG is the maximum power consumption over all the control steps,

$$P_{peak} = \max\,(\,\mathbf{P_c}\,)_{c=\{1 \rightarrow N\}} = \max(\,\mathbf{\sum_{i=\{1 \rightarrow Rc\}} \alpha_{i,c}\, C_{i,c}\, V^2_{i,c}\, f_c}\,)_{c=\{1 \rightarrow N\}}$$

- Average power is characterized as mean cycle power **(P_c)** :

$$\mathbf{P} = 1/N\,(\textstyle\sum_{\mathbf{c=\{1 \rightarrow N\}}} \mathbf{P_c}) = 1/N\,(\textstyle\sum_{\mathbf{c=\{1 \rightarrow N\}}} \mathbf{\sum_{i=\{1 \rightarrow Rc\}} \alpha_{i,c}\, C_{i,c}\, V^2_{i,c}\, f_c}\,)$$

NOTE: The true average power is the energy consumption per cycle/second. The above **P** is an estimate of it.

CPF Minimization: Power Models …

Background Material

- For a set of n observations, $x_1, x_2, x_3, \ldots..x_n$, from a given distribution, the sample mean (which is an unbiased estimator for the population mean, μ) is $m = 1/n \Sigma_i x_i$.
- The absolute deviation of these observations is defined as $\Delta x_i = |x_i - m|$.
- The mean deviation of the observations is given by $MD = 1/n \Sigma_i |x_i - m|$.
- We model the cycle difference power DP_c as the absolute deviation of cycle power Pc from the mean cycle power P.
- Similarly, the mean difference power DP is modeled as mean deviation of the cycle power P_c.

CPF Minimization: Power Models …

•Normalized mean cycle power (P_{norm}) is defined as :

= mean cycle power consumption over all control steps / maximum power consumption in any control step

= Mean (P_c) / Maximum (P_c)

= P / P_{peak}

• Normalized mean cycle difference power (DP_{norm}) is defined as :

= mean cycle difference power over all control steps / maximum cycle difference power for any control step

= Mean (DP_c) / Maximum (DP_c)

= DP / DP_{peak}

CPF Minimization: Power Models …

- Cycle power function is defined as :

$$CPF = P_{norm} + DP_{norm} \qquad (1)$$

- In terms of peak cycle power and peak cycle difference power,

$$CPF = \frac{P}{P_{peak}} + \frac{DP}{DP_{peak}} = \frac{\frac{1}{N}\sum_{c=1}^{N} P_c}{P_{peak}} + \frac{\frac{1}{N}\sum_{c=1}^{N} |P - P_c|}{DP_{peak}} \qquad (2)$$

- Using the switching capacitance, voltage and frequency,

$$CPF = \frac{\frac{1}{N}\sum_{c=1}^{N}\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c}{max\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right)_{\forall c}} + \frac{\frac{1}{N}\sum_{c=1}^{N}\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right) - \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right|\right)}{max\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right) - \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right|\right)_{\forall c}}$$

CPF Minimization: Scheduling Algorithm

Input: Unscheduled data flow graph,

resource constraint,

allowable voltage levels,

number of allowable frequencies,

load capacitance of each resource,

delay of each functional units

Output: Scheduled data flow graph, base frequency, cycle frequency index, operating voltage for each operation

CPF Minimization: Scheduling Algorithm …

Step 1 : Calculate the switching activity at the each node through behavioral simulation of the DFG.

Step 2 : Construct a LUT of effective switching capacitance.

Step 3 : Find ASAP and ALAP schedules of the UDFG.

Step 4 : Determine the number of multipliers and ALUs at different operating voltages.

Step 5 : Modify both ASAP and ALAP schedules obtained in Step 1 using the number of resources found in Step 2.

Step 6 : No. of control steps = Max (ASAP steps, ALAP steps).

Step 7 : Find the vertices having non-zero mobility and vertices with zero mobility.

Step 8 : Use the CPF-Scheduler-Heuristics to assign the time stamp and operating voltage for the vertices, and the cycle frequencies such that CPF and time penalty are minimum (measures as T_D/T_S)

Step 10 : Calculate power, energy and frequency details.

CPF Minimization:
CPF-Scheduler Heuristic Explanations

- The heuristic is used to find proper time stamp, operating voltage for mobile vertices such that the $CPF+R_T$ is minimum for whole DFG.
- Initially assumes the modified ASAP schedule (with relaxed voltage resource constrained) as the current schedule.
- The $CurrentCPF+R_T$ value for the current schedule is calculated.
- The heuristic finds CPF values ($TempCPF+R_T$) for each allowable control step of each mobile vertices and for each available operating voltages.
- The heuristic fixes the time step, operating voltage and hence cycle frequency for which $CPF+R_T$ is minimum.

NOTE: The worst case running time of the heuristic is $\Theta(t_m|V|^3)$.

CPF Minimization: Experimental Results (Benchmarks and Resource Constraints used)

1. Auto-Regressive filter (ARF) (28 nodes, 16*, 12+, 40 edges).
2. Band-Pass filter (BPF) (29 nodes, 10*, 10+, 9-, 40 edges).
3. DCT filter (42 nodes, 13*, 29+, 68 edges).
4. Elliptic-Wave filter (EWF) (34 nodes, 8*, 26+, 53 edges).
5. FIR filter (23 nodes, 8*, 15+, 32 edges).
6. HAL diff. eqn. solver (11 nodes, 6*, 2+, 2-, 1< , 16 edges).

1. Number of multipliers: 1 at 2.4V; Number of ALUs: 1 at 3.3V
2. Number of multipliers: 2 at 2.4V; Number of ALUs: 1 at 3.3V
3. Number of multipliers: 2 at 2.4V; Number of ALUs: 1 at 2.4V and 1 at 3.3V
4. Number of multipliers: 1 at 2.4V and 1 at 3.3V ; Number of ALUs: 1 at 2.4V and 1 at 3.3V

CPF Minimization: Experimental Results (Notations used)

Table 6.2. Notations used to Express the Results

E_S	: total energy consumption assuming single frequency and single supply voltage
E_D	: total energy consumption for dynamic clocking and multiple supply voltage
P_{p_S}	: peak power consumption for single frequency and single supply voltage
P_{p_D}	: peak power consumption for dynamic clocking and multiple supply voltage
P_{m_S}	: minimum power consumption for single frequency and single supply voltage
P_{m_D}	: minimum power consumption for dynamic clocking and multiple supply voltage
T_S	: execution time assuming single frequency
T_D	: execution time assuming dynamic frequency
ΔE	: total energy reduction $= \frac{E_S - E_D}{E_S}$
ΔP	: average power reduction $= \frac{(E_S/T_S)-(E_D/T_D)}{(E_S/T_S)}$
ΔP_p	: peak power reduction $= \frac{P_{p_S} - P_{p_D}}{P_{p_S}}$
ΔDP	: differential power reduction $= \frac{(P_{p_S}-P_{m_S})-(P_{p_D}-P_{m_D})}{(P_{p_S}-P_{m_S})}$
R_T	: time ratio $= \frac{T_D}{T_S}$

CPF Minimization: Experimental Results

KT	RC	P_{pS} (mW)	P_{pD} (mW)	ΔP_p (%)	P_{mS} (mW)	P_{mD} (mW)	ΔDP (%)	ΔP (%)	ΔE (%)	N	r_T
	1	9.30	2.83	69.60	0.26	0.52	74.50	71.40	47.57	18	1.6
A	2	18.33	4.77	73.96	0.26	0.52	76.47	68.30	47.57	13	1.4
R	3	18.59	4.84	73.96	0.26	0.52	76.44	71.72	49.87	11	1.5
F	4	18.59	7.26	60.96	0.26	0.52	63.25	59.10	29.49	11	1.5
	1	9.30	2.45	73.62	0.26	0.52	78.64	65.80	46.69	17	1.3
B	2	18.33	4.20	77.10	0.26	1.67	86.03	58.81	46.69	17	1.2
P	3	18.59	4.84	73.96	0.52	0.97	78.59	71.09	48.61	9	1.4
F	4	18.59	7.33	60.60	0.52	0.97	64.84	64.01	32.02	9	1.4
	1	9.30	2.83	69.60	0.26	0.52	74.50	50.90	42.44	29	1.1
D	2	9.30	2.83	69.60	0.26	0.52	74.50	50.90	42.44	29	1.1
C	3	18.59	4.84	73.96	0.26	0.40	75.75	67.70	42.93	15	1.4
T	4	18.59	7.61	59.05	0.26	0.40	60.63	65.19	38.49	15	1.4
	1	9.30	2.45	73.62	0.26	0.52	78.64	41.17	44.43	27	0.9
E	2	18.07	4.07	77.49	0.26	0.52	80.09	37.49	44.43	27	0.9
W	3	18.07	4.07	77.49	0.26	0.40	79.38	57.89	44.73	16	1.2
F	4	18.07	6.55	63.75	0.26	0.40	65.49	53.10	38.45	16	1.2
	1	9.30	2.74	70.52	0.26	0.52	75.45	58.54	46.11	15	1.3
F	2	9.30	2.74	70.52	0.26	0.52	75.45	58.54	46.11	15	1.3
I	3	18.59	4.77	74.32	0.26	0.40	76.12	51.21	46.77	11	1.0
R	4	18.59	7.04	62.15	0.24	0.40	63.77	40.69	27.21	11	1.2
Average values				**70.52**			**75.04**	**59.59**	**43.29**		**1.3**

CPF Minimization: Power Profiles for RC2

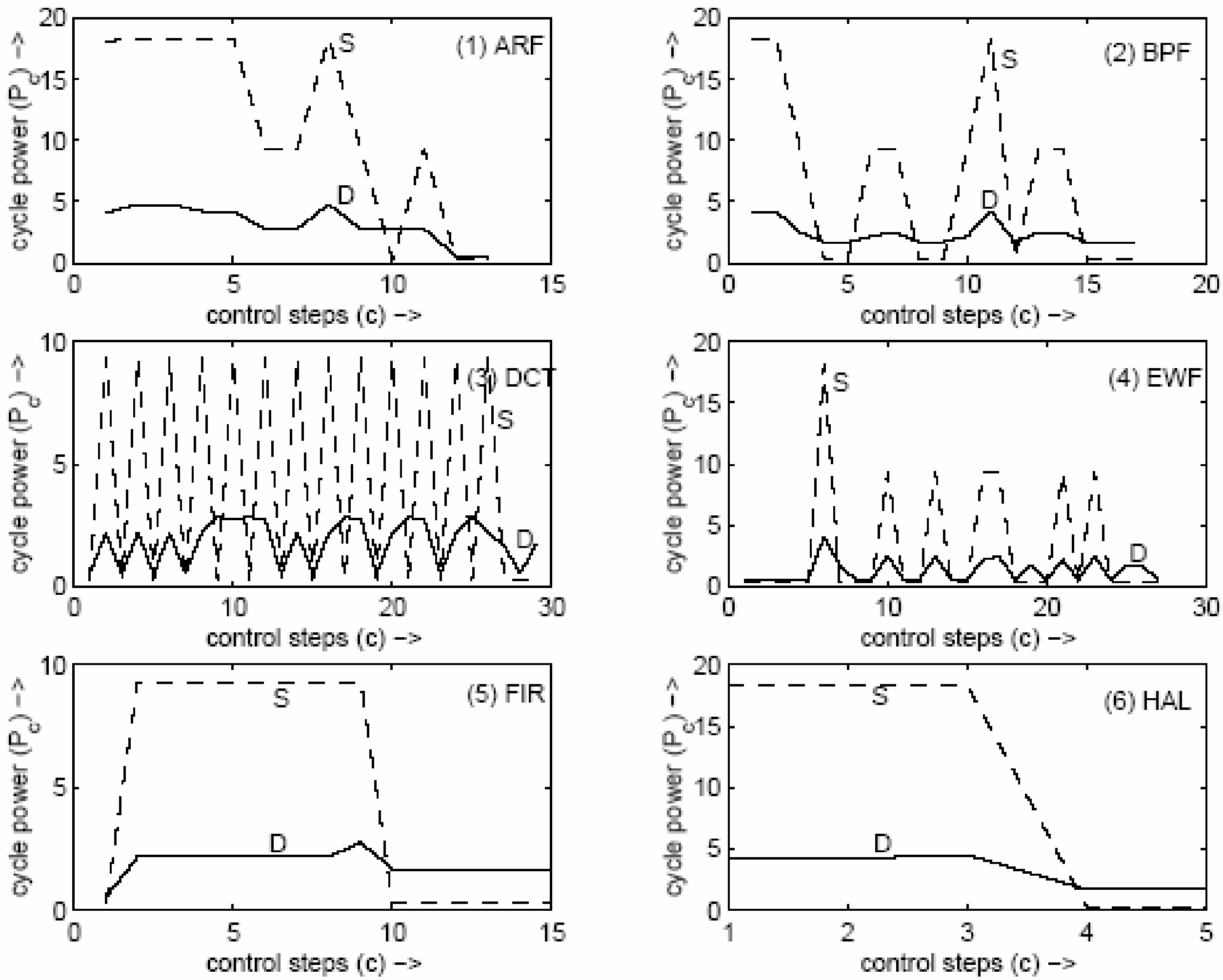


Figure 6.4. Cycle power consumptions for resource constraint RC2

CPF Minimization: Power Profiles for RC3

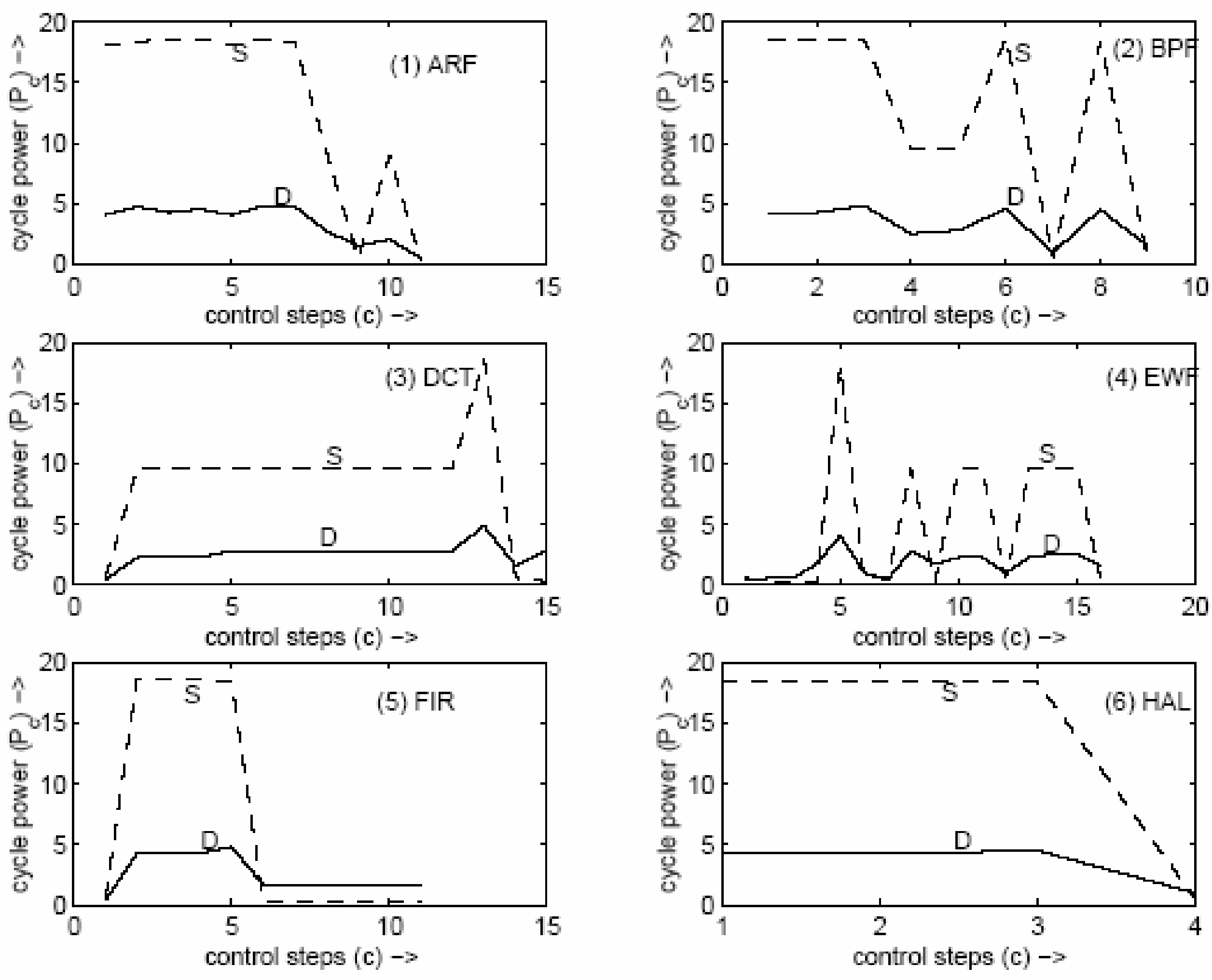


Figure 6.5. Cycle power consumptions for resource constraint RC3

CPF Scheduler Vs Proposed Scheduling Algorithms Available in the Literature

Works	Energy savings	Time penalty	Transient power, etc.
Change and Pedram [15]	40% on average	50% on average	Not addressed
Shiue and Chakrabarti [20]	56% on average	50% on average	Not addressed
Johnson and Roy [14]	46 – 58%	50% on average	Not addressed
Johnson and Roy [13]	0 – 50%	Not available	Not addressed
This work	43% in average	30% on average	70% reduction in peak 75% reduction in differntial

From the above table it is evident that our scheme has less time penalty compared to other popular energy minimization works. Additionally, we have appreciable reductions in transient powers, which the above mentioned works do not address.

ILP-based Framework for Simultaneous Minimization

CPF* Minimization

- Aim: to provide ILP-based minimization for the CPF defined in the previous chapter.
- Two different design options: MVDFC and MVMC
- Observations about CPF:
  - CPF is a *non-linear* function.
  - A function of four parameters, such as, P, P_{peak}, DP and DP_{peak}.
  - The absolute function in the numerator contributes to the nonlinearity.
  - The complex behavior of the function is also contributed by the two different denominator parameters, P_{peak} and DP_{peak}.
- Non-linear programming may be more suitable, but will be large space and time complexity. We are addressing linear programming of the non-linear function.

CPF* Minimization
(Linear Modeling of Nonlinearity)

General LP Formulations involving Absolute

- General form of this type of programming:

$$\begin{aligned} \text{Minimize :} \quad & \textstyle\sum_i |y_i| \\ \text{Subject to :} \quad & y_i + \textstyle\sum_j a_{ij} * x_j \leq b_i, \ \forall i \text{ and } x_j \geq 0, \ \forall j \end{aligned} \quad (1)$$

- Let y_i be expressed as, $y_i = y^1_i - y^2_i$, difference of two non-negative variables.
- After algebraic manipulations using these new variables we have the following model.

$$\begin{aligned} \text{Minimize :} \quad & \textstyle\sum_i y_i^1 + y_i^2 \\ \text{Subject to :} \quad & y_i^1 - y_i^2 + \textstyle\sum_j a_{ij} * x_j \leq b_i, \quad \forall i \\ & x_j \geq 0, \quad \forall j \text{ and } y_i^1, y_i^2 \geq 0, \quad \forall i \end{aligned} \quad (2)$$

CPF* Minimization
(Linear Modeling of Nonlinearity …)

General LP Formulations involving Fraction

- General form of this type of programming:

$$\text{Minimize}: \quad \frac{\sum_j c_j * x_j}{\sum_j d_j * x_j}$$
$$\text{Subject to}: \quad \sum_j a_{ij} * x_j \le b_i, \ \forall i, \quad x_j \ge 0, \ \forall j \qquad (1)$$

- Assume two new variables, $z_0 = 1/(d_0+\Sigma_i d_j x_j)$ and $x_j = z_j / z_0$.
- Using the new variables the formulation becomes.

$$\text{Minimize}: \quad c_0 * z_0 + \sum_j c_j * z_j$$
$$\text{Subject to}: \quad \sum_j a_{ij} * z_j - b_i * z_0 \le b_i, \ \forall i \qquad (2)$$
$$\sum_j d_j * z_j + d_0 * z_0 = 1, \quad z_0, z_j \ge 0, \ \forall j$$

- Once the new formulation is solved substitute $z_j = x_j * z_0$ to get the result for x_j.

CPF* Minimization
(Linear Modeling of Nonlinearity …)

What we learnt from the previous slides ??

- The objective function CPF has both types of nonlinearities.
- In case of a fraction: remove the denominator and introduce as constraints.
- In case of absolute: change difference in objective function to sum and introduce the difference as constraints.

CPF* Minimization
(Modified Cycle Power Function)

- The CPF has two different denominators which may lead to increase in number of constraints and hence the overall solution space.
- We assume that $|P-P_c|$ is upper bounded by P_c for all c, since $|P-P_c|$ is a measure of the mean difference error of P_c. So, instead of normalizing DP with DP_{peak}, we will normalize it with P_{peak}. This reduces the number of denominator to one.
- We have the following Modified Cycle Power Function which is the objective function for the ILP formulation.

$$CPF^* = \frac{P}{P_{peak}} + \frac{DP}{P_{peak}} = \frac{P + DP}{P_{peak}} = \frac{\frac{1}{N}\sum_{c=1}^{N} P_c + \frac{1}{N}\sum_{c=1}^{N} |P - P_c|}{P_{peak}}$$

$$= \frac{\frac{1}{N}\sum_{c=1}^{N}\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c}{max\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right)_{\forall c}} + \frac{\frac{1}{N}\sum_{c=1}^{N}\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right) - \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right|\right)}{max\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right)_{\forall c}}$$

CPF* Minimization: ILP Formulation (Notations)

- $M_{k,v}$: maximum number of functional units of type $F_{k,v}$
- S_i : as soon as possible time stamp for the operation o_i
- E_i : as late as possible time stamp for the operation o_i
- $P(C_{swi},v,f)$: power consumption of any $F_{k,v}$ used by operation o_i
- $x_{i,c,v,f}$: decision variable, which takes the value of 1 if operation o_i is scheduled in control step c using $F_{k,v}$ and c has frequency f
- $y_{i,v,l,m}$: decision variable which takes the value of 1 if operation oi is using the functional unit $F_{k,v}$ and scheduled in control steps l→m
- $L_{i,v}$: latency for operation o_i using resource operating at voltage v (in terms of number of clock cycles)

NOTE: The effective switching capacitance is a function of the average switching activity at the input operands of a functional unit and C_{swi} is a measure of effective switching capacitance FU_i.

$$\alpha_i C_i = C_{sw_i}(\alpha_i{}^1, \alpha_i{}^2)$$

CPF* Minimization: ILP Formulation

MVDFC Design Scenario

- Objective Function: Minimize the CPF* for the whole DFG over all the control steps. Using the previous expressions we have,

$$\text{Minimize}: \frac{\frac{1}{N}\sum_{c=1}^{N} P_c + \frac{1}{N}\sum_{c=1}^{N} |P - P_c|}{P_{peak}} \quad (1)$$

The denominator is removed and introduced as a constraint.

$$\begin{aligned}\text{Minimize}: &\ \frac{1}{N}\sum_{c=1}^{N} P_c + \frac{1}{N}\sum_{c=1}^{N} |P - P_c| \\ \text{Subject to}: &\ \text{Peak power constraints}\end{aligned} \quad (2)$$

The absolute is replaced with sum and the appropriate constraints.

$$\begin{aligned}\text{Minimize}: &\ \frac{1}{N}\sum_{c=1}^{N} P_c + \frac{1}{N}\sum_{c=1}^{N} (P + P_c) \\ \text{Subject to}: &\ \text{Modified peak power constraints}\end{aligned} \quad (3)$$

After simplification,

$$\begin{aligned}\text{Minimize}: &\ \left(\frac{3}{N}\right)\sum_{c=1}^{N} P_c \\ \text{Subject to}: &\ \text{Modified peak power constraints}\end{aligned} \quad (4)$$

Using decision variables,

$$\begin{aligned}\text{Minimize}: &\ \sum_{c}\sum_{i\in F_{k,v}}\sum_{v}\sum_{f} x_{i,c,v,f} * \left(\frac{3}{N}\right) * P(C_{swi}, v, f) \\ \text{Subject to}: &\ \text{Modified peak power constraints}\end{aligned} \quad (5)$$

CPF* Minimization: ILP Formulation (MVDFC)

- **Uniqueness Constraints** : ensure that every operation o_i is scheduled to one unique control step and represented as,

  $$\forall i,\ 1 \le i \le O,\ \Sigma_c \Sigma_v \Sigma_f\, x_{i,c,v,f} = 1$$

- **Precedence Constraints** : guarantee that for an operation o_i , all its predecessors are scheduled in an earlier control step and its successors are scheduled in an later control step and are; $\forall i,j$, o_i belong to $Pred(o_j)$, $\Sigma_v \Sigma_f \Sigma_{\{d=S_i \rightarrow E_i\}} dx_{i,c,v,f} - \Sigma_v \Sigma_f \Sigma_{\{d=S_j \rightarrow E_j\}} ex_{j,c,v,f} \le -1$

- **Resource Constraints** : make sure that no control step contains more than $F_{k,v}$ operations of type k operating at voltage v and are enforced as, $\forall c, 1 \le c \le N$ and $\forall v$, $\Sigma_{\{i \varepsilon F_{k,v}\}} \Sigma_f\, x_{i,c,v,f} \le M_{k,v}$

- **Frequency Constraints** : lower operating voltage functional unit can not be scheduled in a higher frequency control step; these constraints are expressed as,

  $$\forall i,\ 1 \le i \le O,\ \forall c,\ 1 \le c \le N,\ \text{if } f < v, \text{ then } x_{i,c,v,f} = 0.$$

CPF* Minimization: ILP Formulation (MVDFC)

- Peak Power Constraints : introduced to eliminate the fractional non-linearity of the objective function and are enforced as, for all c, 1<= c <= N,

$$\sum_{i \in F_{k,v}} \sum_{v} \sum_{f} x_{i,c,v,f} * P(C_{swi}, v, f) \leq P_{peak}$$

- Modified Peak Power Constraints : To eliminate the non-linearity introduced due to the absolute function introduced as, for all c, 1<= c <= N,

$$\frac{1}{N} \sum_{c} \sum_{i \in F_{k,v}} \sum_{v} \sum_{f} x_{i,c,v,f} * P(C_{swi}, v, f) - \sum_{i \in F_{k,v}} \sum_{v} \sum_{f} x_{i,c,v,f} * P(C_{swi}, v, f) \leq P^{*}_{peak}$$

NOTE: The unknowns P_{peak} and P^{*}_{peak} is added to the objective function and minimized alongwith it.

CPF* Minimization: ILP Formulation

MVMC Design Scenario

- Objective Function: Following the same steps as in the MVDFC case in terms of decision variables we write,

$$\text{Minimize}: \sum_{l} \sum_{i \in F_{k,v}} \sum_{v} y_{i,v,l,(l+L_{i,v}-1)} * \left(\frac{3}{N}\right) P(C_{sw_i}, v, f_{clk})$$

Subject to : Modified peak power constraints

- Uniqueness Constraints: ensure that every operation o_i is scheduled to appropriate control steps within the range (S_i, E_i) and represented as, $\forall$ i, $1 \leq i \leq O$,

$$\Sigma_v \Sigma_{\{l=S_i \rightarrow (S_i+E_i+1-L_{i,v})\}} y_{i,v,l,(l+L_{i,v}-1)} = 1$$

- Precedence Constraints : guarantee that for an operation o_i , all its predecessors are scheduled in an earlier control step and its successors are scheduled in an later control step; $\forall$ i,j, o_i belong to Pred(o_j),

$$\Sigma_v \Sigma_{\{l=S_i \rightarrow E_i\}} (l+L_{i,v}-1) y_{i,v,l,(l+Li,v-1)} - \Sigma_v \Sigma_{\{l=S_j \rightarrow E_j\}} l y_{j,v,l,(l+L_{j,v}-1)} \leq -1$$

CPF* Minimization: ILP Formulation (MVMC)

- Resource Constraints : make sure that no control step contains more than $F_{k,v}$ operations of type k operating at voltage v and are enforced as,

$$\Sigma_{\{i \varepsilon F_{k,v}\}} \Sigma_l \, y_{i,v,l,(l+L_{i,v}-1)} \leq M_{k,v}$$

- Peak Power Constraints : introduced to eliminate the fractional non-linearity of the objective function and are enforced as, for all c, 1<= l <= N,

$$\sum_{i \in F_{k,v}} \sum_{v} y_{i,v,l,(l+L_{i,v}-1)} * P(C_{sw\,i}, v, f_{clk}) \leq P_{peak}$$

- Modified Peak Power Constraints : To eliminate the non-linearity introduced due to the absolute function introduced as, for all c, 1<= l <= N,

$$\frac{1}{N} \sum_{l} \sum_{i \in F_{k,v}} \sum_{v} y_{i,v,l,(l+L_{i,v}-1)} * P(C_{sw\,i}, v, f_{clk}) - \sum_{i \in F_{k,v}} \sum_{v} y_{i,v,l,(l+L_{i,v}-1)} * P(C_{sw\,i}, v, f_{clk}) \leq P^*_{peak}$$

CPF* Minimization: Scheduling Algorithm

Step 1: Construct a look up table for (effective switching capacitance, average switching activity) pairs.

Step 2: Calculate the switching activities at the inputs of each node through behavioral simulation of the DFG.

Step 3: Find ASAP schedule for the UDFG.

Step 4: Find ALAP schedule for the UDFG.

Step 5: Determine the mobility graph of each node.

Step 6: Modify the mobility graph for MVMC.

Step 7: Model the ILP formulations of the DFG for MVDFC or MVMC scheme using AMPL.

Step 8: Solve the ILP formulations using LP-Solve.

Step 9: Find the scheduled DFG.

Step 10: Determine the cycle frequencies, cycle frequency index and base frequency for MVDFC scheme.

Step 11: Estimate power and energy consumptions of the scheduled DFG.

CPF* Minimization: Experimental Results (Benchmarks and Resource Constraints used)

1. Example circuit (EXP) (8 nodes, 3*, 3+, 9 edges)
2. FIR filter (11 nodes, 5*, 4+, 19 edges)
3. IIR filter (11 nodes, 5*, 4+, 19 edges)
4. HAL differential eqn. solver (13 nodes, 6*, 2+, 2-, 1 <, 16 edges)
5. Auto-Regressive filter (ARF) (15 nodes, 5*, 8+, 19 edges)

Multipliers		ALUs		Serial No
2.4V	3.3V	2.4V	3.3V	
2	1	1	1	RC1
3	0	1	1	RC2
2	0	0	2	RC3
1	1	0	1	RC4

CPF* Minimization: Experimental Results …

	RC	P_{pS}	P_{pD}	ΔP_p	P_{mS}	P_{mD}	ΔDP	P_S	P_D	ΔP	E_S	E_D	ΔE
		mW	mW	%	mW	mW	%	mW	mW	%	nJ	nJ	%
E	1	17.28	4.56	73.61	0.46	0.35	74.97	8.87	2.42	72.72	2.96	1.57	46.8
X	2	17.28	4.56	73.61	0.46	0.35	74.97	8.87	2.42	72.72	2.96	1.57	46.8
P	3	17.28	4.56	73.61	0.46	0.9	78.24	8.87	2.61	70.57	2.96	1.6	46.0
F	1	17.51	4.62	73.62	0.23	0.12	73.96	8.82	2.35	73.36	4.9	2.6	47.20
I	2	25.92	6.84	73.61	0.23	0.12	73.84	8.82	2.36	73.24	4.9	2.6	47.20
R	3	17.51	4.67	73.33	0.23	0.45	75.58	8.82	2.5	71.66	4.9	2.6	46.22
H	1	17.51	4.62	73.62	0.46	0.35	74.96	13.25	3.55	73.21	5.9	3.12	47.0
A	2	26.15	6.90	73.61	0.46	0.35	74.50	13.25	3.55	73.21	5.9	3.12	47.0
L	3	17.74	4.78	73.05	0.46	0.9	76.97	13.25	3.73	71.85	5.9	3.17	46.2
I	1	25.92	8.88	65.74	0.23	0.12	65.9	11.03	3.5	68.36	4.9	3.05	37.7
I	2	25.92	6.84	73.61	0.23	0.12	73.84	11.03	2.98	72.98	4.9	2.6	47.96
R	3	17.51	4.67	73.34	0.23	0.45	75.58	8.82	2.57	70.86	4.9	2.64	46.22
A	1	8.87	2.34	73.62	0.23	0.12	74.1	4.5	1.22	72.9	5.0	2.64	47.2
R	2	8.87	2.34	73.62	0.23	0.12	74.1	4.5	1.22	72.9	5.0	2.64	47.2
F	3	8.87	2.39	73.05	0.23	0.45	77.6	4.5	1.4	68.9	5.0	2.74	45.3

CPF* Minimization: Experimental Results …

MVDFC Vs MVMC % Reduction

Power	MVDFC	MVMC
Peak Power	71.70	26.44
Peak Power Differential	74.0	26.73
Average Power	70.82	22.52
Energy	44.36	39.05
Energy Delay Product	17.31	17.99

CPF* Minimization: Power Profile for RC2

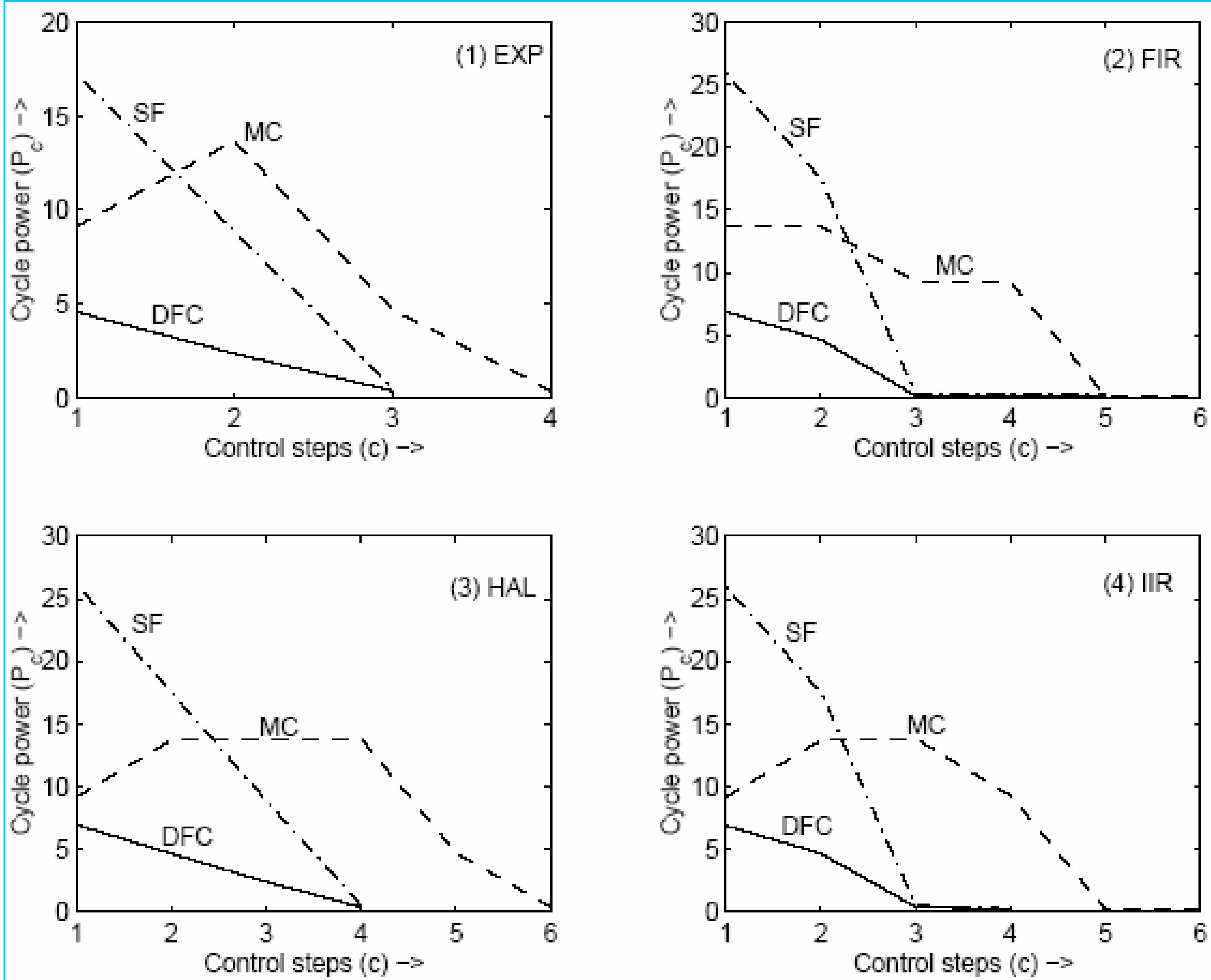


Figure 7.8. Power profile for benchmark for resource constraint RC2

CPF* Minimization: Power Profile for RC3

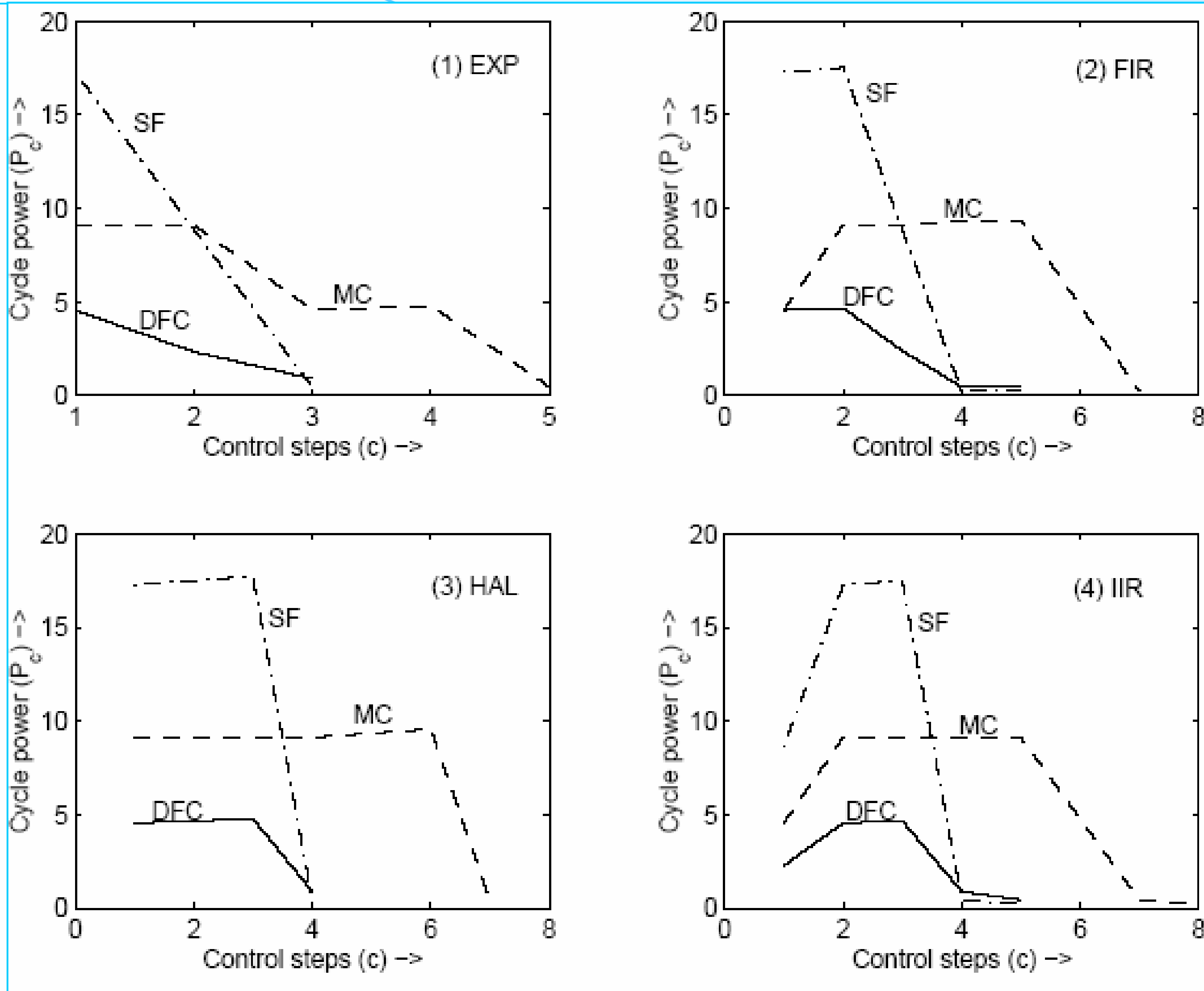


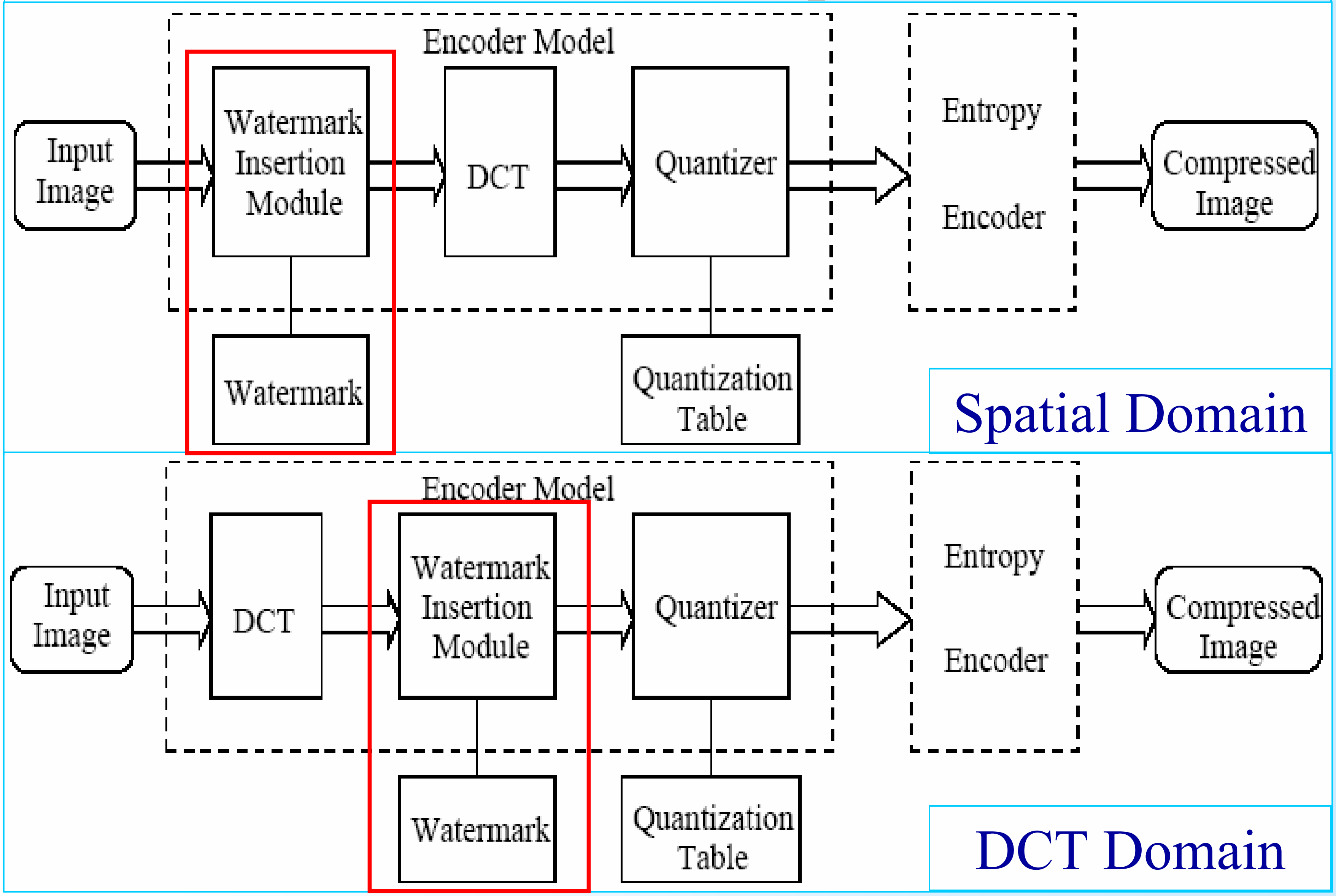
Figure 7.9. Power profile for benchmark for resource constraint RC3

Watermarking Chip Design

1. Architecture and implementation of spatial invisible
2. Architecture and implementation of spatial visible
3. Architecture for DCT invisible and visible (dual voltage and dual frequency operation)

NOTE: Detailed implementation of the DCT domain watermarking chip is being carried out by Karthik, a masters student, as a part of his thesis.

Secure JPEG Encoder (Spatial Vs DCT)



Digital Still Camera

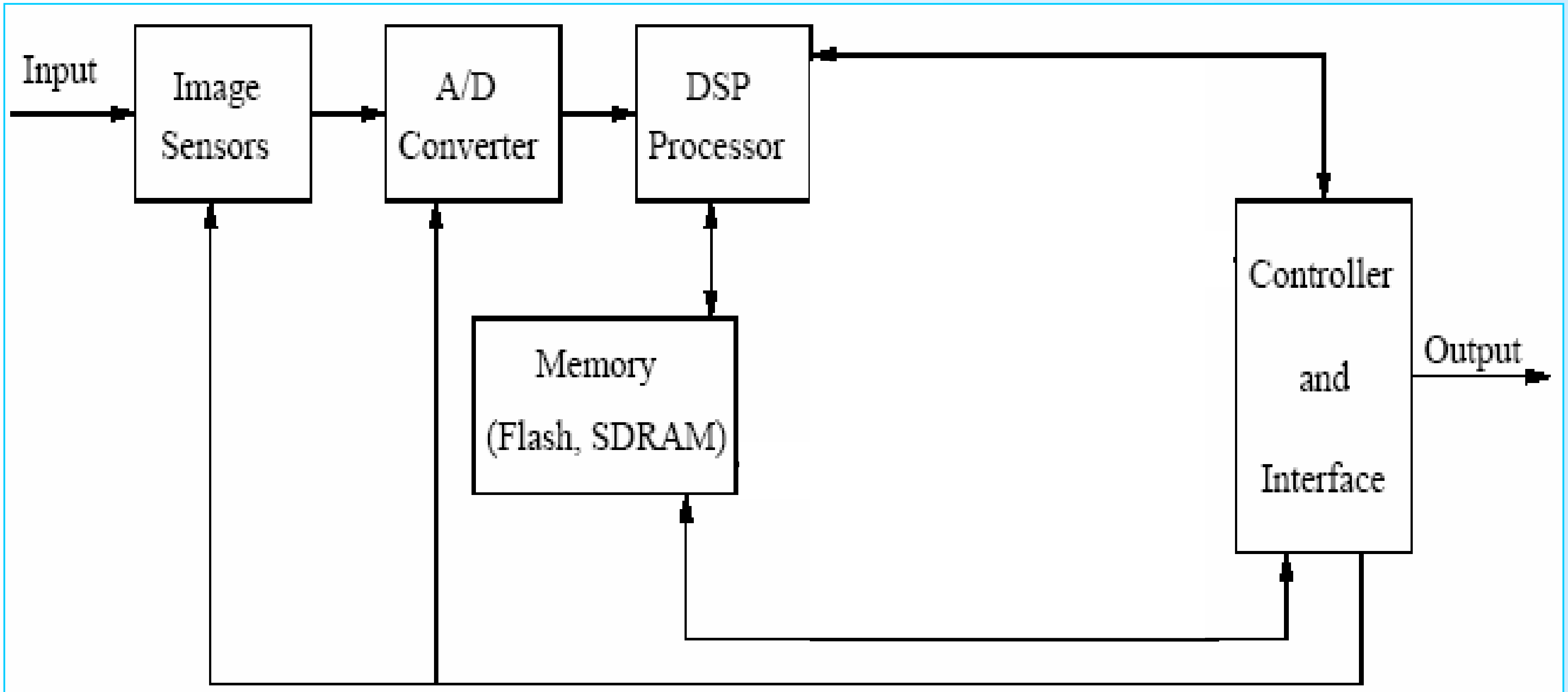


Figure 9.2. Secure Digital Still Camera : Schematic View

Secure Digital Still Camera

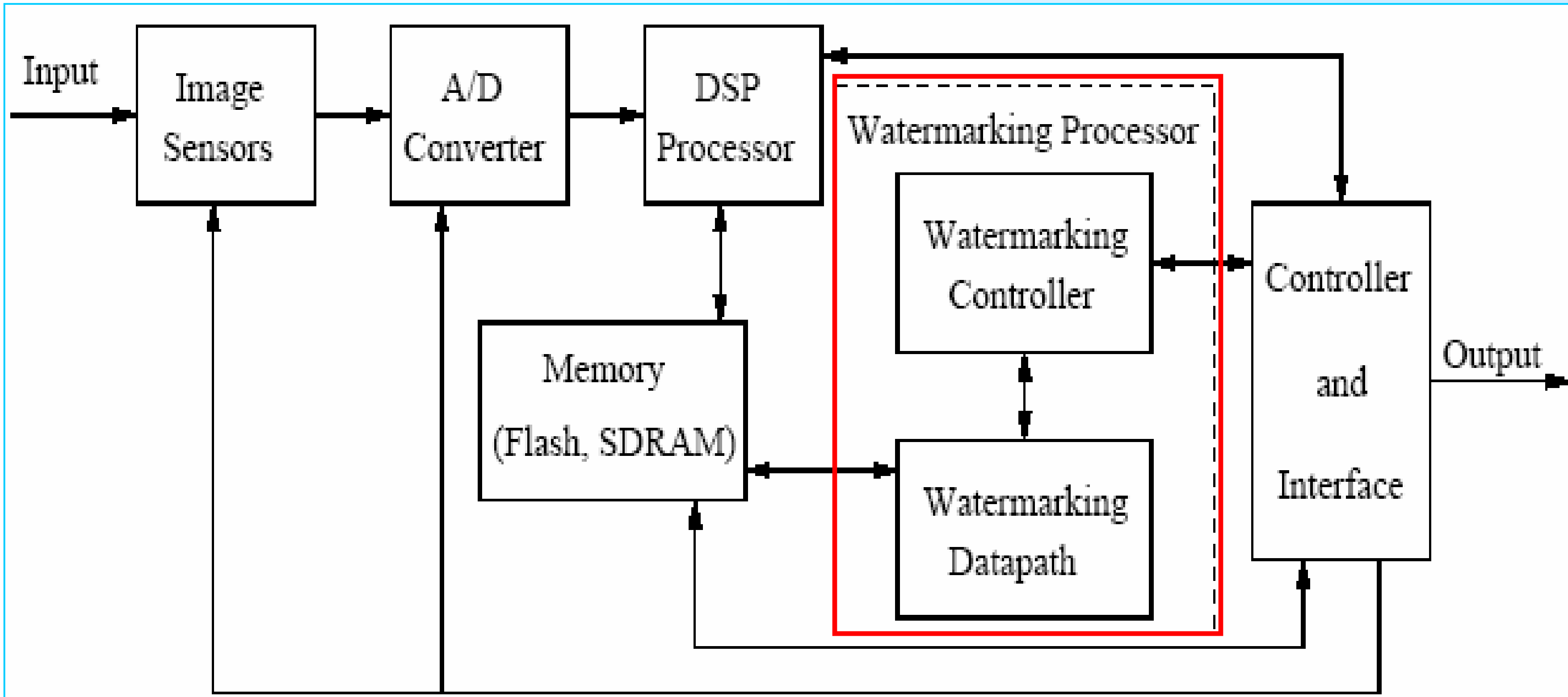
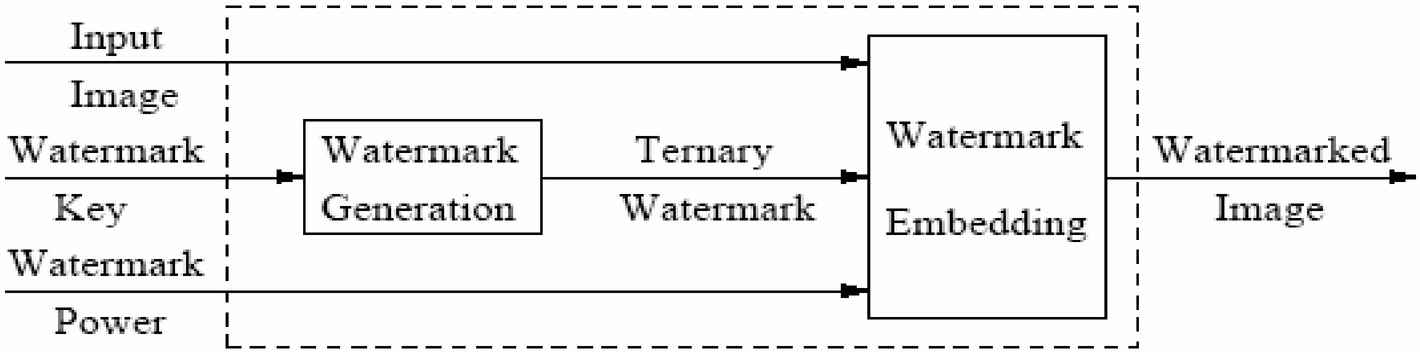


Figure 9.2. Secure Digital Still Camera : Schematic View

Spatial Invisible: Algorithm (Robust)



Watermark insertion

Table 9.1. Notations used to Explain Spatial Domain Watermarking Algorithms

I	: Original image (gray image)
W	: Watermark image (binary or ternary image)
(i, j)	: A pixel location
I_W	: Watermarked image
$N_I \times N_I$	: Image dimension
$N_W \times N_W$	: Watermark dimension
E, E_1, E_2	: Watermark embedding functions
D	: Watermark detection function
r	: Neighborhood radius
I_N	: Neighborhood image (gray image)
K	: Digital (watermark) key
α_1, α_2	: Scaling constants (watermark strength)

Spatial Invisible: Algorithm (Robust) …

- The watermark is a ternary image having pixel values {0,1,2}.
- Insertion: Alter the original image pixels as,

$$I_W(i,j) = \begin{cases} I(i,j) & \text{if } W(i,j) = 0 \\ E_1(I(i,j), I_N(i,j)) & \text{if } W(i,j) = 1 \\ E_2(I(i,j), I_N(i,j)) & \text{if } W(i,j) = 2 \end{cases}$$

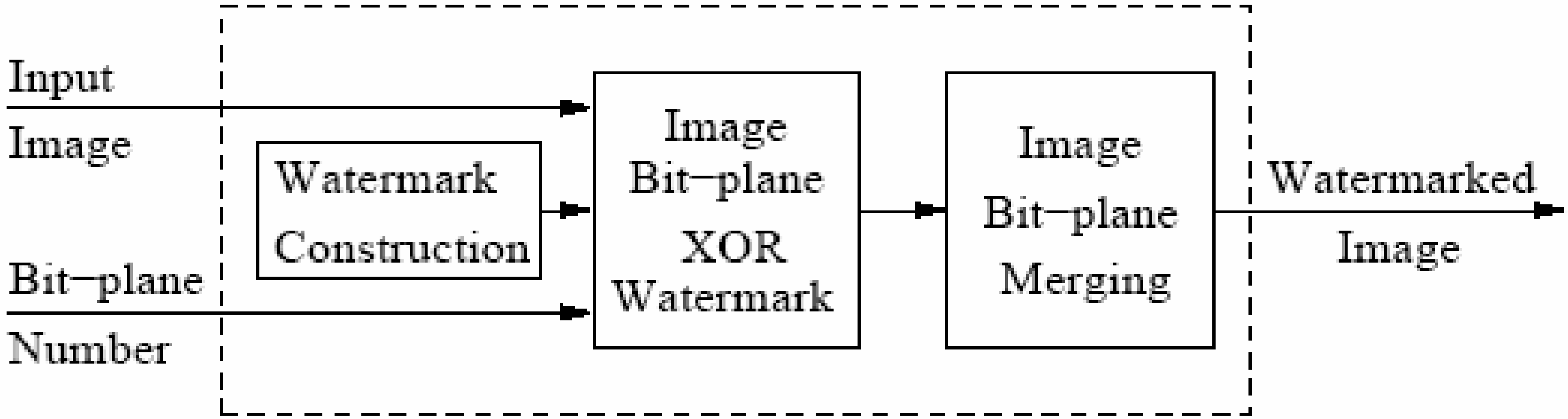
- Encoding function:

$$\begin{aligned} E_1(I, I_N) &= (1-\alpha_1)I_N(i,j) + \alpha_1 I(i,j) \\ E_2(I, I_N) &= (1-\alpha_1)I_N(i,j) - \alpha_2 I(i,j) \end{aligned}$$

- Neighborhood pixel gray value: Calculated as,

$$I_N(i,j) = \frac{\frac{I(i+1,j)+I(i+1,j+1)}{2} + I(i,j+1)}{2}$$

Spatial Invisible: Algorithm (Fragile)

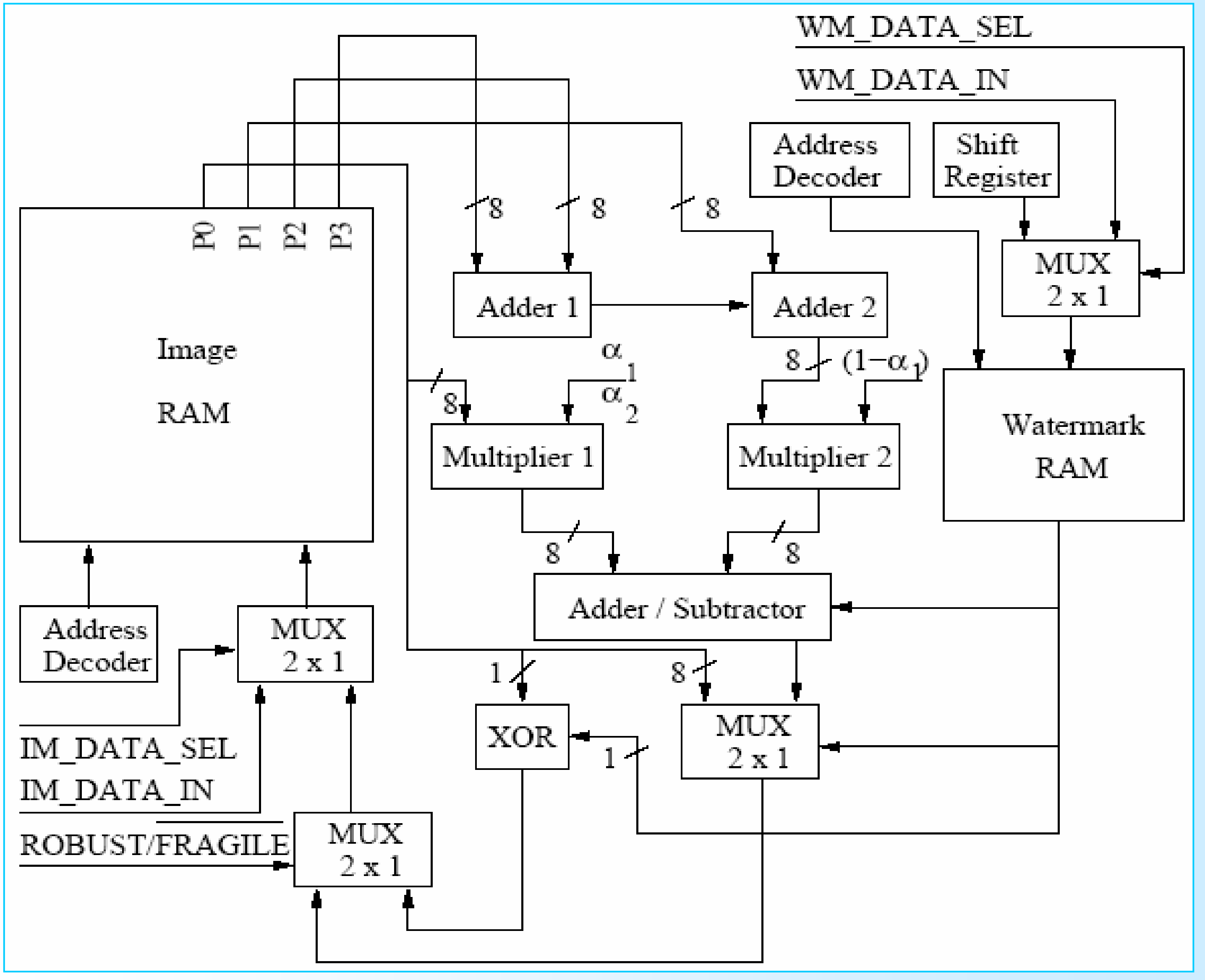


(a) Watermark Insertion

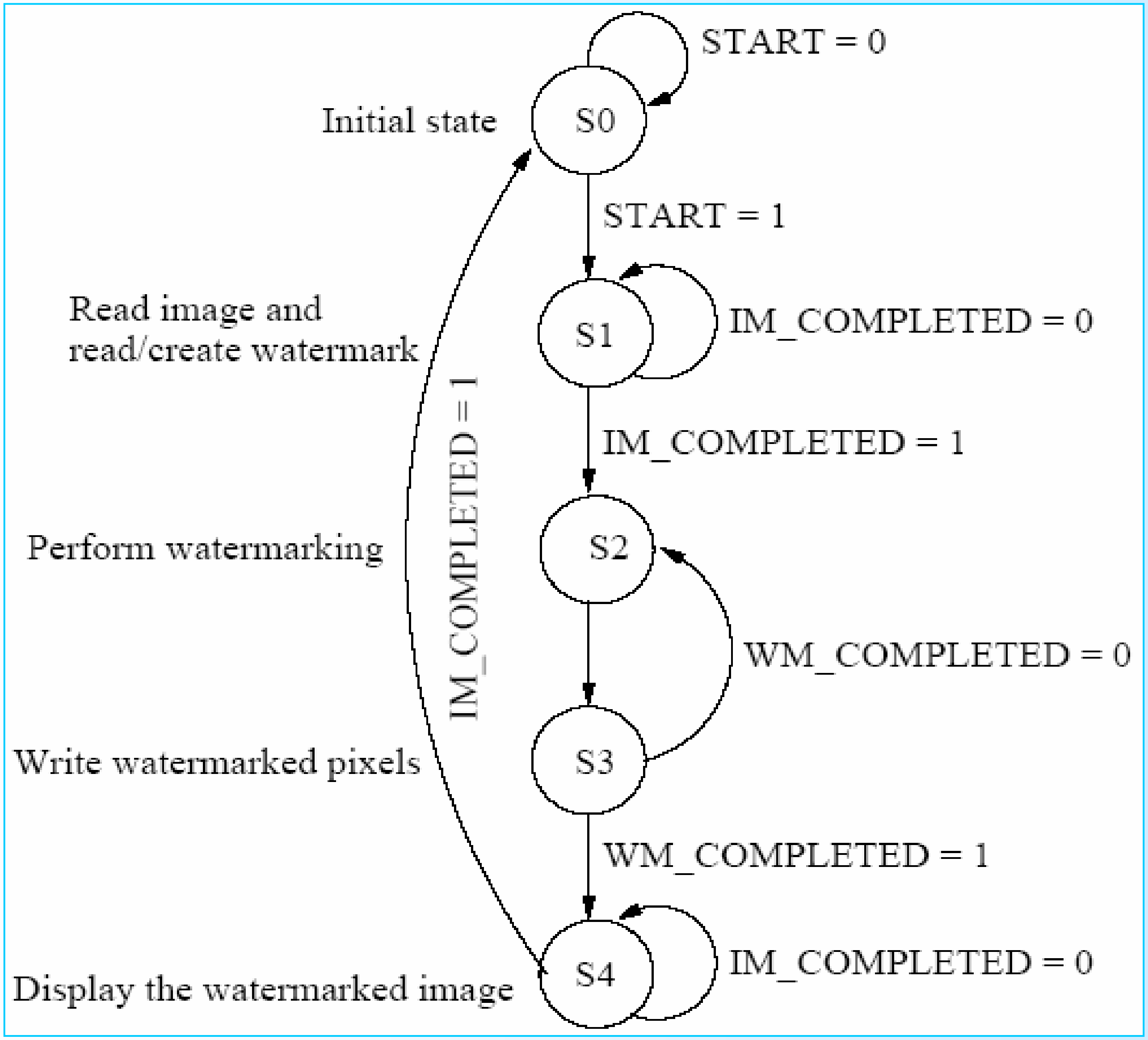
Watermark insertion is performed in the k-th image bit plane using the following function.

$$
\begin{aligned}
I_W[0 \rightarrow k-1](i,j) &= I[0 \rightarrow k-1](i,j) \\
I_W[k](i,j) &= I[k](i,j)\,\mathrm{XOR}\,W(i,j) \\
I_W[k+1 \rightarrow 7](i,j) &= I[k+1 \rightarrow 7](i,j)
\end{aligned}
$$

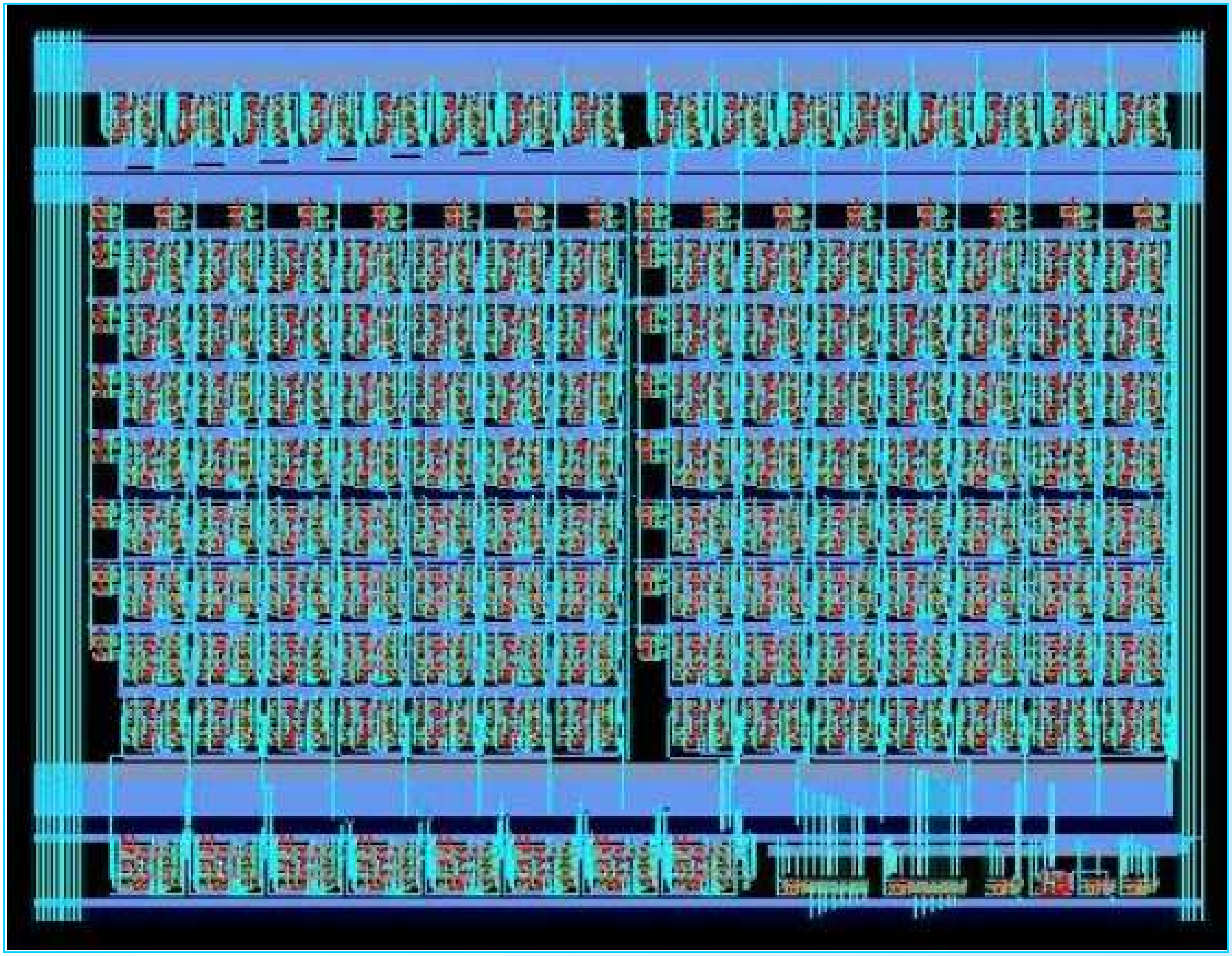
Spatial Invisible: Overall Datapath Architecture



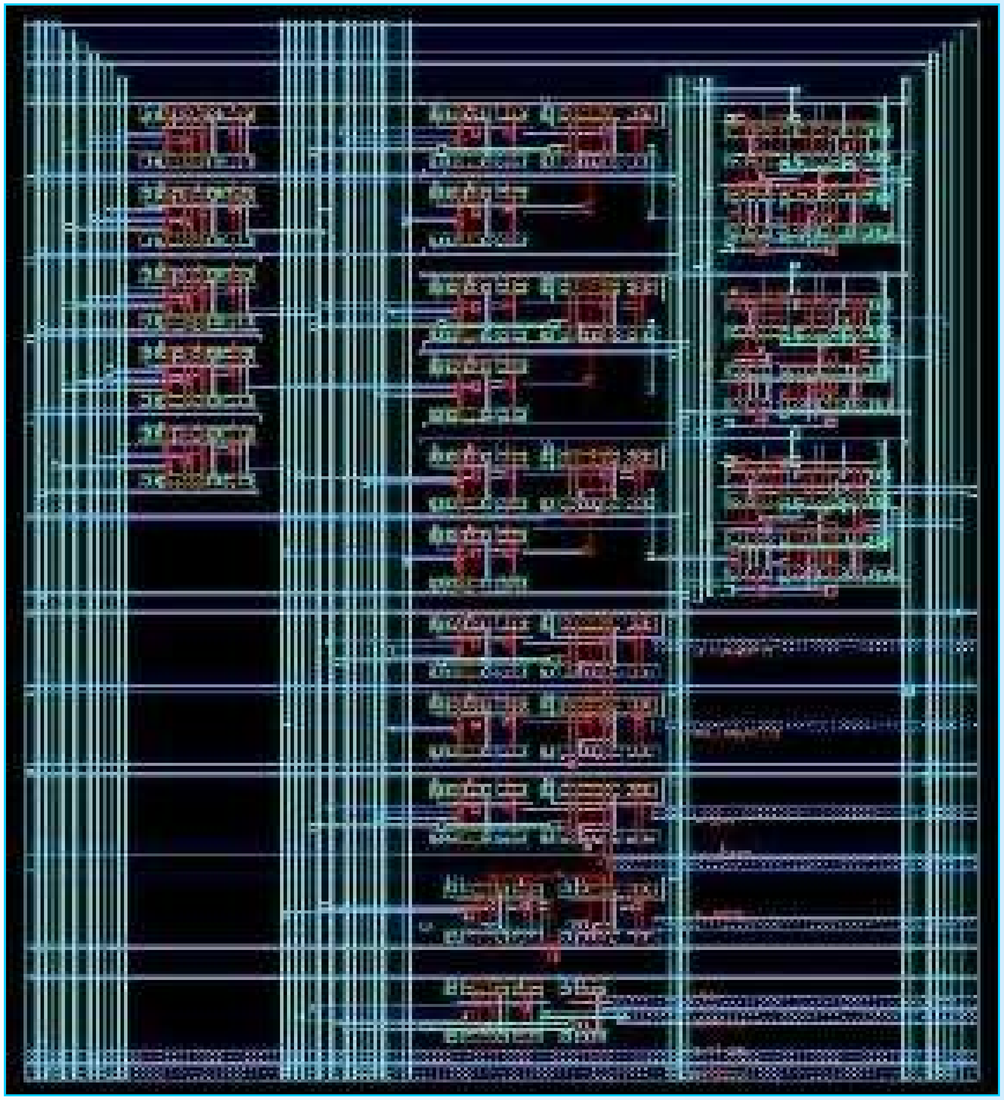
Spatial Invisible: Overall Controller



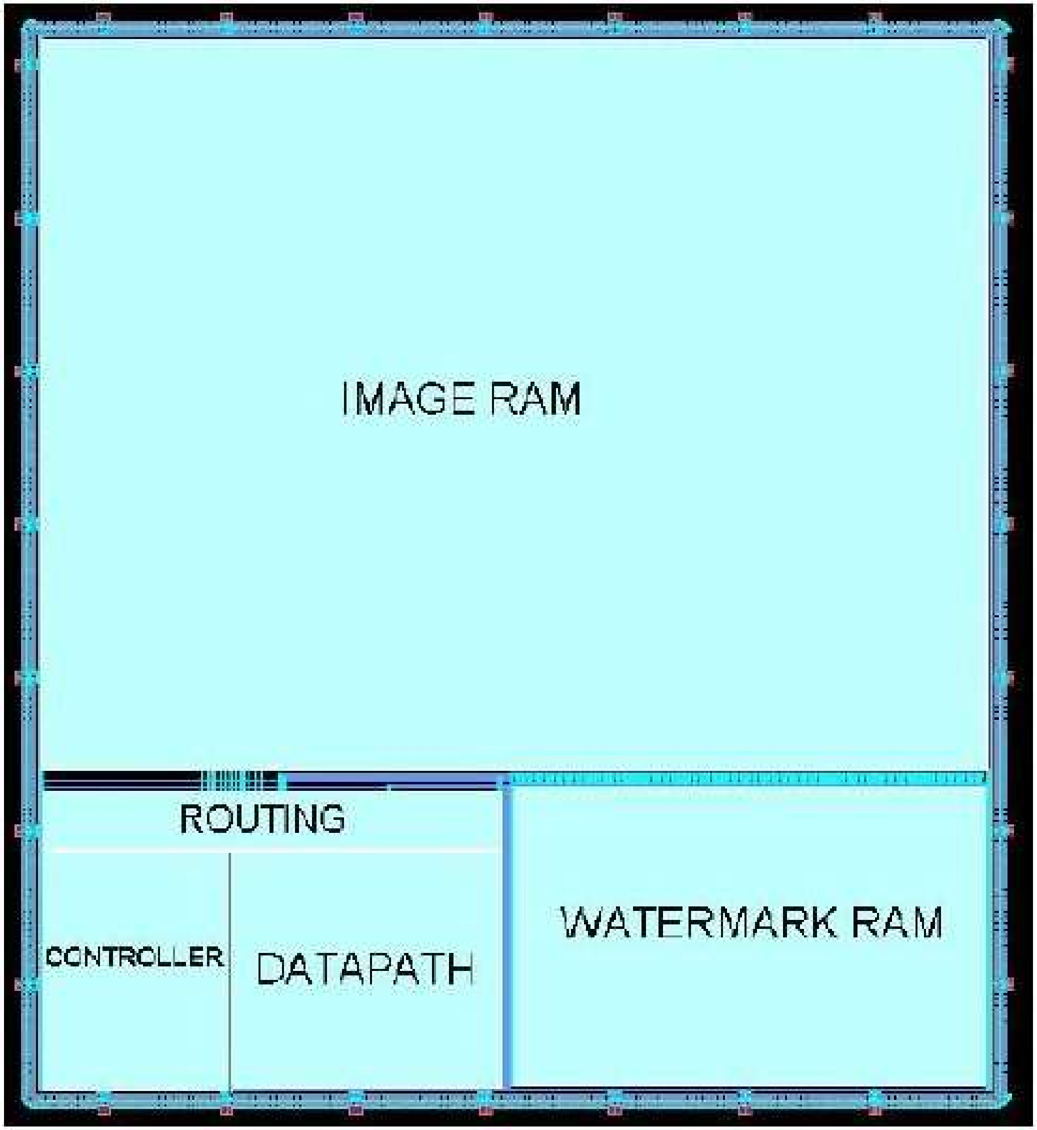
Spatial Invisible: Datapath Layout



Spatial Invisible: Controller Layout



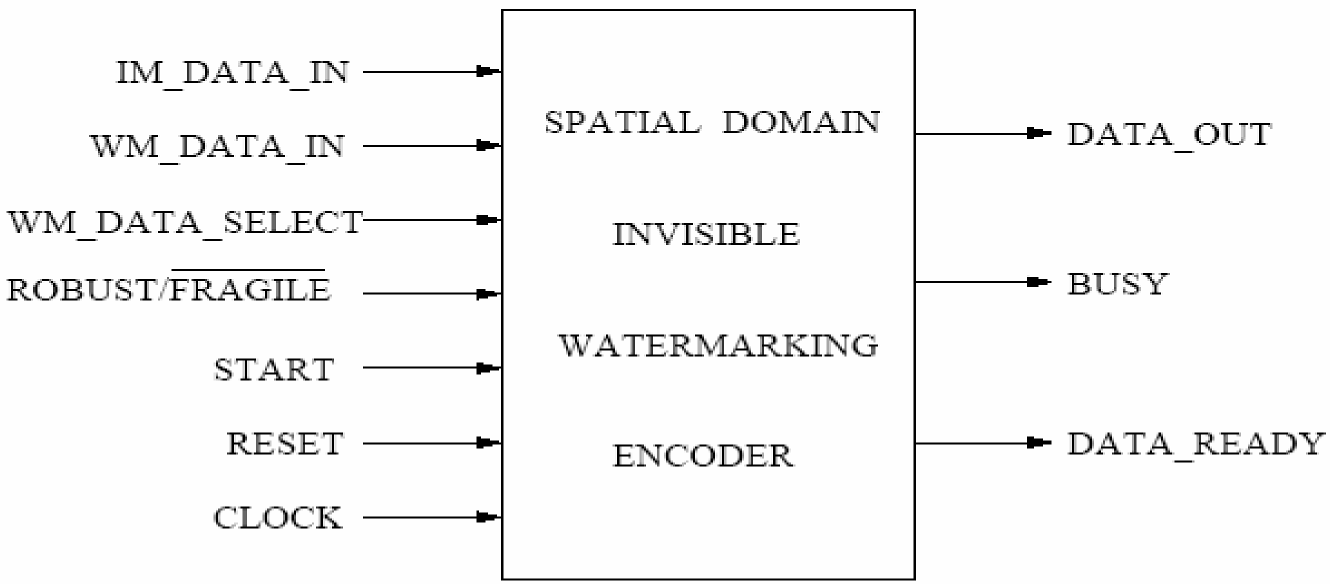
Spatial Invisible: Overall Chip



Spatial Invisible: Overall Chip …

Table 9.4. Overall Chip Statistics

Area (with RAM)	$15.012 \times 14.225 mm^2$
Number of gates (with RAM)	$1188K$
Number of gates (without RAM)	4820
Clock frequency (with RAM)	$151 MHz$
Clock frequency (without RAM)	$545 MHz$
Number of I/O pins	25
Power (with RAM)	$24 mW$
Power (without RAM)	$2.0547 mW$

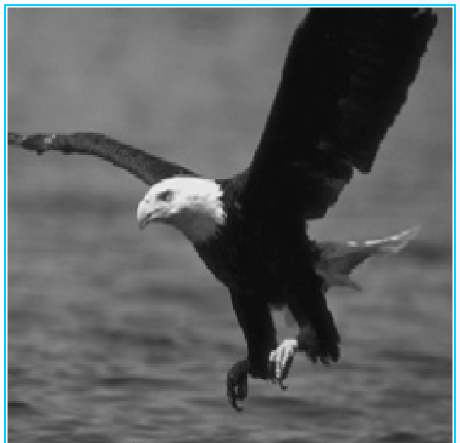


Spatial Invisible: Results

(a) Original Shuttle

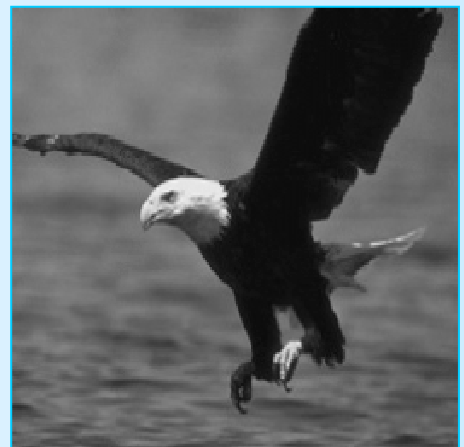
(b) Robust Watermarked

(c) Fragile Watermarked



(a) Original Bird

(b) Robust Watermarked



(c) Fragile Watermarked

Spatial Visible : Notations used in Algorithms

Table 9.5. List of Variables used in Algorithm Explanation

I	: Original (or host) image (a grayscale image)
W	: Watermark image (a grayscale image)
(m, n)	: A pixel location
I_W	: Watermarked image
$N_I \times N_I$	: Original image dimension
$N_W \times N_W$	: Watermark image dimension
i_k	: The k^{th} block of the original image I
w_k	: The k^{th} block of the watermark image W
$i_{W n}$	: The k^{th} block of the watermarked image I_W
α_k	: Scaling factor for k^{th} block (used for host image scaling)
β_k	: Embedding factor for k^{th} block (used for watermark image scaling)
μ_I	: Mean gray value of the original image I
$\mu_{I k}$	: Mean gray value of the original image block i_k
$\sigma_{I k}$	: Variance of the original image block i_k
α_{max}	: The maximum value of α_k
α_{min}	: The minimum value of α_k
β_{max}	: The maximum value of β_k
β_{min}	: The minimum value of β_k
I_{white}	: Gray value corresponding to pure white pixel
α_I	: A global scaling factor
C_1, C_2, C_3, C_4	: Linear regression co-efficients

Spatial Visible : Algorithm 1

- The original algorithm proposed by Braudaway, et. al.

$$I_W(m,n) = \begin{cases} I(m,n) + W(m,n)\left(\frac{I_{white}}{38.667}\right)\left(\frac{I(m,n)}{I_{white}}\right)^{\frac{2}{3}}\alpha_I & \text{for } \frac{I(m,n)}{I_{white}} > 0.008856 \\ I(m,n) + W(m,n)\left(\frac{I(m,n)}{903.3}\right)\alpha_I & \text{for } \frac{I(m,n)}{I_{white}} \le 0.008856 \end{cases}$$

- Assuming $I_{white} = 256$, simplified to:

$$I_W(m,n) = \begin{cases} I(m,n) + \left(\frac{\alpha_I}{6.0976}\right) W(m,n)\,(I(m,n))^{\frac{2}{3}} & \text{for } I(m,n) > 2.2583 \\ I(m,n) + \left(\frac{\alpha_I}{903.3}\right) W(m,n)\,I(m,n) & \text{for } I(m,n) \le 2.2583 \end{cases}$$

- Fitting piecewise linear model and regression co-efficients :

$$I_W(m,n) = \begin{cases} I(m,n) + \left(\frac{\alpha_I}{903.3}\right) W(m,n)\,I(m,n) & \text{for } I(m,n) \le 2 \\ I(m,n) + \left(\frac{\alpha_I C_1}{6.0976}\right) W(m,n)\,I(m,n) & \text{for } 2 < I(m,n) \le 64 \\ I(m,n) + \left(\frac{\alpha_I C_2}{6.0976}\right) W(m,n)\,I(m,n) & \text{for } 64 < I(m,n) \le 128 \\ I(m,n) + \left(\frac{\alpha_I C_3}{6.0976}\right) W(m,n)\,I(m,n) & \text{for } 128 < I(m,n) \le 192 \\ I(m,n) + \left(\frac{\alpha_I C_4}{6.0976}\right) W(m,n)\,I(m,n) & \text{for } 192 < I(m,n) < 256 \end{cases}$$

Spatial Visible : Algorithm 2

- Watermark insertion is carried out block-by-block using:

$$i_{W_k} = \alpha_k i_k + \beta_k w_k \qquad k = 1, 2...$$

- The scaling and embedding factors are found out as,

$$\alpha_k = \frac{1}{\hat{\sigma}_{I_k}} exp\left(-(\hat{\mu}_{I_k} - \hat{\mu}_I)^2\right)$$

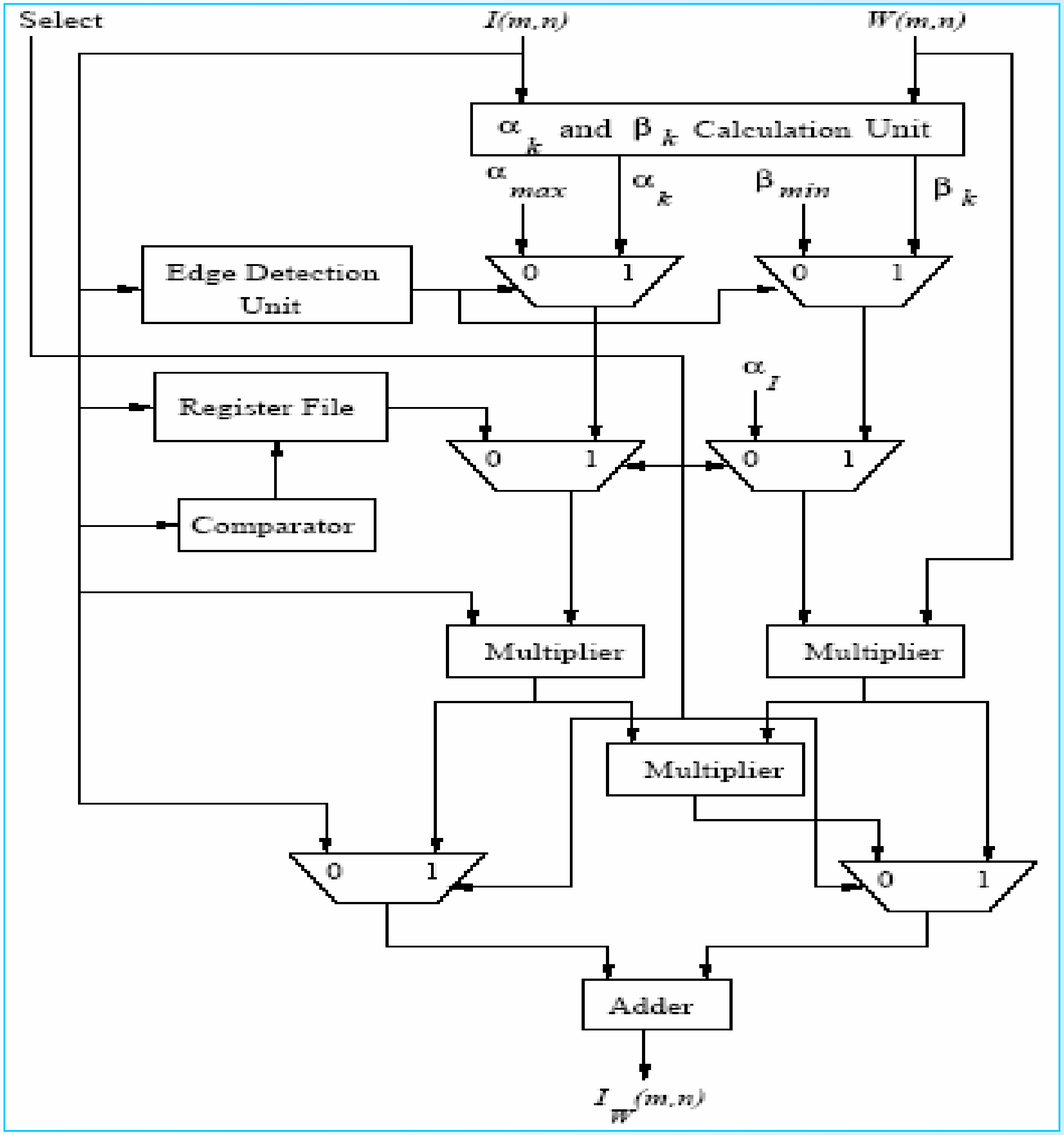
$$\beta_k = \hat{\sigma}_{I_k}\left(1 - exp\left(-(\hat{\mu}_{I_k} - \hat{\mu}_I)^2\right)\right)$$

- Values are scaled to proper range :

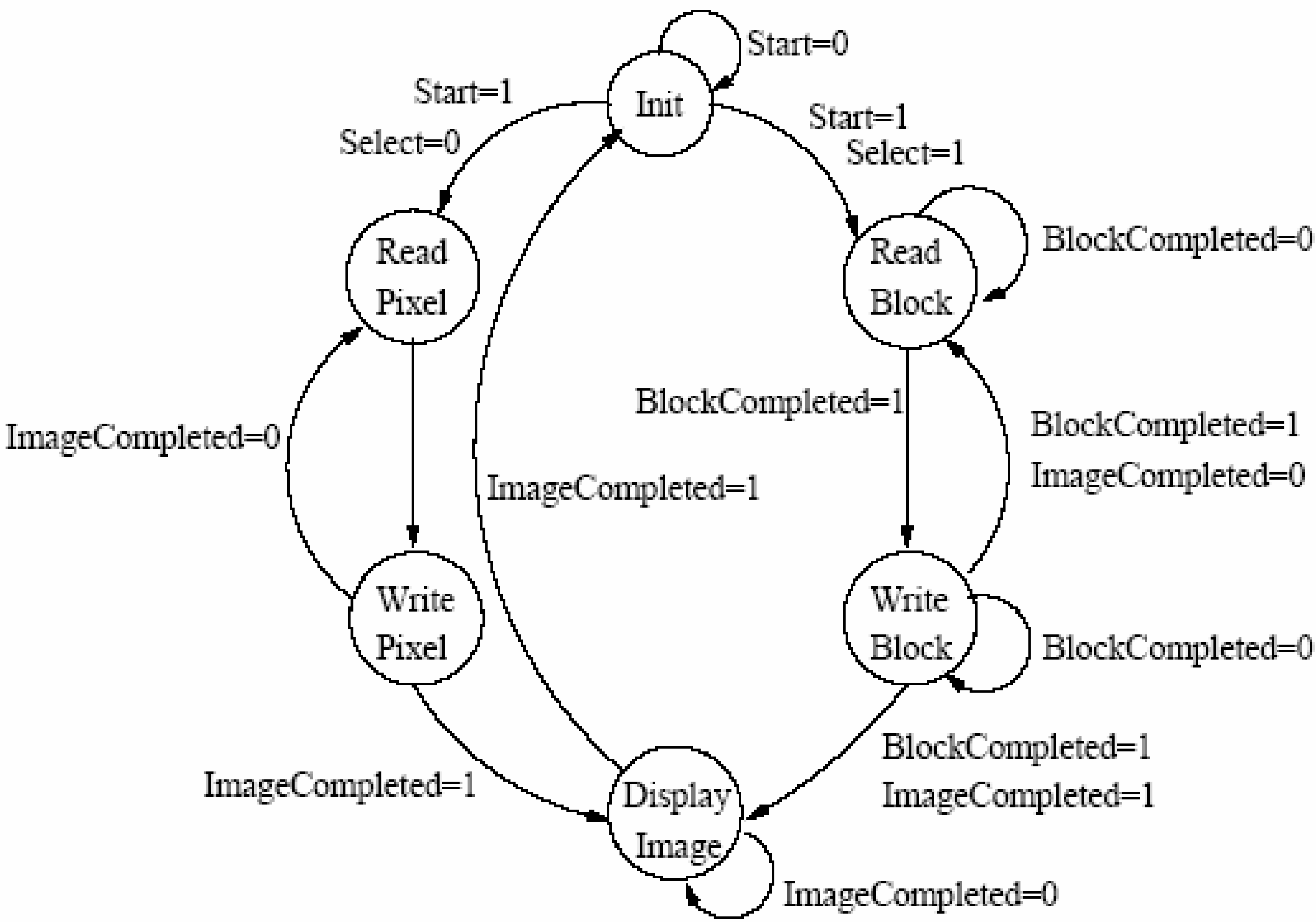
$$\alpha_k = \alpha_{min} + (\alpha_{max} - \alpha_{min})\frac{1}{\hat{\sigma}_{I_k}} exp\left(-(\hat{\mu}_{I_k} - \hat{\mu}_I)^2\right)$$

$$\beta_k = \beta_{min} + (\beta_{max} - \beta_{min})\hat{\sigma}_{I_k}\left(1 - exp\left(-(\hat{\mu}_{I_k} - \hat{\mu}_I)^2\right)\right)$$

Spatial Visible: Proposed Datapath Architecture

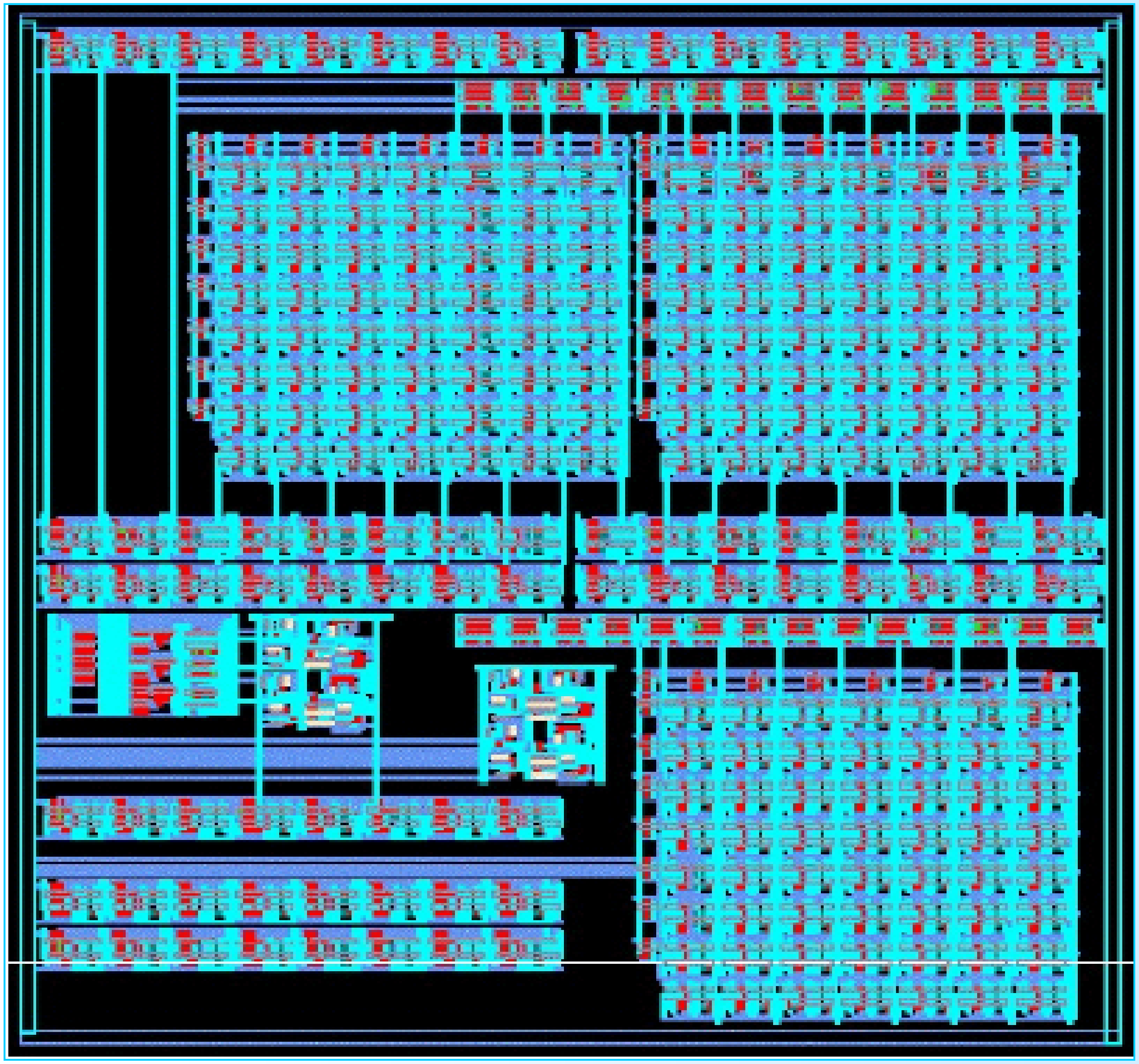


Spatial Visible: Proposed Controller



(b) Controller for the Merged Datapath

Spatial Visible: Overall Chip Layout



Spatial Visible: Overall Chip

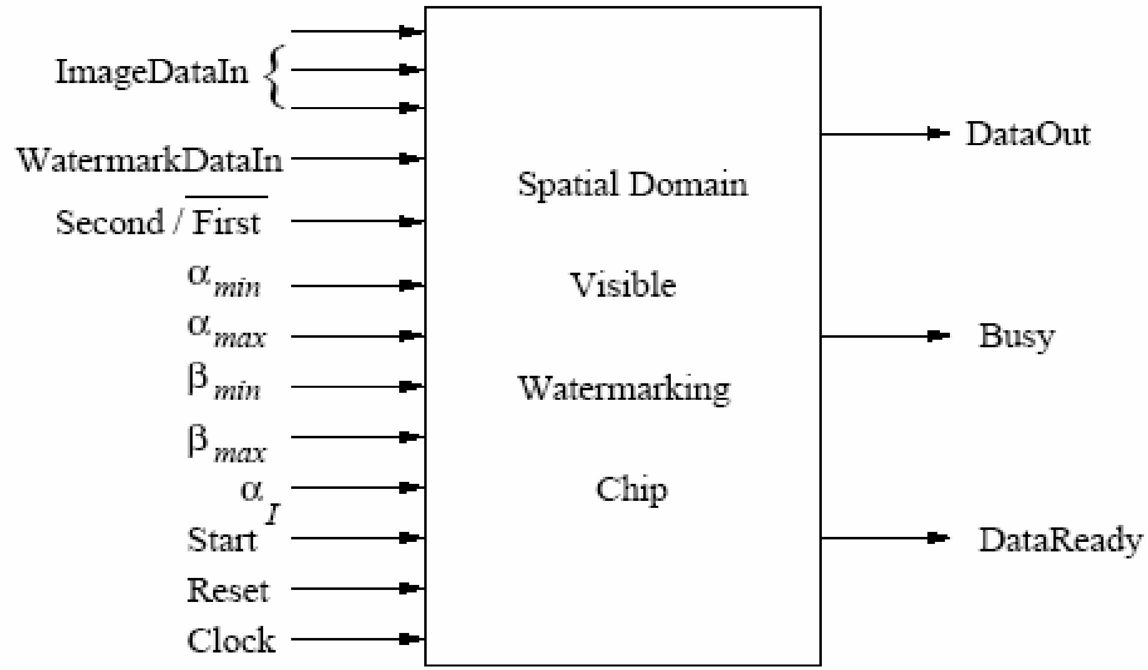


Figure 9.21. Pin Diagram for the Proposed Watermarking Chip

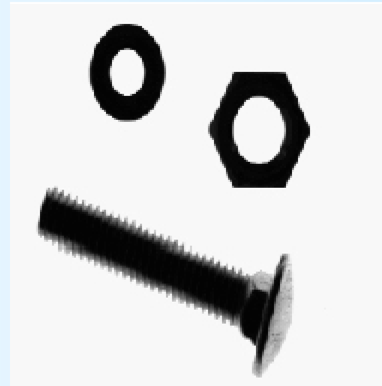
Table 9.7. Overall Statistics of the Watermarking Chip

Area	$3.34 \times 2.89 mm^2$
Number of gates	28469
Clock frequency	$292.27 MHz$
Number of I/O pins	72
Power	$6.9286 mW$

Spatial Visible: Results


(a) Lena


(b) Bird


(c) Nuts and Bolts

(d) Watermark

Original Images and Watermark


(a) Lena


(b) Bird


(c) Nuts and Bolts

Watermarked Images using Algorithm 1

NOTE: Similar watermarked images are obtained using algorithm2. The difference lies in the SNR.

DCT Domain : Algorithms

- The invisible watermark insertion involves addition of random numbers to relatively perceptual significant co-efficients of the host image.

$$c_{I_{W\,k}}(m,n) = c_{I_k}(m,n) + \alpha\, r_k(m,n)$$

- The visible watermark is inserted in the host images block-by-block and watermarked image block is obtained.

$$c_{I_{W\,k}} = \alpha_k\, c_{I\,k} + \beta_k\, c_{W\,k}$$

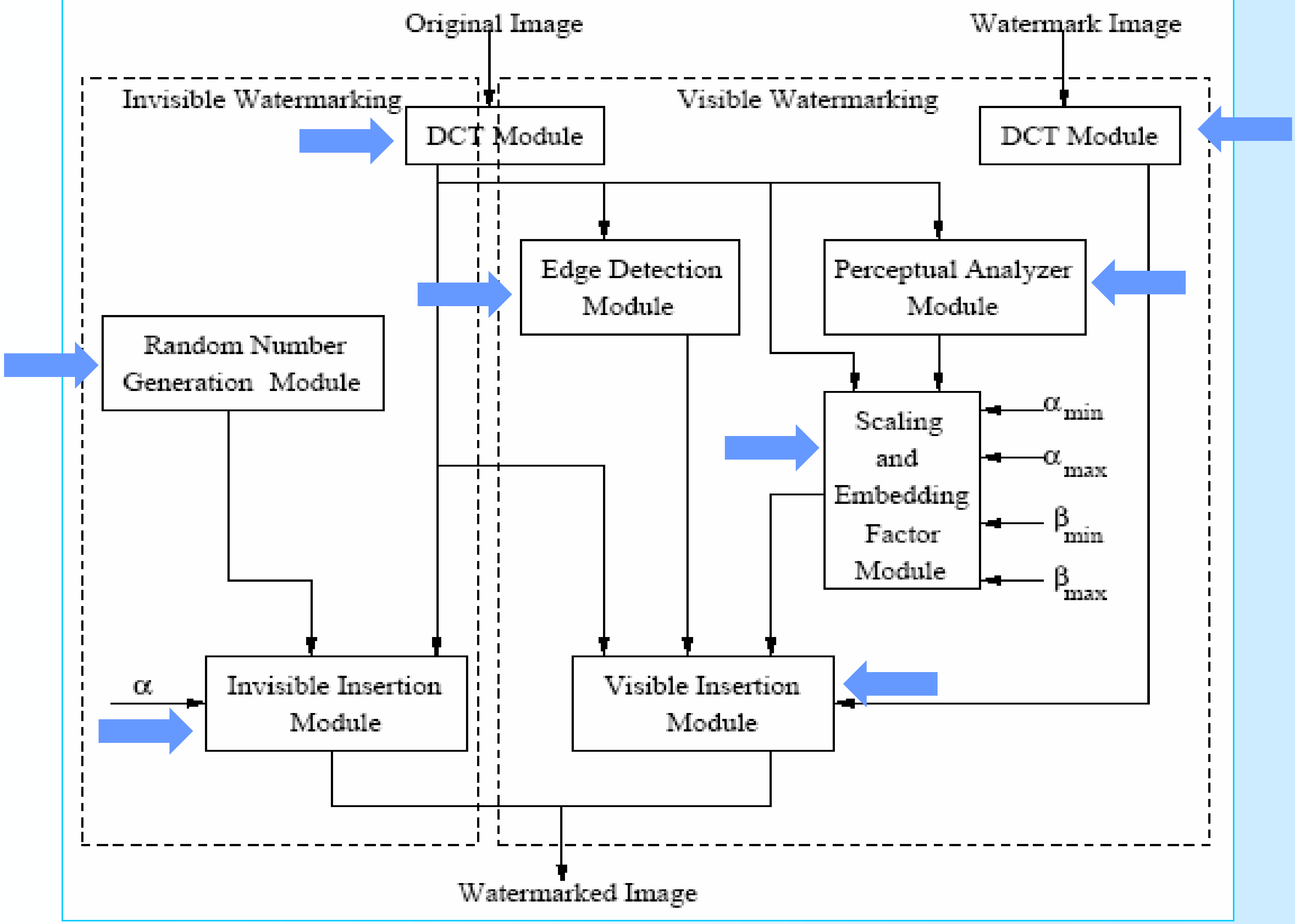
- Current scaling and embedding factors are obtained as,

$$\alpha_k^c = \sigma_{AC\,I_k}\, exp\left(-(\mu^*{}_{DC\,I_k} - \mu^*{}_{DC\,I})^2\right)$$

$$\beta_k^c = \frac{1}{\sigma_{AC\,I_k}}\left(1 - exp\left(-(\mu^*{}_{DC\,I_k} - \mu^*{}_{DC\,I})^2\right)\right)$$

- The current values are then linearly scaled to user defined ranges.

DCT Domain: Proposed Architecture



DCT Domain: Dual Voltage and Freq.

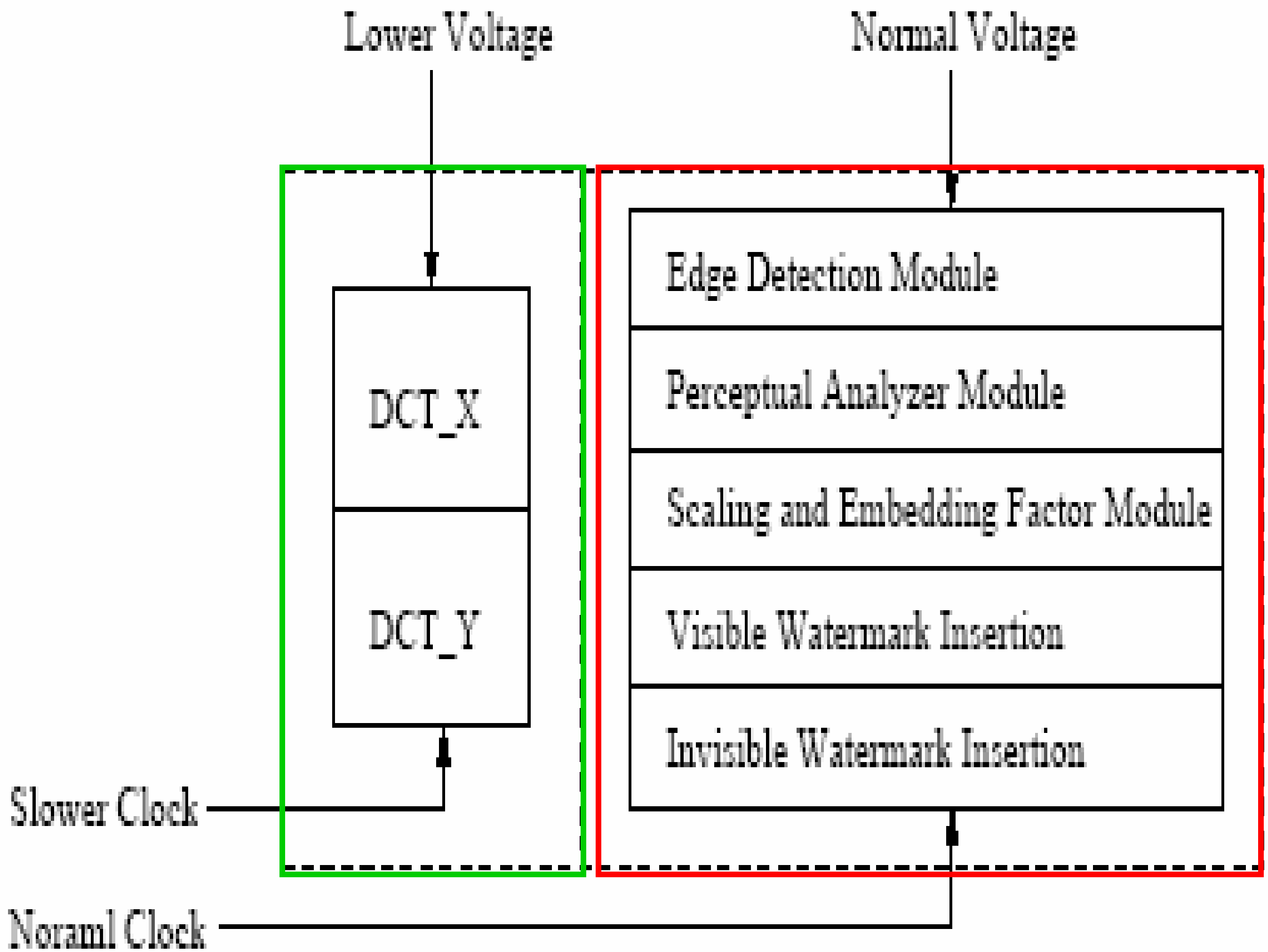
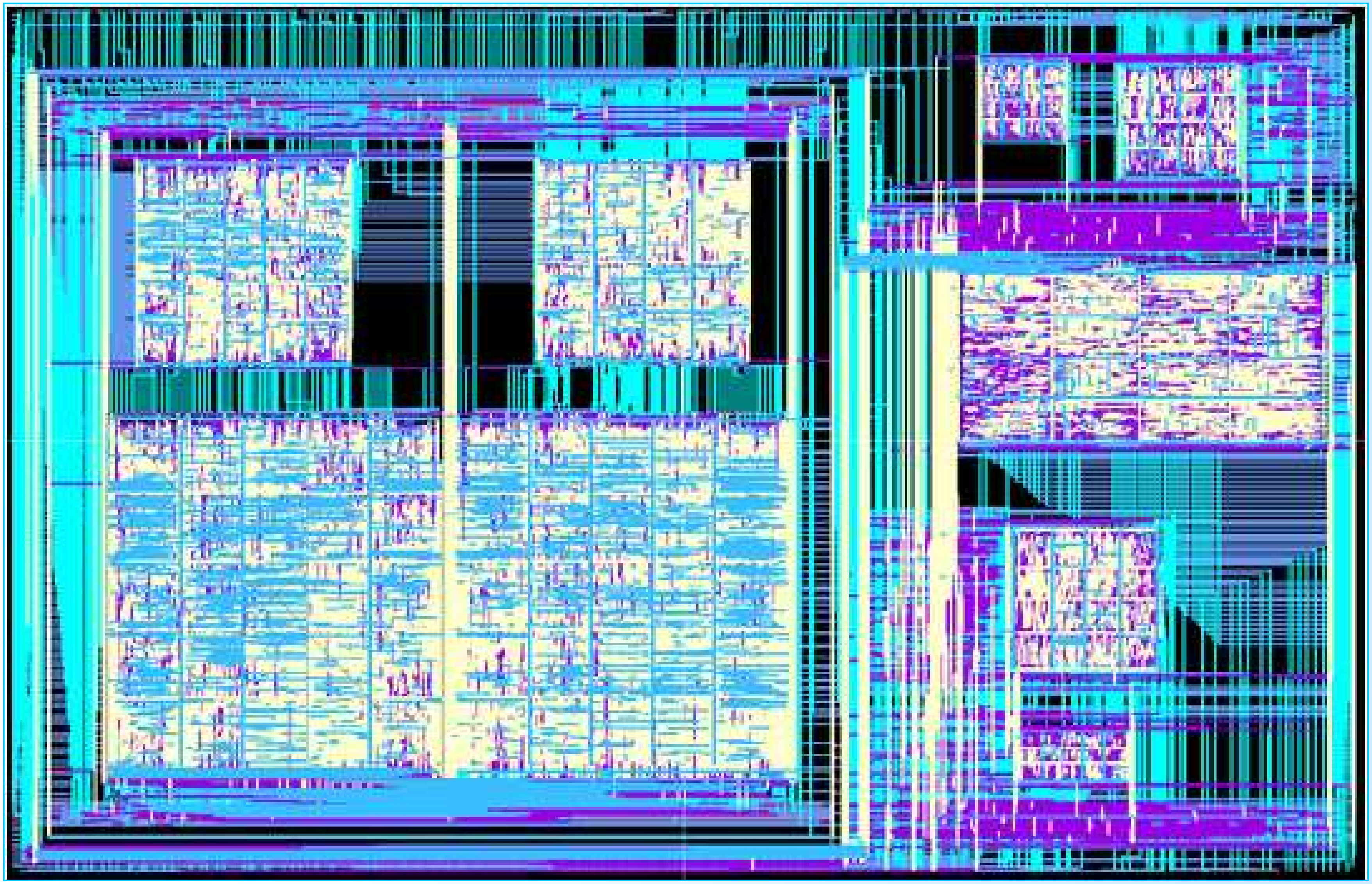


Figure 9.28. Dual Voltage and Dual Frequency Operation of the Datapath

DCT Domain: Overall Chip Layout (borrowed from masters thesis of Karthik)



Conclusions

- The reduction of peak power, peak power differential, average power and energy are equally important.
- The polynomial time-complexity resource and time constrained energy minimization scheduling algorithms could reduce energy consumption significantly with reasonable or no time penalty. ILP-based EDP minimization is an alternative to achieve same thing.
- The function CPF could capture all the different forms of power and its minimization using heuristic or ILP could yield significant reductions in all the different forms of power.
- The MPG function used as an alterative results comparable reductions, except energy reduction.
- The comparison of peak and average power minimization and only peak power minimization shows that there is 5% increase in peak power reduction.
- The MVDFC approach foundout to be better alternative. For the circuits having almost equal number of addition and multiplier operations in the critical path the savings are maximum with no time penalty for MVDFC case.
- The scheduling schemes are useful for data intensive applications.
- It is observed that the results of hardware based watermarking schemes are comparable to that of software.

Impact of this Dissertation

- None of the datapath scheduling algorithms available in current literature minimize transient power. There are few works available that handle peak power minimization. There are no research works handling both voltage and frequency parameters. Thus, we conclude any of the low power datapath scheduling algorithms proposed in this dissertation can create strong impact low power behavioral synthesis research.

- All the watermarking chip designed are the first implementations in the respective category. At this digital age, when the copyright and piracy are threat to industrial growths, the secure digital devices integrated with watermarking chips can produce copyrighted multimedia data in real-time.

Future Works

- The applicability of the scheduling schemes for pipelining is to be investigated.
- The effect of switching activity is to be taken into account.
- The detail design of controller is to be done.
- The effect on clocking network is to be studied.
- Different nonlinear optimization techniques and new linear techniques can be investigated to minimize CPF / MPG.
- Similarly, the design works can be extended to develop pipelined and / or SIMD based designs.
- Implementation of video and audio watermarking algorithms can also be considered.

Publications from this Dissertation

1. S. P. Mohanty, N. Ranganathan and R. K. Namballa, "VLSI Implementation of Visible Watermarking for a Secure Digital Still Camera Design", to appear in Proc. of the 17th IEEE Intl. Conf. on VLSI Design, 2004.
2. S. P. Mohanty, N. Ranganathan and S. K. Chappidi, "ILP Models for Energy and Transient Power Minimization During Behavioral Synthesis", to appear in Proc. of the 17th IEEE Intl. Conf. on VLSI Design, 2004.
3. S. P. Mohanty, N. Ranganathan and S. K. Chappidi, "Power Fluctuation Minimization During Behavioral Synthesis using ILP-Based Datapath Scheduling", to appear in Proc. ICCD 2003.
4. S. P. Mohanty, N. Ranganathan and S. K. Chappidi, "Transient Power Minimization Through Datapath Scheduling in Multiple Supply Voltage Environment", to appear in Proc. of the 10th IEEE International Conference on Electronics, Circuits and Systems, 2003.
5. S. P. Mohanty, N. Ranganathan and R. K. Namballa, "VLSI Implementation of Invisible Digital Watermarking Algorithms Towards the Development of a Secure JPEG Encoder", in Proc. of the IEEE Workshop on Signal Processing Systems, pp. 183-188, 2003.
6. S. P. Mohanty, N. Ranganathan and S. K. Chappidi, "Simultaneous Peak and Average Power Minimization during Datapath Scheduling for DSP Processors", in Proc. of GLSVLSI pp. 215-220, 2003.
7. S. P. Mohanty, N. Ranganathan and S. K. Chappidi, "An ILP-Based Scheduling Scheme for Energy Efficient High Performance Datapath Synthesis", in Proc. of ISCAS, Vol. 5, pp. 313-316, 2003.
8. S. P. Mohanty, N. Ranganathan and S. K. Chappidi, "Peak Power Minimization Through Datapath Scheduling", in Proc. of ISVLSI, pp. 121-126, 2003.
9. S. P. Mohanty and N. Ranganathan, "A Framework for Energy and Transient Power Reduction during Behavioral Synthesis", in Proc. of the 16th IEEE Intl. Conf. on VLSI Design 2003, pp. 539-545, 2003, (Nominated for best paper award; ranked within top 5 papers out of 210 submissions).
10. S. P. Mohanty and N. Ranganathan, "Energy Efficient Scheduling for Datapath Synthesis", in Proc. of the 16th IEEE Intel. Conf. on VLSI Design, pp. 446-451, 2003.
11. S. P. Mohanty, N. Ranganathan and V. Krishna, "Datapath Scheduling using Dynamic Frequency Clocking", in Proc. of ISVLSI, pp. 65-70, 2002.

Thank you