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
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

\[ P_{\text{dynamic}} = \frac{1}{2} C_L V_{dd}^2 N f \]

- \( C_L \) = load capacitor, \( V_{dd} \) = supply voltage,
- \( N \) = average number of transitions/clock cycle
  = \( E(\text{sw}) = 2 a_{0->1} \) = switching activity
- \( f \) = clock frequency

Note:
1. \( N \times f \) is transition density
2. \( C_L \times N \) (\( = C_{\text{sw}} = C_{\text{eff}} \)) 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

More details:
- Ranganathan, et.al.
- Byrnjofson 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

<table>
<thead>
<tr>
<th>Proposed Scheme</th>
<th>Optimization Method Used</th>
<th>Constraints Assumed</th>
<th>Operating Voltage Levels</th>
<th>Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnson and Roy [89, 90]</td>
<td>ILP</td>
<td>Time</td>
<td>(5.0V → 2.0V)</td>
<td>Exponential</td>
</tr>
<tr>
<td>Johnson and Roy [6]</td>
<td>ILP</td>
<td>Time</td>
<td>(5.0V, 3.3V, 2.4V)</td>
<td>Exponential</td>
</tr>
<tr>
<td>Chang and Pedram [63, 91]</td>
<td>Dynamic Programming</td>
<td>Time</td>
<td>(5.0V, 3.3V, 2.4V)</td>
<td>Pseudo-Polynomial</td>
</tr>
<tr>
<td>Lin. Hwang and Wu [92]</td>
<td>ILP and Heuristic</td>
<td>Time and Resource</td>
<td>(5.0V, 3.3V)</td>
<td>Exponential</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$O(n^3\log n)$</td>
</tr>
<tr>
<td>Sarrafzadeh and Raje [93]</td>
<td>Dynamic Prog Geometric</td>
<td>Time and Resource</td>
<td>(5.0V, 3.3V)</td>
<td>$O\left(\frac{n^2k\beta</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$O(nC\log nC)$</td>
</tr>
<tr>
<td>Kumar and Bayoumi [94, 95, 96]</td>
<td>Stochastic Evolution</td>
<td>Resource</td>
<td>(5.0V, 3.3V, 2.4V)</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Elgamel and Bayoumi [97]</td>
<td>Genetic Algorithms</td>
<td>Time and Area</td>
<td>(5.0V, 3.3V, 2.4V)</td>
<td>NA</td>
</tr>
<tr>
<td>Shiu and Chakrabarti [98, 99]</td>
<td>List-Based</td>
<td>Time and Resource</td>
<td>(5.0V, 3.3V) or (5.0V, 3.3V, 2.4V)</td>
<td>Polynomial</td>
</tr>
<tr>
<td>Manzak and Chakrabarti [100]</td>
<td>Lagrangian Multiplier</td>
<td>Time and Resource</td>
<td>(5.0V, 3.3V, 2.4V)</td>
<td>$O(n^2)$ and $O(n^2\log L)$</td>
</tr>
<tr>
<td>Manzak and Chakrabarti [101]</td>
<td>List-Based</td>
<td>Time and Resource</td>
<td>(5.0V, 3.3V, 2.4V)</td>
<td>$O(r^2L^2)$</td>
</tr>
</tbody>
</table>

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

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

- 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

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

- Do not handle MV or DFC
- High time penalty
- Do not minimize other forms of power
## Scheduling for Variable Frequency Processor

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

### Key Points
- Handle variable frequency at OS or compiler level.
- Minimize average power or energy only.
# Design and Synthesis using Variable Frequency

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

- Low-power or High-performance synthesis or design works using variable frequency.
- Minimize only average power.
## Table 2.6. Watermarking Chips Proposed in Current Literature

<table>
<thead>
<tr>
<th>Proposed Work</th>
<th>Type of Watermark</th>
<th>Target Object</th>
<th>Working Domain</th>
<th>Technology</th>
<th>Chip Area</th>
<th>Chip Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathai and et. al. [161]</td>
<td>Invisible Robust</td>
<td>Video</td>
<td>Wavelet</td>
<td>0.18μ</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Tsai and Lu [162]</td>
<td>Invisible Robust</td>
<td>Image</td>
<td>DCT</td>
<td>0.35μ</td>
<td>3.064 x 3.064 mm²</td>
<td>62.78 mW, 3.3V, 50 MHz</td>
</tr>
<tr>
<td>Garimella and et. al. [163]</td>
<td>Invisible Fragile</td>
<td>Image</td>
<td>Spatial</td>
<td>0.13μ</td>
<td>3453 x 3453 μm²</td>
<td>37.6 μW, 1.2V</td>
</tr>
</tbody>
</table>

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
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 (\(c_{fi_c}\)).

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>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>total number of control steps in the DFG</td>
</tr>
<tr>
<td>( O )</td>
<td>total number of operations in the DFG</td>
</tr>
<tr>
<td>( c )</td>
<td>a control step or a clock cycle in the DFG</td>
</tr>
<tr>
<td>( o_i )</td>
<td>any operation ( i ), where ( 1 \leq i \leq O ),</td>
</tr>
<tr>
<td>( P_c )</td>
<td>the total power consumption of all functional units active in control step ( c ) (cycle power consumption)</td>
</tr>
<tr>
<td>( P_{peak} )</td>
<td>peak power consumption for the DFG equal to ( \max(P_c) \forall c )</td>
</tr>
<tr>
<td>( P )</td>
<td>mean power consumption of the DFG (average ( P_c ) over all control steps)</td>
</tr>
<tr>
<td>( P_{norm} )</td>
<td>normalised mean power consumption of the DFG</td>
</tr>
<tr>
<td>( D P_c )</td>
<td>cycle difference power (for cycle ( c ); a measure of cycle power fluctuation)</td>
</tr>
<tr>
<td>( D P_{peak} )</td>
<td>peak differential power consumption for the DFG equal to ( \max(DP_c) \forall c )</td>
</tr>
<tr>
<td>( D P )</td>
<td>mean of the cycle difference powers for all control steps in DFG</td>
</tr>
<tr>
<td>( D P_{norm} )</td>
<td>normalised mean of the mean difference powers for all steps in DFG</td>
</tr>
<tr>
<td>( CPF )</td>
<td>cycle power function</td>
</tr>
<tr>
<td>( FU_{k,v} )</td>
<td>any functional unit of type ( k ) operating at voltage level ( v )</td>
</tr>
<tr>
<td>( FU_i )</td>
<td>any functional unit ( FU_{k,v} ) needed by ( o_i ) for its execution (( o_i \in FU_{k,v} ))</td>
</tr>
<tr>
<td>( FU_{i,c} )</td>
<td>any functional unit ( FU_i ) active in control step ( c )</td>
</tr>
<tr>
<td>( R_c )</td>
<td>total number of functional units active in step ( c )</td>
</tr>
<tr>
<td>( \alpha_{i,c} )</td>
<td>switching activity of resource ( FU_{i,c} )</td>
</tr>
<tr>
<td>( V_{i,c} )</td>
<td>operating voltage of resource ( FU_{i,c} )</td>
</tr>
<tr>
<td>( C_{i,c} )</td>
<td>load capacitance of resource ( FU_{i,c} )</td>
</tr>
<tr>
<td>( f_c )</td>
<td>frequency of control step ( c )</td>
</tr>
</tbody>
</table>
The power consumption for any control step $c$ is given by,

$$P_c = \sum_{i=1 \rightarrow Rc} a_{i,c} C_{i,c} V_{i,c}^2 f_c$$

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

$$P_{\text{peak}} = \max \left( P_c \right)_{c=1 \rightarrow N} = \max \left( \sum_{i=1 \rightarrow Rc} a_{i,c} C_{i,c} V_{i,c}^2 f_c \right)_{c=1 \rightarrow N}$$

Average power is characterized as mean cycle power ($P_c$):

$$P = \frac{1}{N} \left( \sum_{c=1 \rightarrow N} P_c \right) = \frac{1}{N} \left( \sum_{c=1 \rightarrow N} \sum_{i=1 \rightarrow Rc} a_{i,c} C_{i,c} V_{i,c}^2 f_c \right)$$

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, \( \mu \)) is \( m = \frac{1}{n} \sum x_i \).

- The **absolute deviation** of these observations is defined as \( \Delta x_i = |x_i - m| \).

- The **mean deviation** of the observations is given by \( MD = \frac{1}{n} \sum |x_i - m| \).

- We model the cycle difference power \( DP_c \) as the absolute deviation of cycle power \( P_c \) 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 ($P_{\text{norm}}$) is defined as:
  $$= \frac{\text{mean cycle power consumption over all control steps}}{\text{maximum power consumption in any control step}}$$
  $$= \frac{\text{Mean } (P_c)}{\text{Maximum } (P_c)}$$
  $$= \frac{P}{P_{\text{peak}}}$$

- Normalized mean cycle difference power ($DP_{\text{norm}}$) is defined as:
  $$= \frac{\text{mean cycle difference power over all control steps}}{\text{maximum cycle difference power for any control step}}$$
  $$= \frac{\text{Mean } (DP_c)}{\text{Maximum } (DP_c)}$$
  $$= \frac{DP}{DP_{\text{peak}}}$$
CPF Minimization: Power Models ...

- Cycle power function is defined as:

\[ \text{CPF} = P_{\text{norm}} + DP_{\text{norm}} \quad (1) \]

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

\[
CPF = \frac{P}{P_{\text{peak}}} + \frac{DP}{DP_{\text{peak}}} = \frac{1}{N} \sum_{c=1}^{N} P_c + \frac{1}{N} \sum_{c=1}^{N} |P - P_c| \quad (2)
\]

- Using the switching capacitance, voltage and frequency,

\[
CPF = \frac{1}{N} \sum_{c=1}^{N} \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c \max \left( \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c \right) \psi_c + \frac{1}{N} \sum_{c=1}^{N} \left( \frac{1}{N} \sum_{c=1}^{N} \left( \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c \right) - \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c \right) \psi_c \]

Dept. of CSE
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).

<table>
<thead>
<tr>
<th>Case</th>
<th>Multipliers</th>
<th>ALUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 at 2.4V</td>
<td>1 at 3.3V</td>
</tr>
<tr>
<td>2</td>
<td>2 at 2.4V</td>
<td>1 at 3.3V</td>
</tr>
<tr>
<td>3</td>
<td>2 at 2.4V</td>
<td>1 at 2.4V and 1 at 3.3V</td>
</tr>
<tr>
<td>4</td>
<td>1 at 2.4V and 1 at 3.3V</td>
<td>1 at 2.4V and 1 at 3.3V</td>
</tr>
</tbody>
</table>
### CPF Minimization: Experimental Results

(Notations used)

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_S$</td>
<td>total energy consumption assuming single frequency and single supply voltage</td>
</tr>
<tr>
<td>$E_D$</td>
<td>total energy consumption for dynamic clocking and multiple supply voltage</td>
</tr>
<tr>
<td>$P_{pS}$</td>
<td>peak power consumption for single frequency and single supply voltage</td>
</tr>
<tr>
<td>$P_{pD}$</td>
<td>peak power consumption for dynamic clocking and multiple supply voltage</td>
</tr>
<tr>
<td>$P_{mS}$</td>
<td>minimum power consumption for single frequency and single supply voltage</td>
</tr>
<tr>
<td>$P_{mD}$</td>
<td>minimum power consumption for dynamic clocking and multiple supply voltage</td>
</tr>
<tr>
<td>$T_S$</td>
<td>execution time assuming single frequency</td>
</tr>
<tr>
<td>$T_D$</td>
<td>execution time assuming dynamic frequency</td>
</tr>
<tr>
<td>$\Delta E$</td>
<td>total energy reduction $= \frac{E_S - E_D}{E_S}$</td>
</tr>
<tr>
<td>$\Delta P$</td>
<td>average power reduction $= \frac{(E_S/T_S) - (E_D/T_D)}{(E_S/T_S)}$</td>
</tr>
<tr>
<td>$\Delta P_p$</td>
<td>peak power reduction $= \frac{P_{pS} - P_{pD}}{P_{pS}}$</td>
</tr>
<tr>
<td>$\Delta DP$</td>
<td>differential power reduction $= \frac{(P_{pS} - P_{mS}) - (P_{pD} - P_{mD})}{(P_{pS} - P_{mS})}$</td>
</tr>
<tr>
<td>$R_T$</td>
<td>time ratio $= \frac{T_D}{T_S}$</td>
</tr>
</tbody>
</table>
## CPF Minimization: Experimental Results

<table>
<thead>
<tr>
<th>K</th>
<th>R</th>
<th>C</th>
<th>$P_{PS}$ (mW)</th>
<th>$P_{PD}$ (mW)</th>
<th>$\Delta P_p$ (%)</th>
<th>$P_{mS}$ (mW)</th>
<th>$P_{mD}$ (mW)</th>
<th>$\Delta DP$ (%)</th>
<th>$\Delta P$ (%)</th>
<th>$\Delta E$ (%)</th>
<th>N</th>
<th>$r_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>2</td>
<td>9.30</td>
<td>2.83</td>
<td>69.60</td>
<td>0.26</td>
<td>0.52</td>
<td>74.50</td>
<td>71.40</td>
<td>47.57</td>
<td>18</td>
<td>1.6</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>3</td>
<td>18.33</td>
<td>4.77</td>
<td>73.96</td>
<td>0.26</td>
<td>0.52</td>
<td>76.47</td>
<td>68.30</td>
<td>47.57</td>
<td>13</td>
<td>1.4</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>4</td>
<td>18.59</td>
<td>7.26</td>
<td>60.96</td>
<td>0.26</td>
<td>0.52</td>
<td>63.25</td>
<td>59.10</td>
<td>29.49</td>
<td>11</td>
<td>1.5</td>
</tr>
<tr>
<td>1</td>
<td>B</td>
<td>2</td>
<td>9.30</td>
<td>2.45</td>
<td>73.62</td>
<td>0.26</td>
<td>0.52</td>
<td>78.64</td>
<td>65.80</td>
<td>46.69</td>
<td>17</td>
<td>1.3</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>3</td>
<td>18.33</td>
<td>4.20</td>
<td>77.10</td>
<td>0.26</td>
<td>1.67</td>
<td>86.03</td>
<td>58.81</td>
<td>46.69</td>
<td>17</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>4</td>
<td>18.59</td>
<td>4.84</td>
<td>73.96</td>
<td>0.52</td>
<td>0.97</td>
<td>78.59</td>
<td>71.09</td>
<td>48.61</td>
<td>9</td>
<td>1.4</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
<td>2</td>
<td>9.30</td>
<td>2.83</td>
<td>69.60</td>
<td>0.26</td>
<td>0.52</td>
<td>74.50</td>
<td>50.90</td>
<td>42.44</td>
<td>29</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>3</td>
<td>18.30</td>
<td>4.84</td>
<td>73.96</td>
<td>0.26</td>
<td>0.40</td>
<td>75.75</td>
<td>67.70</td>
<td>42.93</td>
<td>15</td>
<td>1.4</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>4</td>
<td>18.59</td>
<td>7.61</td>
<td>59.05</td>
<td>0.26</td>
<td>0.40</td>
<td>60.63</td>
<td>65.19</td>
<td>38.49</td>
<td>15</td>
<td>1.4</td>
</tr>
<tr>
<td>1</td>
<td>D</td>
<td>2</td>
<td>9.30</td>
<td>2.45</td>
<td>73.62</td>
<td>0.26</td>
<td>0.52</td>
<td>78.64</td>
<td>41.17</td>
<td>44.43</td>
<td>27</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>D</td>
<td>3</td>
<td>18.07</td>
<td>4.07</td>
<td>77.49</td>
<td>0.26</td>
<td>0.52</td>
<td>80.09</td>
<td>37.49</td>
<td>44.43</td>
<td>27</td>
<td>0.9</td>
</tr>
<tr>
<td>3</td>
<td>D</td>
<td>4</td>
<td>18.07</td>
<td>6.55</td>
<td>63.75</td>
<td>0.26</td>
<td>0.40</td>
<td>65.49</td>
<td>53.10</td>
<td>38.45</td>
<td>16</td>
<td>1.2</td>
</tr>
<tr>
<td>1</td>
<td>E</td>
<td>2</td>
<td>9.30</td>
<td>2.74</td>
<td>70.52</td>
<td>0.26</td>
<td>0.52</td>
<td>75.45</td>
<td>58.54</td>
<td>46.11</td>
<td>15</td>
<td>1.3</td>
</tr>
<tr>
<td>2</td>
<td>E</td>
<td>3</td>
<td>18.07</td>
<td>4.07</td>
<td>77.49</td>
<td>0.26</td>
<td>0.52</td>
<td>75.45</td>
<td>58.54</td>
<td>46.11</td>
<td>15</td>
<td>1.3</td>
</tr>
<tr>
<td>3</td>
<td>E</td>
<td>4</td>
<td>18.59</td>
<td>7.04</td>
<td>62.15</td>
<td>0.24</td>
<td>0.40</td>
<td>63.77</td>
<td>40.69</td>
<td>27.21</td>
<td>11</td>
<td>1.2</td>
</tr>
</tbody>
</table>

**Average values**
- $P_{PS}$: 70.52 mW
- $P_{PD}$: 75.04 mW
- $\Delta P_p$: 59.59 (%)
- $\Delta E$: 43.29 (%)
- $r_T$: 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

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

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:

\[
\begin{align*}
\text{Minimize:} & \quad \sum_i |y_i| \\
\text{Subject to:} & \quad y_i + \sum_j a_{ij} \cdot x_j \leq b_i, \quad \forall i \text{ and } x_j \geq 0, \quad \forall j
\end{align*}
\]  

(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.

\[
\begin{align*}
\text{Minimize:} & \quad \sum_i y_1^i + y_2^i \\
\text{Subject to:} & \quad y_1^i - y_2^i + \sum_j a_{ij} \cdot x_j \leq b_i, \quad \forall i \\
& \quad x_j \geq 0, \quad \forall j \text{ and } y_1^i, y_2^i \geq 0, \quad \forall i
\end{align*}
\]  

(2)
CPF* Minimization
(Linear Modeling of Nonlinearity ...)

General LP Formulations involving Fraction

- General form of this type of programming:

\[
\text{Minimize : } \frac{\sum_j c_j \cdot x_j}{\sum_j d_j \cdot x_j} \\
\text{Subject to : } \sum_j a_{ij} \cdot x_j \leq b_i, \quad \forall i, \quad x_j \geq 0, \quad \forall j
\]

(1)

- Assume two new variables, \( z_0 = \frac{1}{d_0 + \sum d_j x_j} \) and \( x_j = \frac{z_j}{z_0} \).

- Using the new variables the formulation becomes.

\[
\text{Minimize : } c_0 \cdot z_0 + \sum_j c_j \cdot z_j \\
\text{Subject to : } \sum_j a_{ij} \cdot z_j - b_i \cdot z_0 \leq b_i, \quad \forall i \\
\sum_j d_j \cdot z_j + d_0 \cdot z_0 = 1, \quad z_0, z_j \geq 0, \quad \forall j
\]

(2)

- Once the new formulation is solved substitute \( z_j = x_j \cdot z_0 \) to get the result for \( x_j \).
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 $\text{DP}_{\text{peak}}$, we will normalize it with $P_{\text{peak}}$. 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.

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

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

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

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

(1)

The denominator is removed and introduced as a constraint.

\[
\text{Minimize: } \frac{1}{N} \sum_{c=1}^{N} P_c + \frac{1}{N} \sum_{c=1}^{N} |P - P_c|
\]

Subject to: Peak power constraints

(2)

The absolute is replaced with sum and the appropriate constraints.

\[
\text{Minimize: } \frac{1}{N} \sum_{c=1}^{N} P_c + \frac{1}{N} \sum_{c=1}^{N} (P + P_c)
\]

Subject to: Modified peak power constraints

(3)

After simplification,

\[
\text{Minimize: } \left( \frac{3}{N} \right) \sum_{c=1}^{N} P_c
\]

Subject to: Modified peak power constraints

(4)

Using decision variables,

\[
\text{Minimize: } \sum_{c} \sum_{i \in F_{a, c}} \sum_{v} \sum_{f} x_{i,c,v,f} \cdot \left( \frac{3}{N} \right) \cdot P(C_{\text{swi}, v, f})
\]

Subject to: Modified peak power constraints

(5)
CPF* Minimization: ILP Formulation (MVDFC)

- **Uniqueness Constraints**: ensure that every operation $o_i$ is scheduled to one unique control step and represented as,
  $\forall i, 1 \leq i \leq O, \sum_c \sum_v \sum_f x_{i,c,v,f} = 1$

- **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 a later control step and are; $\forall i,j, o_i$ belong to $\text{Pred}(o_j), \sum_v \sum_f \sum_{d=S_i \rightarrow E_i} dx_{i,c,v,f} - \sum_v \sum_f \sum_{d=S_j \rightarrow E_j} ex_{j,c,v,f} \leq -1$

- **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, $\forall c, 1 \leq c \leq N$ and $\forall v, \sum_{\{i \in F_{k,v}\}} \sum_f x_{i,c,v,f} \leq M_{k,v}$

- **Frequency Constraints**: lower operating voltage functional unit can not be scheduled in a higher frequency control step; these constraints are expressed as,
  $\forall i, 1 \leq i \leq O, \forall c, 1 \leq c \leq N$, if $f < v$, then $x_{i,c,v,f} = 0$. 

Dept. of CSE
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 \( c, \ 1 \leq c \leq N, \)

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

• **Modified Peak Power Constraints**: To eliminate the non-linearity introduced due to the absolute function introduced as, for all \( c, \ 1 \leq c \leq N, \)

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

**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,

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

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, \(\forall i, 1 \leq i \leq N,\)

\[
\sum_v \sum_l \{l = S_i \rightarrow (S_i + E_i + 1 - L_{i,v})\} y_{i,v,l,(l+L_{i,v}-1)} = 1
\]

- **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 \(\text{Pred}(o_j),\)

\[
\sum_v \sum_l \{l = S_i \rightarrow E_i\} (l + L_{i,v} - 1) y_{i,v,l,(l+L_{i,v}-1)} - \sum_v \sum_l \{l = S_j \rightarrow E_j\} l y_{j,v,l,(l+L_{j,v}-1)} \leq -1
\]
**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,

  $$\sum_{i \in F_{k,v}} \sum_{l} y_{i,v,l,(l+L_{i,v}-1)} \leq M_{k,v}$$

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

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

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

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

<table>
<thead>
<tr>
<th>Multipliers</th>
<th>ALUs</th>
<th>Serial No</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4V</td>
<td>3.3V</td>
<td>2.4V</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
## CPF* Minimization: Experimental Results

<table>
<thead>
<tr>
<th>R</th>
<th>C</th>
<th>$P_{PS}$ (mW)</th>
<th>$P_{PD}$ (mW)</th>
<th>$\Delta P_p$ (%)</th>
<th>$P_{mS}$ (mW)</th>
<th>$P_{mD}$ (mW)</th>
<th>$\Delta DP$ (%)</th>
<th>$P_S$ (mW)</th>
<th>$P_D$ (mW)</th>
<th>$\Delta P$ (%)</th>
<th>$E_S$ (nJ)</th>
<th>$E_D$ (nJ)</th>
<th>$\Delta E$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>1</td>
<td>17.28</td>
<td>4.56</td>
<td>73.61</td>
<td>0.46</td>
<td>0.35</td>
<td>74.97</td>
<td>8.87</td>
<td>2.42</td>
<td>72.72</td>
<td>2.96</td>
<td>1.57</td>
<td>46.8</td>
</tr>
<tr>
<td>X</td>
<td>2</td>
<td>17.28</td>
<td>4.56</td>
<td>73.61</td>
<td>0.46</td>
<td>0.35</td>
<td>74.97</td>
<td>8.87</td>
<td>2.42</td>
<td>72.72</td>
<td>2.96</td>
<td>1.57</td>
<td>46.8</td>
</tr>
<tr>
<td>P</td>
<td>3</td>
<td>17.28</td>
<td>4.56</td>
<td>73.61</td>
<td>0.46</td>
<td>0.9</td>
<td>78.24</td>
<td>8.87</td>
<td>2.61</td>
<td>70.57</td>
<td>2.96</td>
<td>1.6</td>
<td>46.0</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>17.51</td>
<td>4.62</td>
<td>73.62</td>
<td>0.23</td>
<td>0.12</td>
<td>73.96</td>
<td>8.82</td>
<td>2.35</td>
<td>73.36</td>
<td>4.9</td>
<td>2.6</td>
<td>47.2</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>25.92</td>
<td>6.84</td>
<td>73.61</td>
<td>0.23</td>
<td>0.12</td>
<td>73.84</td>
<td>8.82</td>
<td>2.36</td>
<td>73.24</td>
<td>4.9</td>
<td>2.6</td>
<td>47.2</td>
</tr>
<tr>
<td>R</td>
<td>3</td>
<td>17.51</td>
<td>4.67</td>
<td>73.33</td>
<td>0.23</td>
<td>0.45</td>
<td>75.58</td>
<td>8.82</td>
<td>2.5</td>
<td>71.66</td>
<td>4.9</td>
<td>2.6</td>
<td>46.2</td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>17.51</td>
<td>4.62</td>
<td>73.62</td>
<td>0.46</td>
<td>0.35</td>
<td>74.96</td>
<td>13.25</td>
<td>3.55</td>
<td>73.21</td>
<td>5.9</td>
<td>3.12</td>
<td>47.0</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>26.15</td>
<td>6.90</td>
<td>73.61</td>
<td>0.46</td>
<td>0.35</td>
<td>74.50</td>
<td>13.25</td>
<td>3.55</td>
<td>73.21</td>
<td>5.9</td>
<td>3.12</td>
<td>47.0</td>
</tr>
<tr>
<td>L</td>
<td>3</td>
<td>17.74</td>
<td>4.78</td>
<td>73.05</td>
<td>0.46</td>
<td>0.9</td>
<td>76.97</td>
<td>13.25</td>
<td>3.73</td>
<td>71.85</td>
<td>5.9</td>
<td>3.17</td>
<td>46.2</td>
</tr>
<tr>
<td>I</td>
<td>1</td>
<td>25.92</td>
<td>8.88</td>
<td>65.74</td>
<td>0.23</td>
<td>0.12</td>
<td>65.9</td>
<td>11.03</td>
<td>3.5</td>
<td>68.36</td>
<td>4.9</td>
<td>3.05</td>
<td>37.7</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>25.92</td>
<td>6.84</td>
<td>73.61</td>
<td>0.23</td>
<td>0.12</td>
<td>73.84</td>
<td>11.03</td>
<td>2.98</td>
<td>72.98</td>
<td>4.9</td>
<td>2.6</td>
<td>47.96</td>
</tr>
<tr>
<td>R</td>
<td>3</td>
<td>17.51</td>
<td>4.67</td>
<td>73.34</td>
<td>0.23</td>
<td>0.45</td>
<td>75.58</td>
<td>8.82</td>
<td>2.57</td>
<td>70.86</td>
<td>4.9</td>
<td>2.64</td>
<td>46.2</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>8.87</td>
<td>2.34</td>
<td>73.62</td>
<td>0.23</td>
<td>0.12</td>
<td>74.1</td>
<td>4.5</td>
<td>1.22</td>
<td>72.9</td>
<td>5.0</td>
<td>2.64</td>
<td>47.2</td>
</tr>
<tr>
<td>R</td>
<td>2</td>
<td>8.87</td>
<td>2.34</td>
<td>73.62</td>
<td>0.23</td>
<td>0.12</td>
<td>74.1</td>
<td>4.5</td>
<td>1.22</td>
<td>72.9</td>
<td>5.0</td>
<td>2.64</td>
<td>47.2</td>
</tr>
<tr>
<td>F</td>
<td>3</td>
<td>8.87</td>
<td>2.39</td>
<td>73.05</td>
<td>0.23</td>
<td>0.45</td>
<td>77.6</td>
<td>4.5</td>
<td>1.4</td>
<td>68.9</td>
<td>5.0</td>
<td>2.74</td>
<td>45.3</td>
</tr>
</tbody>
</table>
## CPF* Minimization: Experimental Results …

### MVDFC Vs MVMC % Reduction

<table>
<thead>
<tr>
<th>Power</th>
<th>MVDFC</th>
<th>MVMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak Power</td>
<td>71.70</td>
<td>26.44</td>
</tr>
<tr>
<td>Peak Power Differential</td>
<td>74.0</td>
<td>26.73</td>
</tr>
<tr>
<td>Average Power</td>
<td>70.82</td>
<td>22.52</td>
</tr>
<tr>
<td>Energy</td>
<td>44.36</td>
<td>39.05</td>
</tr>
<tr>
<td>Energy Delay Product</td>
<td>17.31</td>
<td>17.99</td>
</tr>
</tbody>
</table>
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)

**Spatial Domain**

Input Image → Watermark Insertion Module → DCT → Quantizer → Entropy Encoder → Compressed Image

**DCT Domain**

Input Image → DCT → Watermark Insertion Module → Quantizer → Entropy Encoder → Compressed Image
Digital Still Camera

Figure 9.2. Secure Digital Still Camera: Schematic View
Figure 9.2. Secure Digital Still Camera: Schematic View
Spatial Invisible: Algorithm (Robust)

Table 9.1. Notations used to Explain Spatial Domain Watermarking Algorithms

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>Original image (gray image)</td>
</tr>
<tr>
<td>$W$</td>
<td>Watermark image (binary or ternary image)</td>
</tr>
<tr>
<td>$(i, j)$</td>
<td>A pixel location</td>
</tr>
<tr>
<td>$I_W$</td>
<td>Watermarked image</td>
</tr>
<tr>
<td>$N_I \times N_I$</td>
<td>Image dimension</td>
</tr>
<tr>
<td>$N_W \times N_W$</td>
<td>Watermark dimension</td>
</tr>
<tr>
<td>$E, E_1, E_2$</td>
<td>Watermark embedding functions</td>
</tr>
<tr>
<td>$D$</td>
<td>Watermark detection function</td>
</tr>
<tr>
<td>$r$</td>
<td>Neighborhood radius</td>
</tr>
<tr>
<td>$I_N$</td>
<td>Neighborhood image (gray image)</td>
</tr>
<tr>
<td>$K$</td>
<td>Digital (watermark) key</td>
</tr>
<tr>
<td>$\alpha_1, \alpha_2$</td>
<td>Scaling constants (watermark strength)</td>
</tr>
</tbody>
</table>
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**:
  \[
  E_1(I, I_N) = (1 - \alpha_1)I_N(i, j) + \alpha_1 I(i, j) \\
  E_2(I, I_N) = (1 - \alpha_1)I_N(i, j) - \alpha_2 I(i, j)
  \]

- **Neighborhood pixel gray value**: Calculated as,
  \[
  I_N(i, j) = \frac{I(i+1,j)+I(i+1,j+1)}{2} + I(i, j+1)
  \]
Spatial Invisible: Algorithm (Fragile)

(a) Watermark Insertion

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

\[
I_W[0 \rightarrow k-1](i,j) = I[0 \rightarrow k-1](i,j)
\]

\[
I_W[k](i,j) = I[k](i,j) \text{XOR} W(i,j)
\]

\[
I_W[k + 1 \rightarrow 7](i,j) = I[k + 1 \rightarrow 7](i,j)
\]
Spatial Invisible: Overall Datapath Architecture
Spatial Invisible: Overall Controller

Initial state

START = 0

Read image and read/create watermark

START = 1
IM_COMPLETED = 0

Perform watermarking

IM_COMPLETED = 1

Write watermarked pixels

IM_COMPLETED = 1
WM_COMPLETED = 0

Display the watermarked image

WM_COMPLETED = 1
IM_COMPLETED = 0
Spatial Invisible: Datapath Layout
Spatial Invisible: Controller Layout
Table 9.4. Overall Chip Statistics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area (with RAM)</td>
<td>$15.012 \times 14.225 \text{mm}^2$</td>
</tr>
<tr>
<td>Number of gates (with RAM)</td>
<td>$1188K$</td>
</tr>
<tr>
<td>Number of gates (without RAM)</td>
<td>4820</td>
</tr>
<tr>
<td>Clock frequency (with RAM)</td>
<td>$151 \text{MHz}$</td>
</tr>
<tr>
<td>Clock frequency (without RAM)</td>
<td>$545 \text{MHz}$</td>
</tr>
<tr>
<td>Number of I/O pins</td>
<td>25</td>
</tr>
<tr>
<td>Power (with RAM)</td>
<td>$24mW$</td>
</tr>
<tr>
<td>Power (without RAM)</td>
<td>$2.0547mW$</td>
</tr>
</tbody>
</table>

**Diagram:**

- IM_DATA_IN
- WM_DATA_IN
- WM_DATA_SELECT
- ROBUST/FRAGILE
- START
- RESET
- CLOCK

**Nodes:**

- SPATIAL DOMAIN
- INVISIBLE
- WATERMARKING
- ENCODER

**Connections:**

- DATA_OUT
- BUSY
- DATA_READY
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>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>Original (or host) image (a grayscale image)</td>
</tr>
<tr>
<td>$W$</td>
<td>Watermark image (a grayscale image)</td>
</tr>
<tr>
<td>$(m, n)$</td>
<td>A pixel location</td>
</tr>
<tr>
<td>$I_W$</td>
<td>Watermarked image</td>
</tr>
<tr>
<td>$N_I \times N_I$</td>
<td>Original image dimension</td>
</tr>
<tr>
<td>$N_W \times N_W$</td>
<td>Watermark image dimension</td>
</tr>
<tr>
<td>$i_k$</td>
<td>The $k^{th}$ block of the original image $I$</td>
</tr>
<tr>
<td>$w_k$</td>
<td>The $k^{th}$ block of the watermark image $W$</td>
</tr>
<tr>
<td>$i_{Wn}$</td>
<td>The $k^{th}$ block of the watermarked image $I_W$</td>
</tr>
<tr>
<td>$\alpha_k$</td>
<td>Scaling factor for $k^{th}$ block (used for host image scaling)</td>
</tr>
<tr>
<td>$\beta_k$</td>
<td>Embedding factor for $k^{th}$ block (used for watermark image scaling)</td>
</tr>
<tr>
<td>$\mu_I$</td>
<td>Mean gray value of the original image $I$</td>
</tr>
<tr>
<td>$\mu_{I_k}$</td>
<td>Mean gray value of the original image block $i_k$</td>
</tr>
<tr>
<td>$\sigma_{I_k}$</td>
<td>Variance of the original image block $i_k$</td>
</tr>
<tr>
<td>$\alpha_{max}$</td>
<td>The maximum value of $\alpha_k$</td>
</tr>
<tr>
<td>$\alpha_{min}$</td>
<td>The minimum value of $\alpha_k$</td>
</tr>
<tr>
<td>$\beta_{max}$</td>
<td>The maximum value of $\beta_k$</td>
</tr>
<tr>
<td>$\beta_{min}$</td>
<td>The minimum value of $\beta_k$</td>
</tr>
<tr>
<td>$I_{white}$</td>
<td>Gray value corresponding to pure white pixel</td>
</tr>
<tr>
<td>$\alpha_I$</td>
<td>A global scaling factor</td>
</tr>
<tr>
<td>$C_1, C_2, C_3, C_4$</td>
<td>Linear regression co-efficients</td>
</tr>
</tbody>
</table>
Spatial Visible : Algorithm 1

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

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

- Assuming \( I_{white} = 256 \), simplified to:

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

- Fitting piecewise linear model and regression co-efficients:

\[ I_W(m, n) = \begin{cases} 
I(m, n) + \left( \frac{\alpha_f}{903.3} \right) W(m, n) I(m, n) & \text{for } I(m, n) \leq 2 \\
I(m, n) + \left( \frac{\alpha_f C_1}{6.0976} \right) W(m, n) I(m, n) & \text{for } 2 < I(m, n) \leq 64 \\
I(m, n) + \left( \frac{\alpha_f C_2}{6.0976} \right) W(m, n) I(m, n) & \text{for } 64 < I(m, n) \leq 128 \\
I(m, n) + \left( \frac{\alpha_f C_3}{6.0976} \right) W(m, n) I(m, n) & \text{for } 128 < I(m, n) \leq 192 \\
I(m, n) + \left( \frac{\alpha_f C_4}{6.0976} \right) W(m, n) I(m, n) & \text{for } 192 < I(m, n) < 256
\end{cases} \]
Spatial Visible : Algorithm 2

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

\[ iW_k = \alpha_k i_k + \beta_k w_k \quad k = 1, 2, \ldots \]

- The scaling and embedding factors are found out as,

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

- Values are scaled to proper range:

\[
\alpha_k = \alpha_{\text{min}} + (\alpha_{\text{max}} - \alpha_{\text{min}}) \frac{1}{\sigma_{I_k}} \exp\left(-\left(\hat{\mu}_{I_k} - \hat{\mu}_I\right)^2\right) \\
\beta_k = \beta_{\text{min}} + (\beta_{\text{max}} - \beta_{\text{min}}) \hat{\sigma}_{I_k} \left(1 - \exp\left(-\left(\hat{\mu}_{I_k} - \hat{\mu}_I\right)^2\right)\right)
\]
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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>$3.34 \times 2.89,\text{mm}^2$</td>
</tr>
<tr>
<td>Number of gates</td>
<td>28469</td>
</tr>
<tr>
<td>Clock frequency</td>
<td>$292.27,\text{MHz}$</td>
</tr>
<tr>
<td>Number of I/O pins</td>
<td>72</td>
</tr>
<tr>
<td>Power</td>
<td>6.9286,\text{mW}</td>
</tr>
</tbody>
</table>
**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

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

**Watermarked Images using Algorithm 1**
The invisible watermark insertion involves addition of random numbers to relatively perceptual significant co-efficients of the host image.

\[ c_{IW_k}(m,n) = c_{Ik}(m,n) + \alpha r_k(m,n) \]

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

\[ c_{IW_k} = \alpha_k c_{Ik} + \beta_k c_{Wk} \]

Current scaling and embedding factors are obtained as,

\[ \alpha_k^c = \sigma_{ACI_k} \exp\left(-(\mu_{DCI_k}^* - \mu_{DCI}^*)^2\right) \]
\[ \beta_k^c = \frac{1}{\sigma_{ACI_k}} \left(1 - \exp\left(-(\mu_{DCI_k}^* - \mu_{DCI}^*)^2\right)\right) \]

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


Thank you