Fault Tolerant Design for Low Power Hierarchical Search Motion Estimation Algorithms

By Charvi Dhoot*[Π], Vincent J. Mooney[&Π],
-Shubhajit Roy Chowdhury*, Lap Pui Chau[#Π]

*International Institute of Information Technology, Hyderabad, India
[&] School of Electrical and Computer Engineering, Georgia Institute of Technology, Georgia, USA
[#] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[Π]Institute of Sustainable and Applied Infodynamics, Nanyang Technological University, Singapore

Outline

- Motivation
- Research Goal
- Research Problem Statement
- Background: Motion Estimation
  - Three Step Search (TSS) Algorithm
  - Performance Metric
  - Architecture for TSS Algorithm
- Background: Probabislitic CMOS
- Proposed Methodology
  - Modeling the PCMOS Architecture
  - Multiple Candidate Three Step Search (MCTSS)
  - Architecture for MCTSS Algorithm
- Results
- Conclusion

Motivation

- Moore’s law, proposed around 1970, has driven the semiconductor industry to innovate itself every 26 months and to push the limit on the computing power.

- Today, the industry is growing increasingly skeptic towards this law. For what might be true, beliefs are we might be able to push the silicon to about 8nm, enough to keep up with the law until 2020 but the question everyone is concerned about is how and at what cost?

Motivation

- The cost of fabrication, mask set, and turn around times increases each generation
- The mask set cost for 22 nm is estimated to be about more than a million dollars!

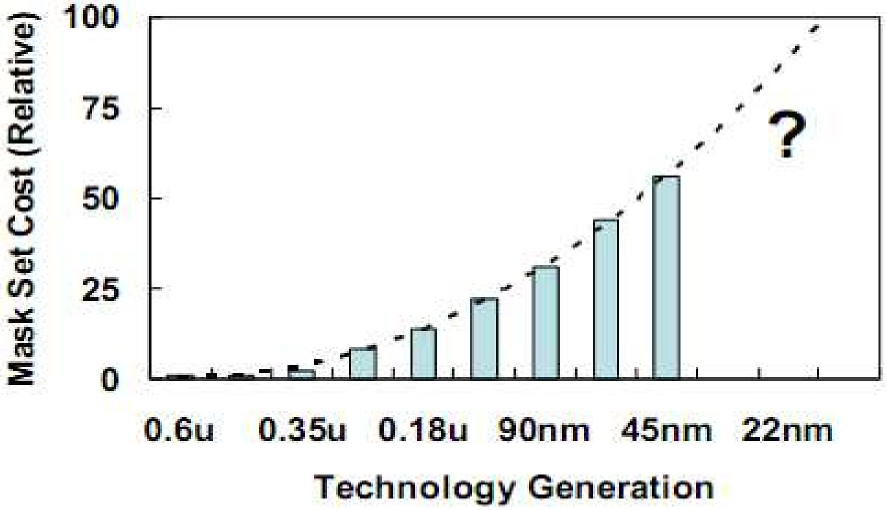


Fig. 1: Mask Set Cost Trend w.r.t. technology nodes

*S. Borkar, Design Perspectives on 22nm CMOS and Beyond, DAC’09, July, 2009, pp. 93-94

Motivation

- The reliability of computing via future technology nodes is seriously being questioned with predictions about thermal noise, and process variations resulting in soft errors.

- What are we letting go if we decide to stop?
  - Double transistor integration
  - 30% reduction in gate delay
  - 65% reduction in energy per logic operation
  - 50% reduction in power consumption

- One possible Solution:
  - Resilience and error tolerance!

Research goal

- Low power design for Motion Estimation in the presence of thermal noise responsible for soft errors.

- Why Motion Estimation?
  - Computationally the most intensive part of video compression.
  - As for video compression, much of our advancements in wireless technology and embedded systems enable and empower us with high speed online video streaming, transmission of image and video data, video conferencing all of which require **low power video compression**!

Research Problem Statement

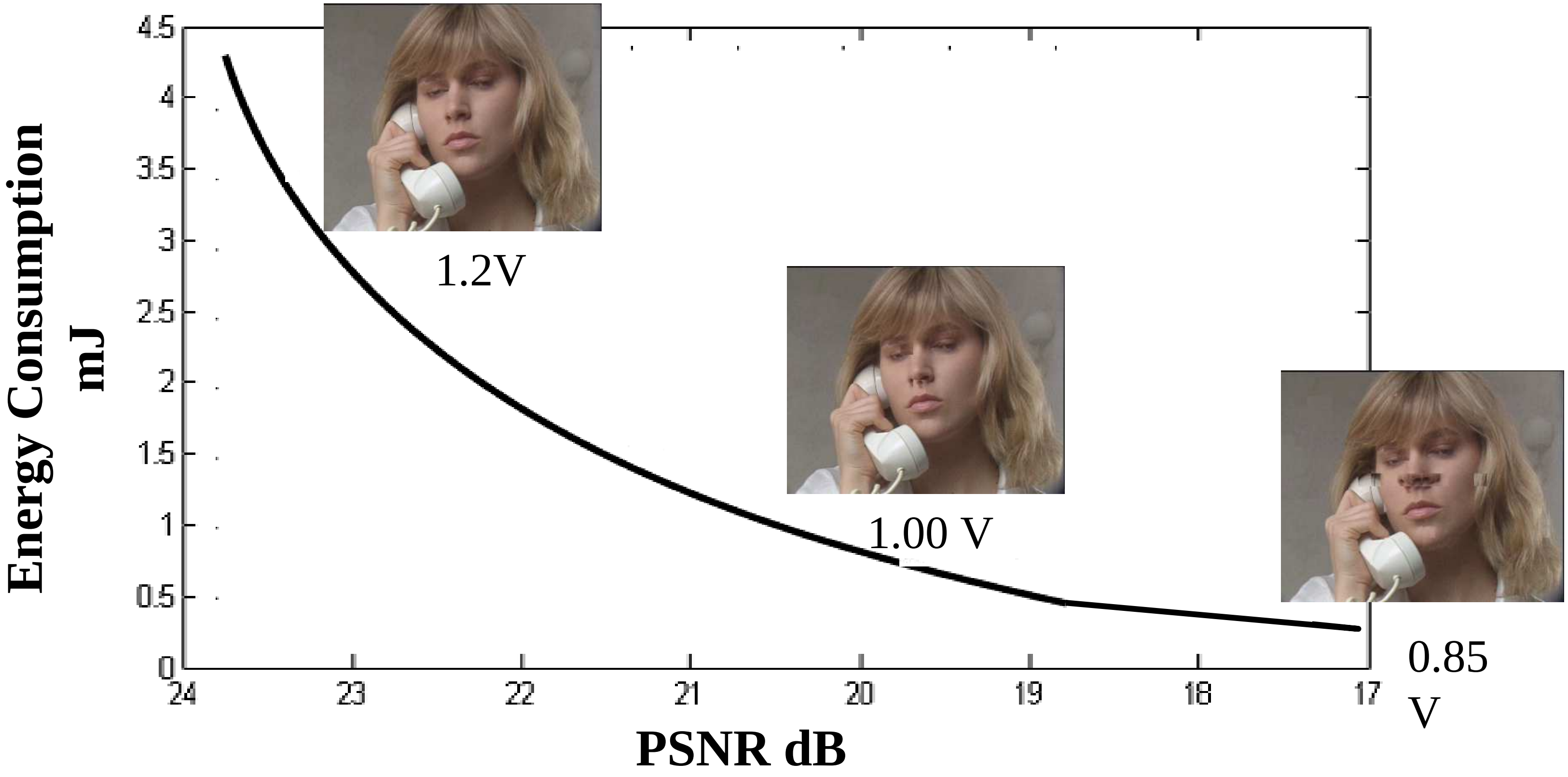


Fig. 2: Decreasing Picture Quality with Voltage Scaling

- The picture quality decreases as error increases with voltage scaling. The goal was to find algorithmic modifications to motion estimation such that energy savings could be increased while maintaining the quality requirements.

Background: Motion Estimation

- Uses temporal correlation present between subsequent frames to reduce redundancies for compression
- Represents the transformations from one frame to another in terms of motion vectors
- The most popular method used to calculate the motion vectors is block matching

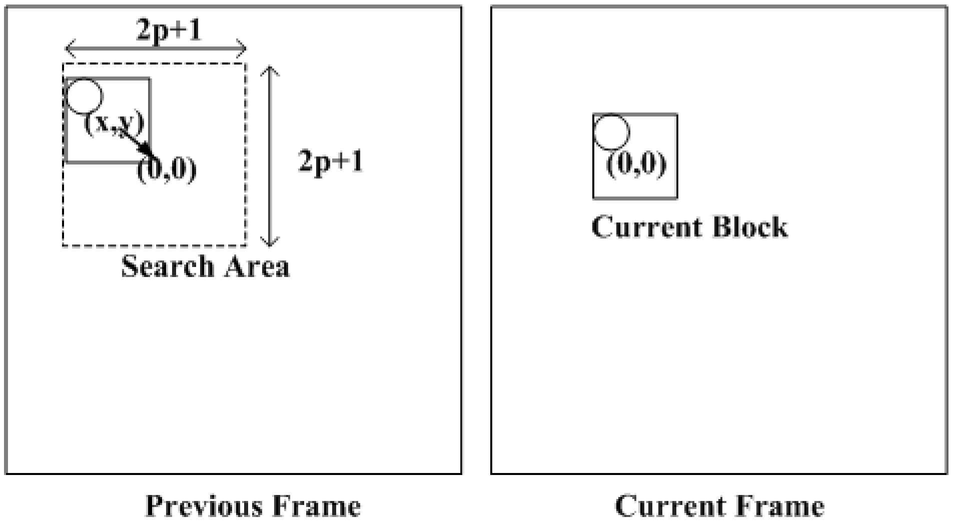


Fig. 3: Block Matching

Background: Motion Estimation

- The criterion for arriving upon the best match out of the candidate macro-blocks is sum of absolute differences (SAD).

- SAD is calculated by summing up the absolute difference between pixel intensity values of the current block ‘a’ and the corresponding pixel intensity values of the candidate block ‘b’.

$$SAD = \sum_{j}^{N} \sum_{i}^{N} \left| a(i, j) - b(i, j) \right|$$

- The candidate macro-block locations are decided by a block matching algorithm. We consider Three Step Search (TSS) which belongs to a class of hierarchical search motion estimation algorithm. The search strategy is to move from a coarse to fine search with every step.

Three Step Search Algorithm

1. Use the current macro-block location as the reference location and take a search area of (±7, ±7) around this location
2. Start with an initial step size Δ=4
3. Evaluate all candidate locations at (± Δ, ± Δ) around the reference/winner candidate for the previous location. Take the winner candidate to be the one with the least SAD.
4. Reduce the step size Δ = Δ/2 repeat (3) until Δ ≥ 1

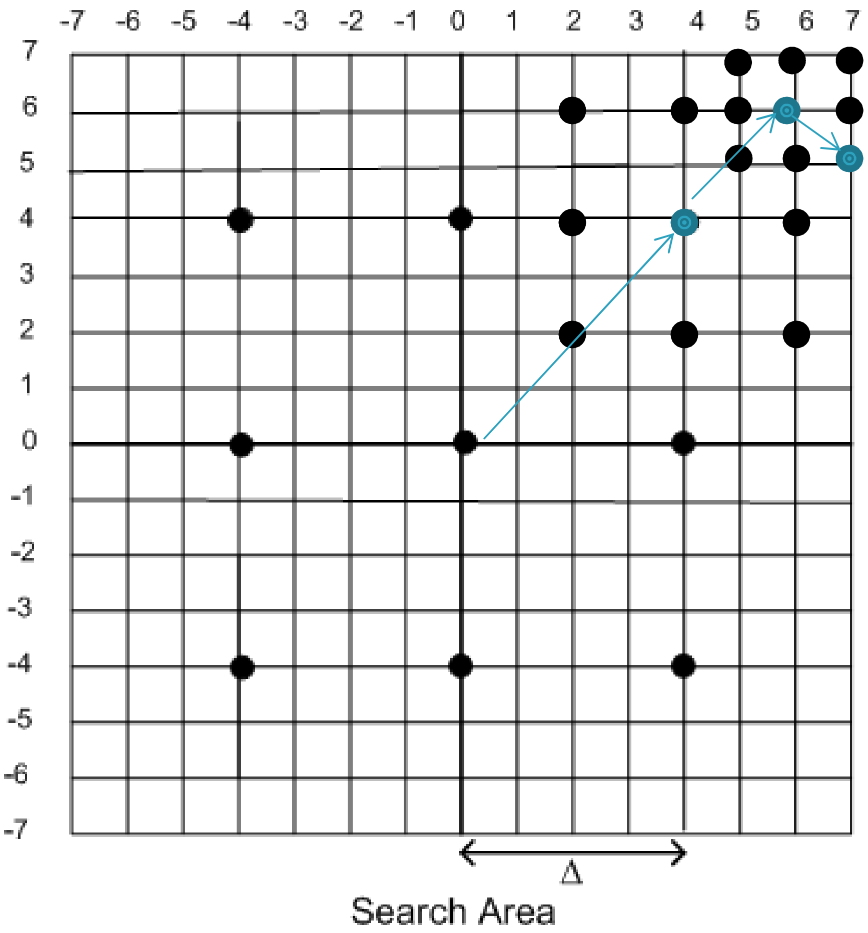


Fig. 4: Search Strategy for Three Step Search Algorithm

Motion Estimation: Performance Metric

- Peak Signal to Noise Ratio

$$PSNR = 10 \times \log\left(\frac{255^2}{(1/(H \times W)) \sum_{i,j}^{H,W} \left(F_I(i,j) - F_{MC}(i,j)\right)^2} \right)$$

- where H and W are the dimensions of the frame.
- $F_I(i,j)$ and $F_{MC}(i,j)$ are the pixel luminance values for the input and the motion compensated frames.

Tree Architecture for TSS

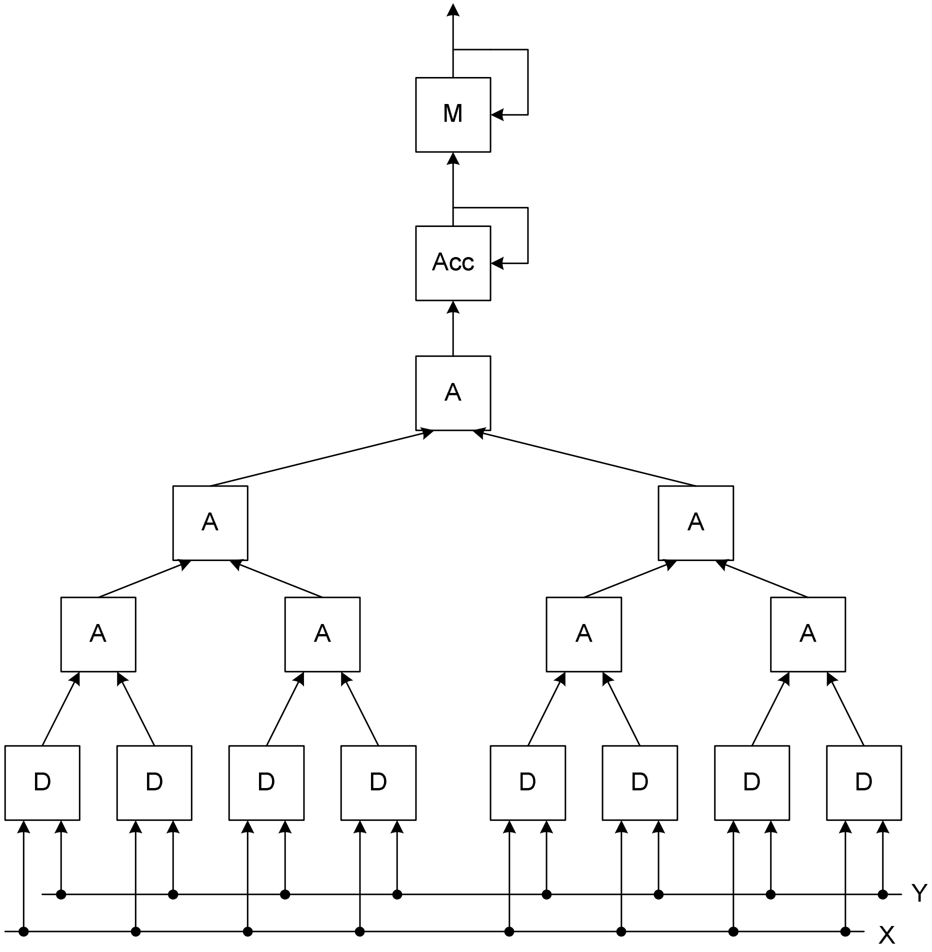


Fig. 5: Systolic Array Architecture for FSBMA*

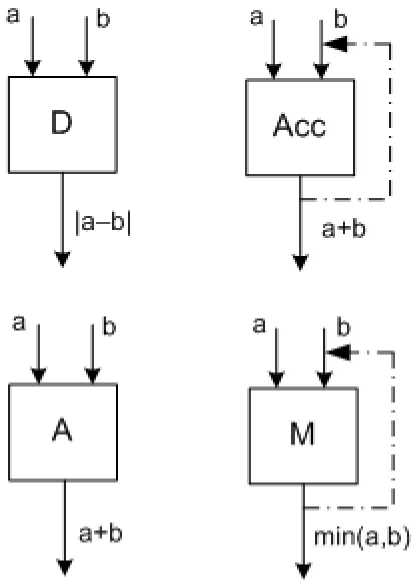


Fig. 6: Subtractor, Accumulator, Adder and Comparator Units

* T. Komarek and P. Pirsch, “Array architectures for block matching algorithms,” *IEEE Transactions on Circuits and System*, vol. 36, no. 10, pp. 1301-1308, Oct. 1989

Background: Probabilistic CMOS

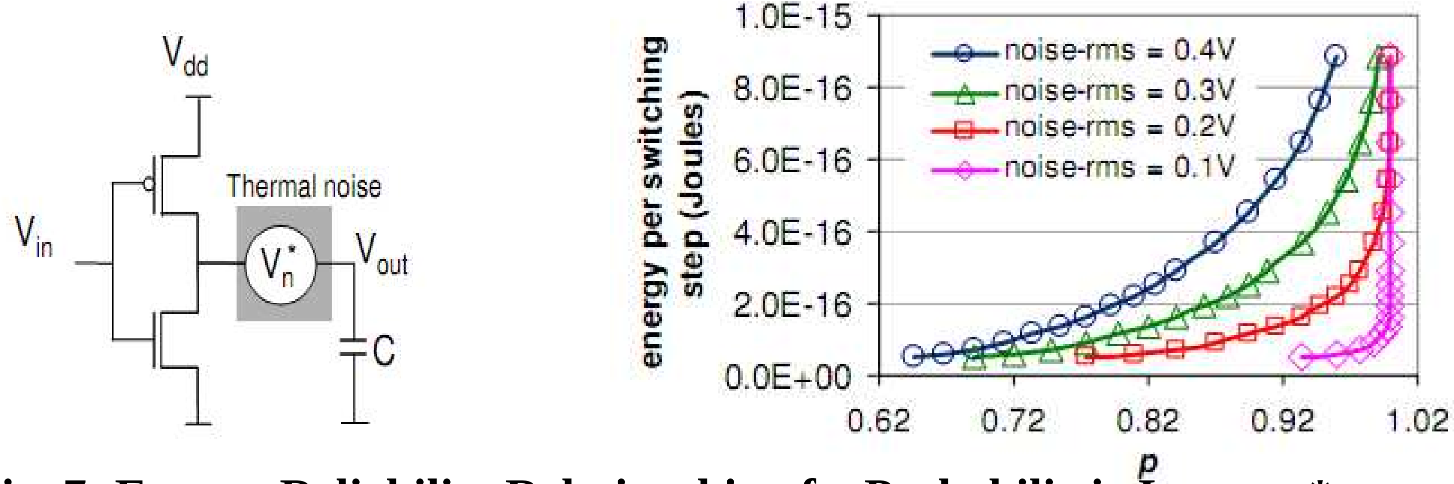


Fig. 7: Energy-Reliability Relationship of a Probabilistic Inverter*

- ➢ A PCMOS gate is modeled by coupling a noise source at the output of the gate
- ➢ Experiments with different values of the noise RMS showed that Energy decreases exponentially with respect to increase in the probability of error

* P. Korkmaz, B. E. S. Akgul, L. N. Chakrapani, and K. V. Palem, "Advocating noise as an agent for ultra low-energy computing: Probabilistic CMOS devices and their characteristics," Japanese Journal of Applied Physics, vol. 45, pp. 3307–3316, Apr. 2006.

Outline

- Motivation
- Research Goal
- Research Problem Statement
- Background: Motion Estimation
  - Three Step Search (TSS) Algorithm
  - Performance Metric
  - Architecture for TSS Algorithm
- Background: Probabislitic CMOS
- **Proposed Methodology**
  - Modeling the PCMOS Architecture
  - Multiple Candidate Three Step Search
  - Architecture for MCTSS
- Results
- Conclusion

Modeling the PCMOS Architecture

- Building Architectures using PCMOS Gates

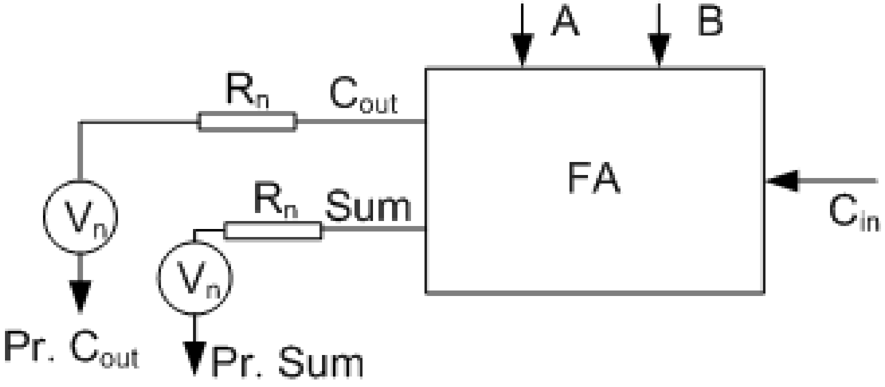


Fig. 8: Probabilistic Full Adder

- Fig. 8 above shows a probabilistic full adder modeled from a deterministic full adder
- The modeling involves coupling a noise source at the output of the gate
- All the gates in the architecture are modeled as PCMOS gates

Modeling the PCMOS Architecture

- Measuring the Error and modeling the error rates observed into a C code

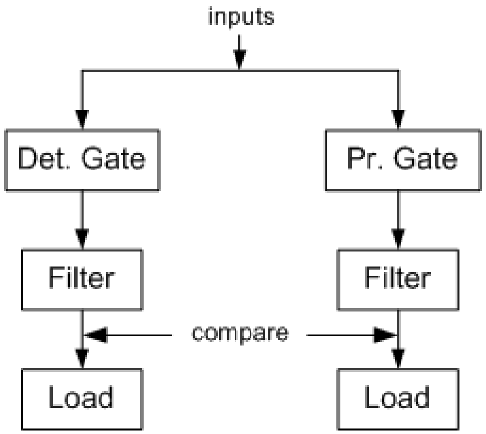


Fig. 9: Three Stage Model to Estimate Error Rates for Pr. Circuits

- We first measure the error for a single gate in the Architecture using the three stage modeling $^{\&}$
- The filter and load gates are deterministic versions of the gates attached to the gate in the Architecture whose error rate is being measured
- Error is checked at output of the filter gate for over 1 lac random input configurations

Modeling the PCMOS Architecture

- Calculating the Energy Consumption
  - The entire Architecture is built through PCMOS Gates in HSPICE
  - The supply voltage for the architecture is scaled as per the error tolerance of the application: Motion Estimation decided by C simulations using error values from HSPICE simulations
  - The base case for comparison is the architecture maintained at 1.2 V for Synopsys 90 nm Generic Library

[&]A. Singh, A. Basu, K.V. Ling, and V. Mooney, "Modeling Multi-output Filtering Effects in PCMOS," *Proceedings of the VLSI Design and Test Conference (VLSIDAT 2011)*, April 2011.. Komarek and P. Pirsch, "Array architectures for block matching algorithms," *IEEE Transactions on Circuits and System*, vol. 36, no. 10, pp. 1301-1308, Oct. 1989

Multiple Candidate Three Step Search Algorithm (MCTSS)

- The MC-TSS evaluates nine candidate locations in the first step to select three winner candidate locations with the least SAD.
- The next step involves a finer search around all three winner candidates to select the next three winner candidates.
- The number of candidates locations increases from 25 to 57
- To keep the total number of calculations almost the same, we halve the number of SAD computations

$$SAD = \sum_{j}^{N} \sum_{i}^{N/2} \left| a(2i, j) - b(2i, j) \right|$$

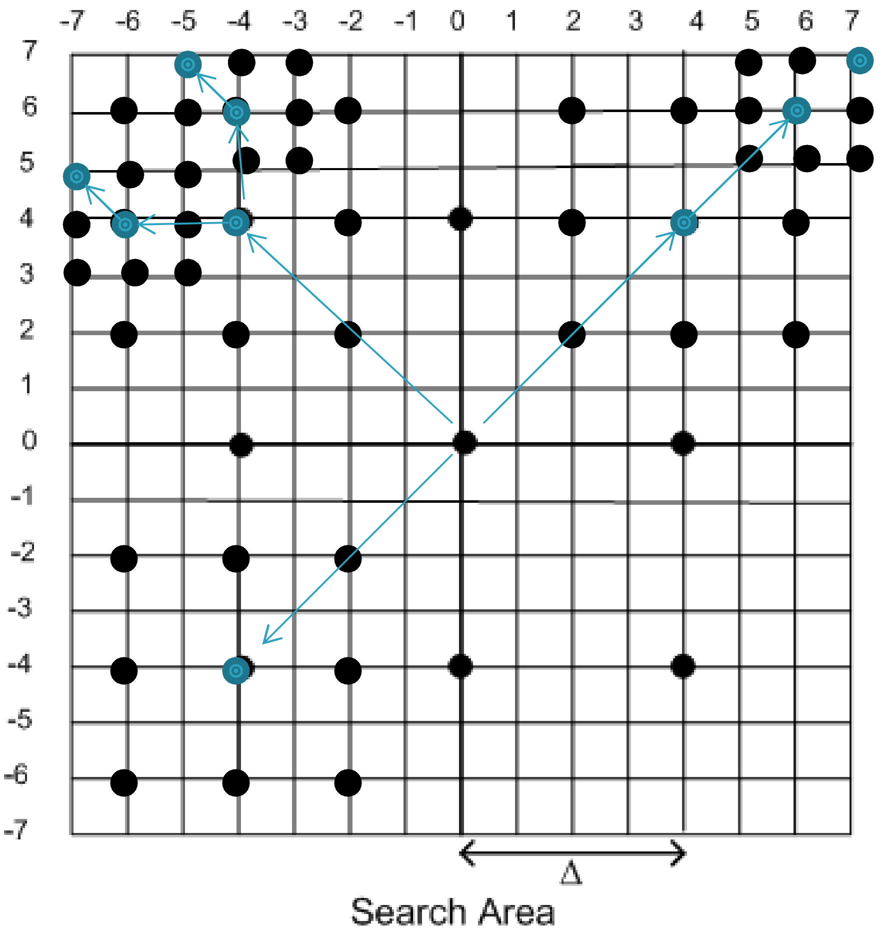


Fig. 10: MCTSS Search Strategy

Architecture for MCTSS

- The architecture for MC-TSS is also the tree architecture with a simple modification for the comparator and register unit that stores the minimum SAD.
- The required number of comparators increases to three.
- In Fig. 11, SAD_C corresponds to the SAD of the candidate block, and SAD_{M1}, SAD_{M2} and SAD_{M3} correspond to the three least SADs.

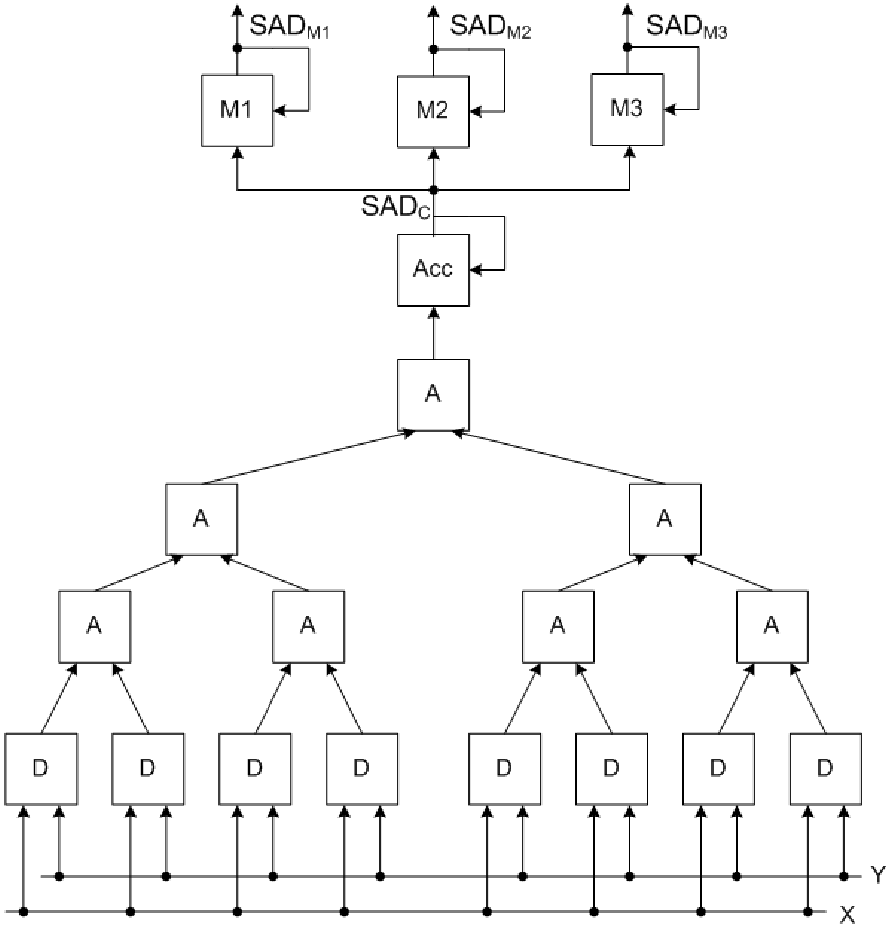
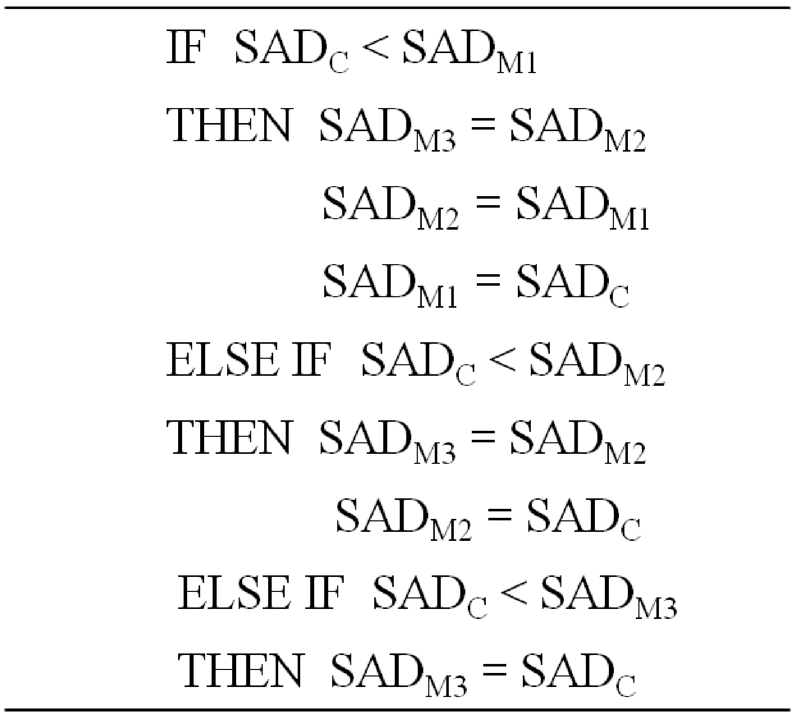


Fig. 11: MCTSS Tree Architecture

Architecture for MCTSS



```
IF $SAD_C < SAD_{M1}$
THEN $SAD_{M3} = SAD_{M2}$
     $SAD_{M2} = SAD_{M1}$
     $SAD_{M1} = SAD_C$
ELSE IF $SAD_C < SAD_{M2}$
THEN $SAD_{M3} = SAD_{M2}$
     $SAD_{M2} = SAD_C$
ELSE IF $SAD_C < SAD_{M3}$
THEN $SAD_{M3} = SAD_C$
```

Fig. 12: Logic for Data Movement between Register Units

- The number of register units required to store the least SADs also increases to three.
- The movement of data between these registers is dependent on the outcome of the three comparators.
- The logic to implement this is shown in Fig 12.

Architecture for MCTSS

- The logic to follow described in Fig. 12 can be implemented with the help of shift registers.
- Fig. 13 describes the shift register unit for the *j*th bit of SAD_C, SAD_{M1}, SAD_{M2} and SAD_{M3}, and the gate level implementation of the logic required for movement of SAD values between registers dependent on the Sign bits provided by the comparators.
- Unit is replicated sixteen times for all the 16 bits of the SADs.

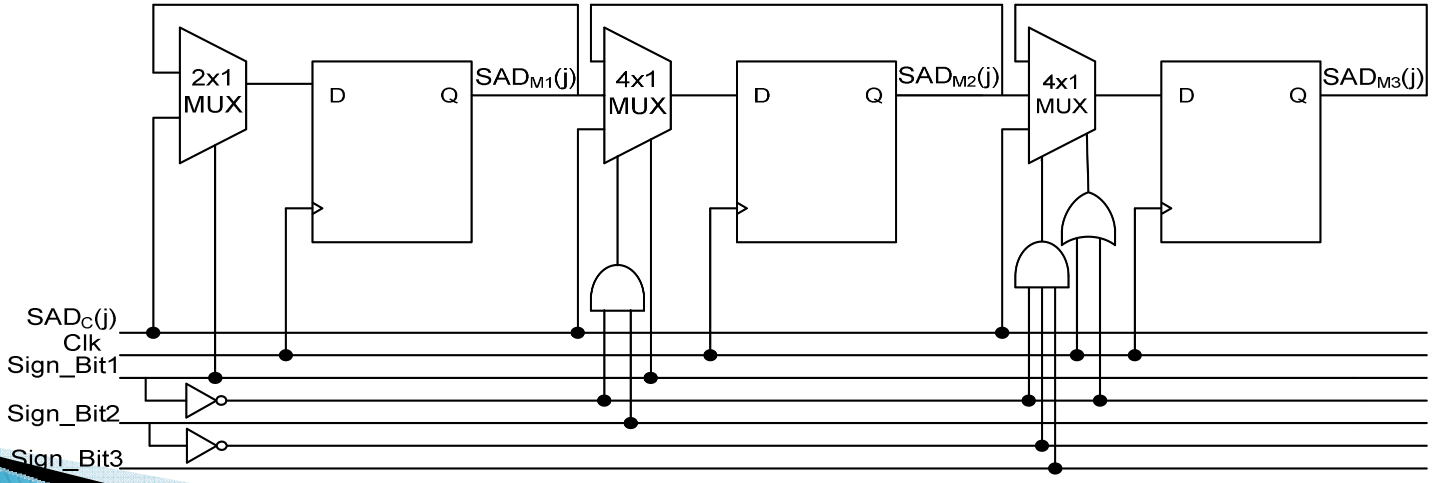


Fig. 13: Shift Register Unit for Data Movement

Results

- Experimental Set-Up for PCMOS based Architecture
  - All simulations are done in HSPICE using Synopsys 90nm Generic Library
  - Transistor level netlist developed for the entire architecture with noise modeling for the gates of the architecture
  - Choice of noise RMS is made such that no errors are observed at the output of the gates at the nominal supply voltage of 1.2 V, which was found empirically to be 0.2 V
  - Supply voltage is then scaled down to 1.15 V, 1.1 V down till 0.7 V with a step size of 0.05 V

Results

- Experimental Set-Up for Motion Estimation
  - All simulations are done using the MPEG-2 Test Model 5 Codec
  - Input video sequences are standard CIF video sequences of size 352x288 with varying type of motion from slow to fast
  - A C code is developed for motion estimation that accounts for the errors through the PCMOS architecture
  - Error increases in the output of motion estimation due to increase in error probabilities with scaling of supply voltage
  - Choice of supply voltage is the one that results in maximum energy savings for less than 0.5 dB degradation in PSNR

Results

- Results for percentage of the times the correct winner candidate is present amongst 1 to 4 best candidates
  - Simulations are carried out using MPEG-2 Test Model-5
  - The number of winner candidates to keep is increased from 1 to 4 to see keeping what number of candidates would give the best result
  - The winner candidates for the four cases are compared with the actual winner candidate of TSS to arrive at the percentage
  - Difference between percentages for 3 winner candidates and 4 winner candidates is less than 2%

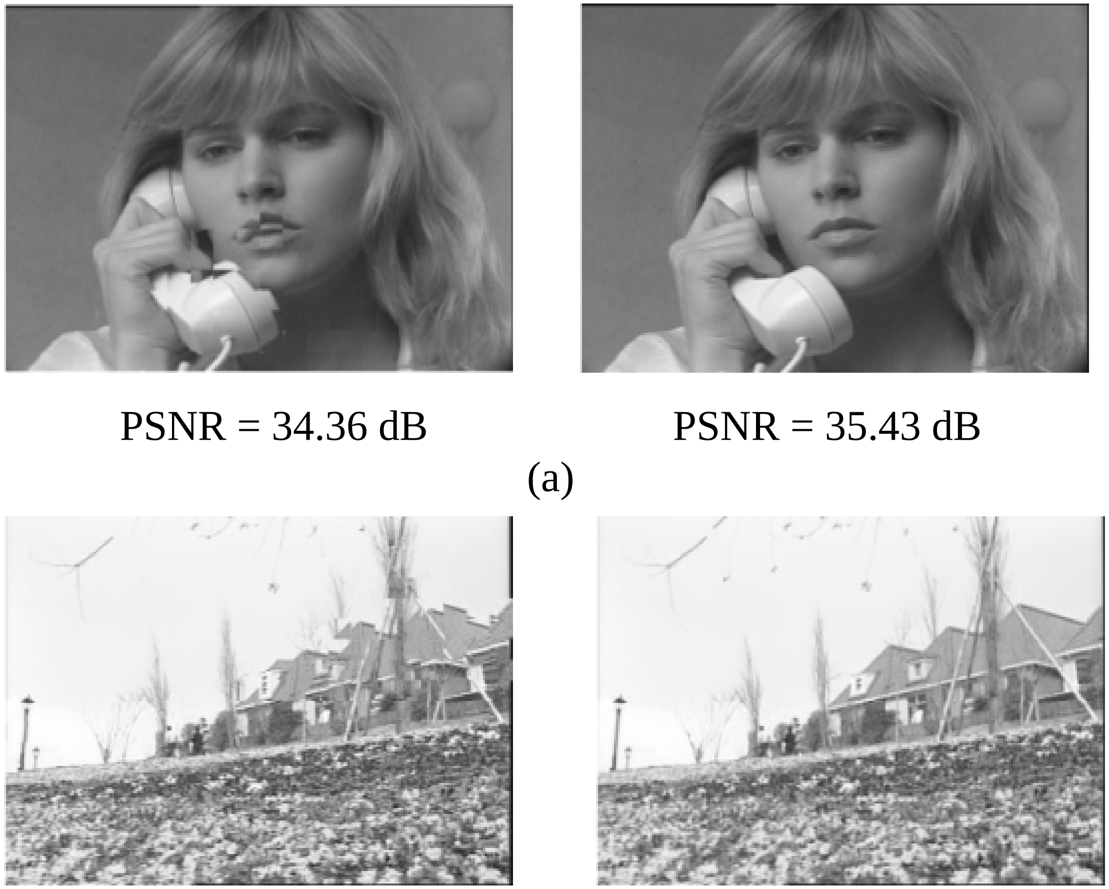
Video Sequence	Circuit Voltage (V)	Winner-1	Winner-2	Winner-3	Winner-4
Susie	0.95	92.54%	95%	98%	98.8%
Mobile Calendar	0.85	89%	92%	96%	97.15%
Flower Garden	0.85	82%	90%	94.21%	96.35%
Foreman	0.90	92.47%	95%	97.26%	98%

Results

- Results tabulated for energy savings with PSNR loss within 0.5dB
  - Base Case: TSS at 1.2V
  - Case 1: TSS with voltage scaling
  - Case 2: MCTSS with voltage scaling

Video Sequences	Base Case: PSNR (dB)	Case 1 (TSS)			Case 2 (MCTSS)		
		PSNR (dB)	Energy Savings (%)	Circuit Supply Voltage	PSNR (dB)	Energy Savings (%)	Circuit Supply Voltage
Susie	35.64	35.21	40	1.05	35.43	55	0.95
Mobile Calendar	23.72	23.23	57	0.95	23.36	70	0.85
Flower Garden	25.2	24.74	57	0.95	25.02	70	0.85
Foreman	31.3	31.03	49	1.00	30.89	64	0.90

Results



PSNR = 34.36 dB

PSNR = 35.43 dB

(a)

PSNR = 23.21 dB

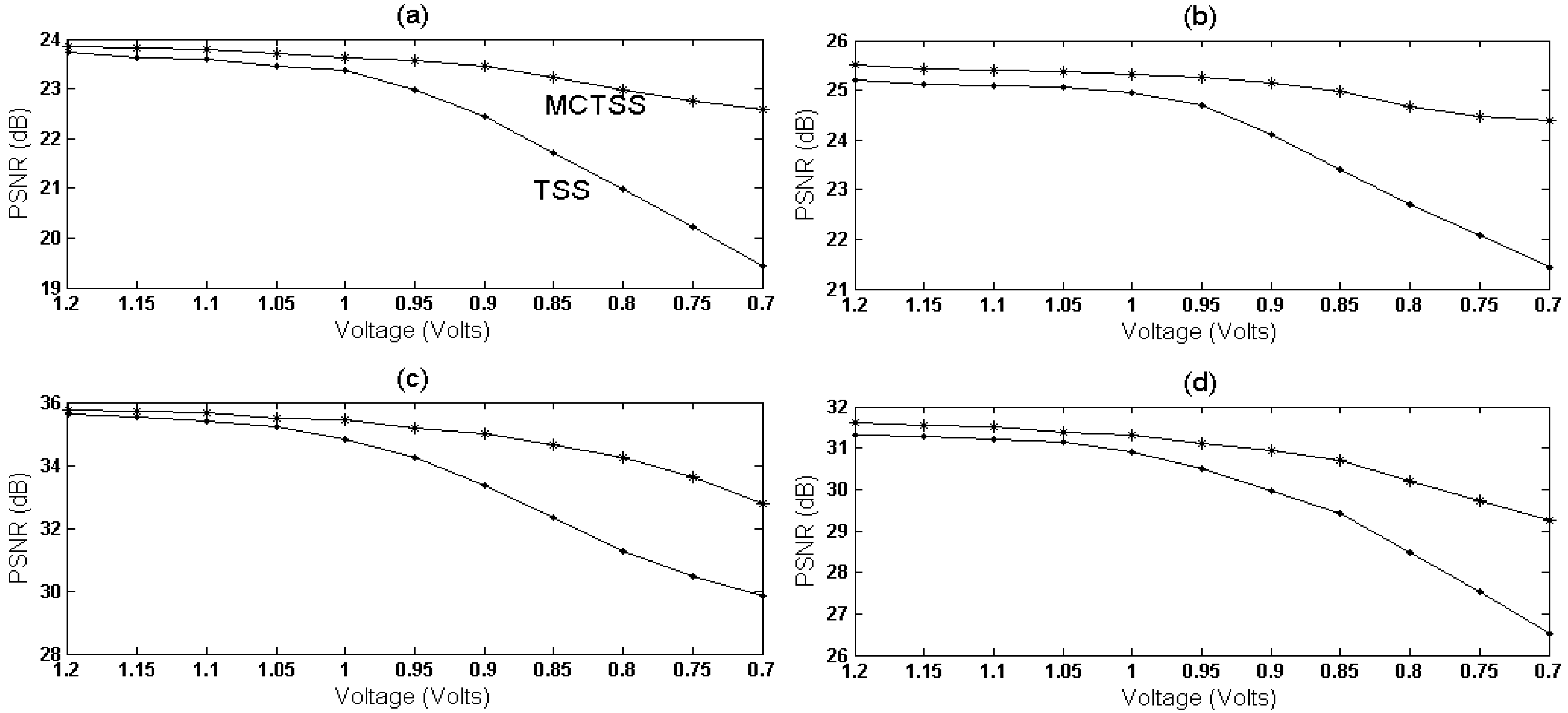
PSNR = 25.02 dB

(b)

Fig. 14: Case 1 (TSS) and Case 2 (MCTSS) at approx. same energy savings for video sequence (a) Susie (55% savings) (b) Flower Garden (70% savings)

Results

- Trend for PSNR values for MCTSS and TSS with voltage scaling over a range of voltage values (1.2 to .7 V)
  - In Fig. 15 (a) Mobile Calendar, (b) Flower Garden (c) Susie (d) Foreman



Results

- Variation of the PSNR over the frames of Video Sequence 'Flower Garden for TSS and MCTSS when energy savings through both is 70% (0.85V)

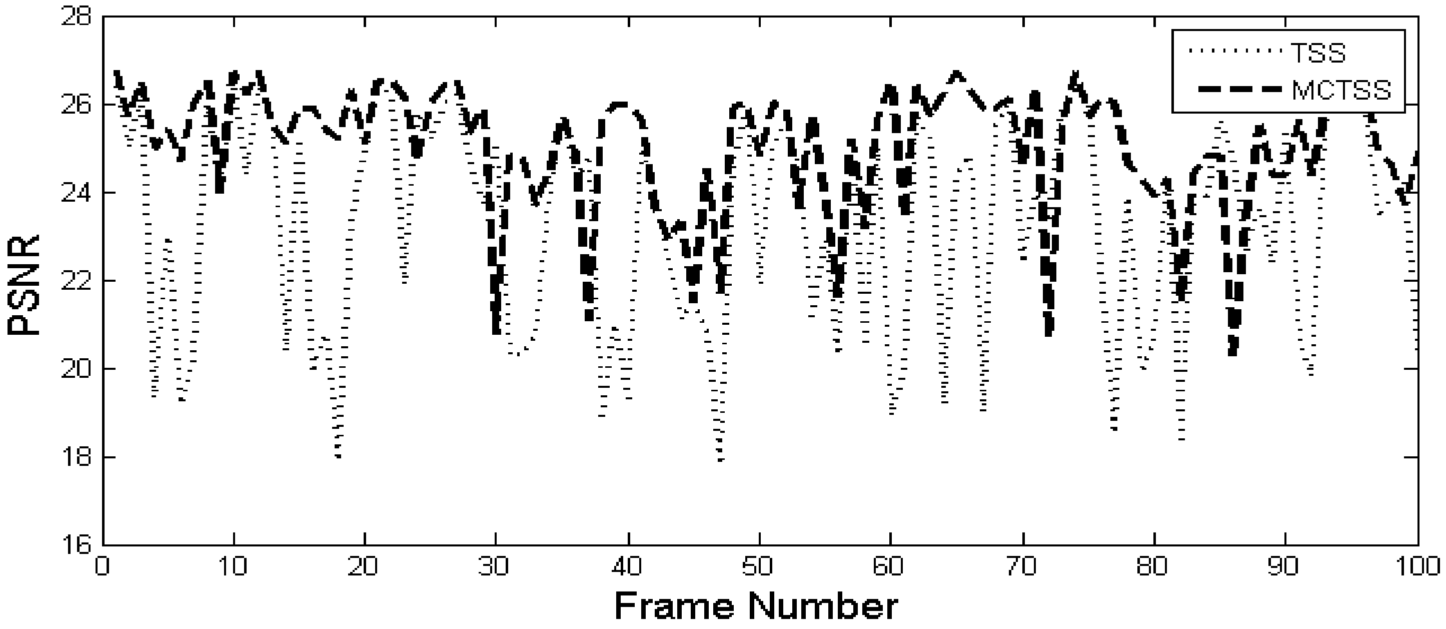


Fig. 15: PSNR variation for TSS and MC-TSS Algorithm over the frames of the video sequence 'Flower Garden' at approx. energy savings of 70%

Conclusion

- We have demonstrated the applicability of error tolerance with both standard prior art (TSS) and our new algorithm (MCTSS)
- The proposed fault tolerant algorithm, MCTSS does much better than the previously established TSS algorithm.
- Increment of 1.8 dB when energy savings through both TSS and MCTSS is same
- Under the limit of 0.5 dB for quality reduction, energy savings increase by about 13% to 15% with MCTSS over that achievable through TSS with overall energy savings as high as 70%
- Algorithmic modifications such as the one proposed in this paper can result in better error resilience while capitalizing on the energy savings and area reductions that the future technology nodes can provide

Thank You!