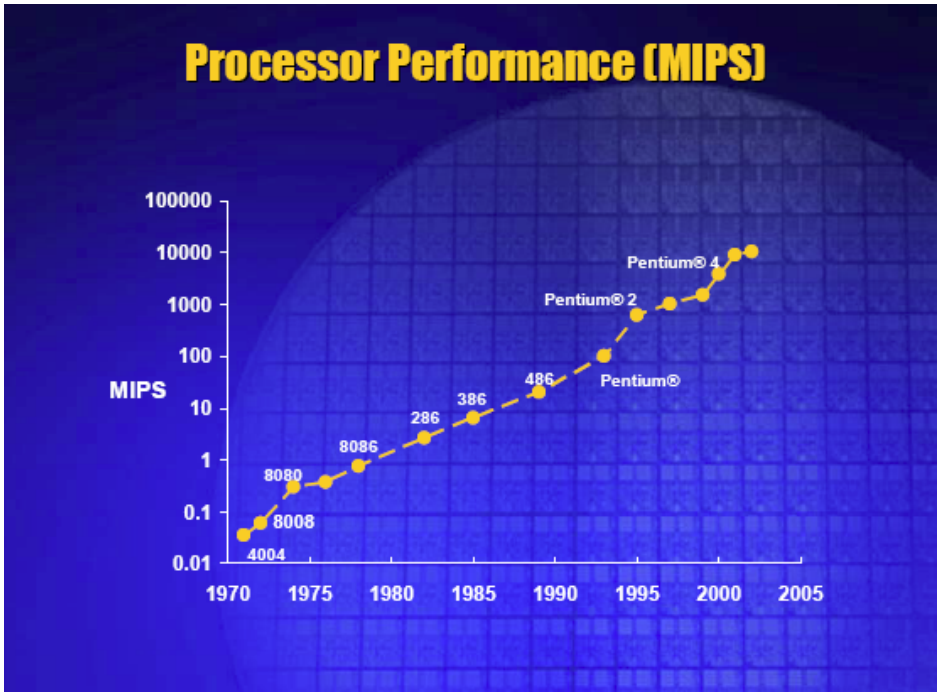
UCIrvine

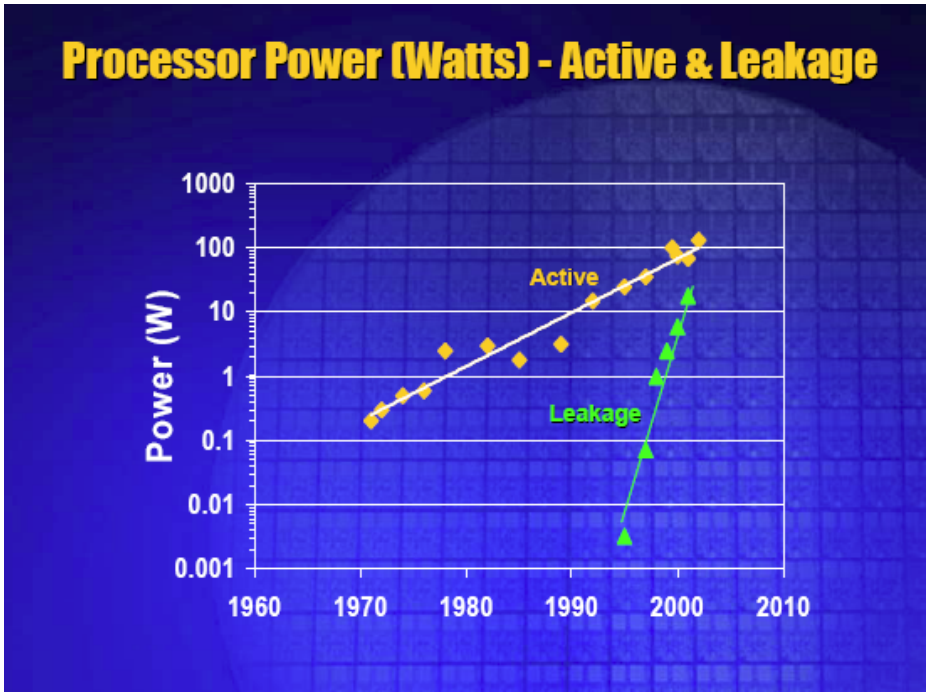
Variability, Design Margining, Low Power and Reliability, How to Bring Them Together?

Fadi J. Kurdahi
Dept of EECS & Center for Embedded Computer Systems
University of California
Irvine, CA USA

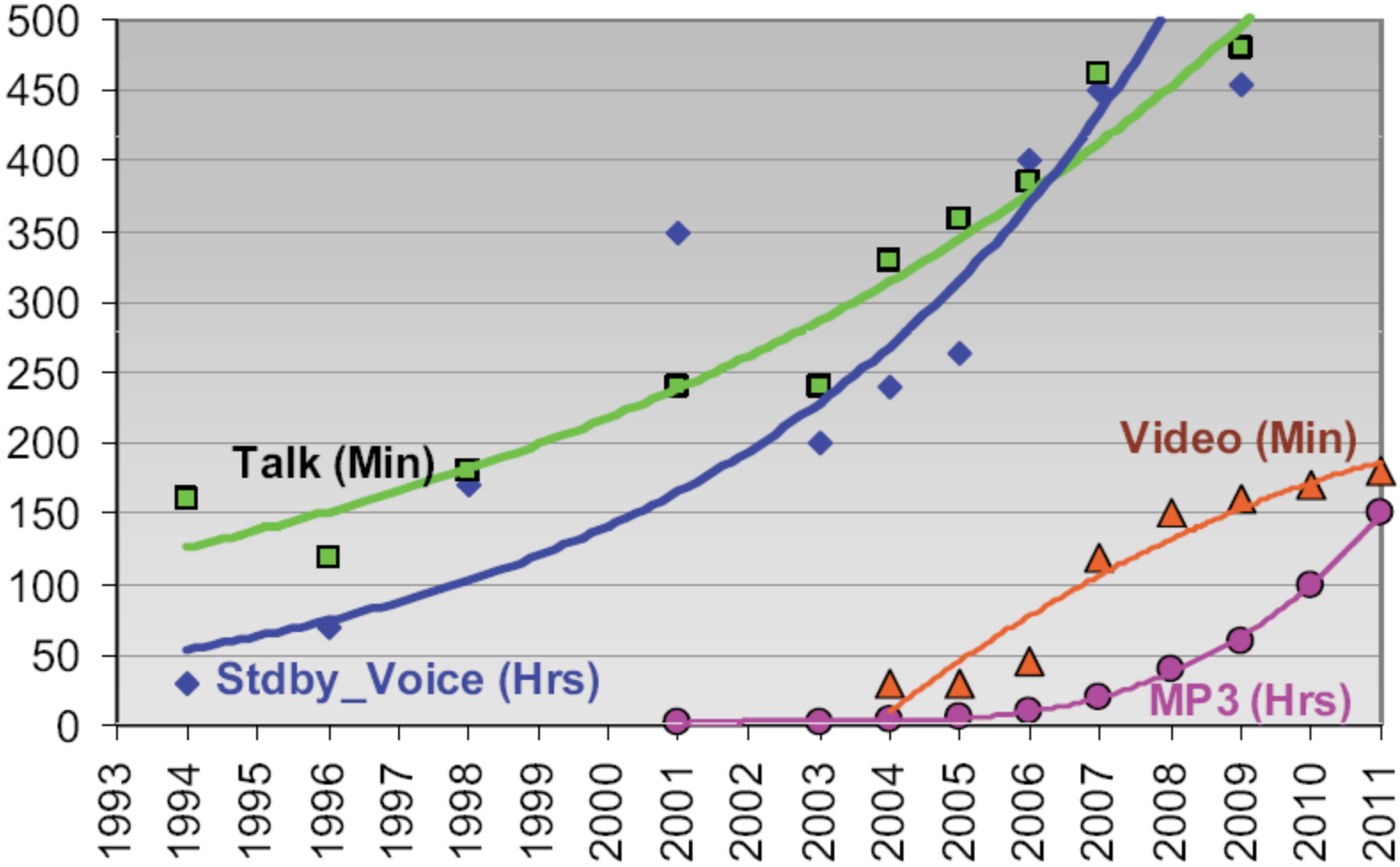
Highly Scaled Geometries Deliver Performance

but at a cost



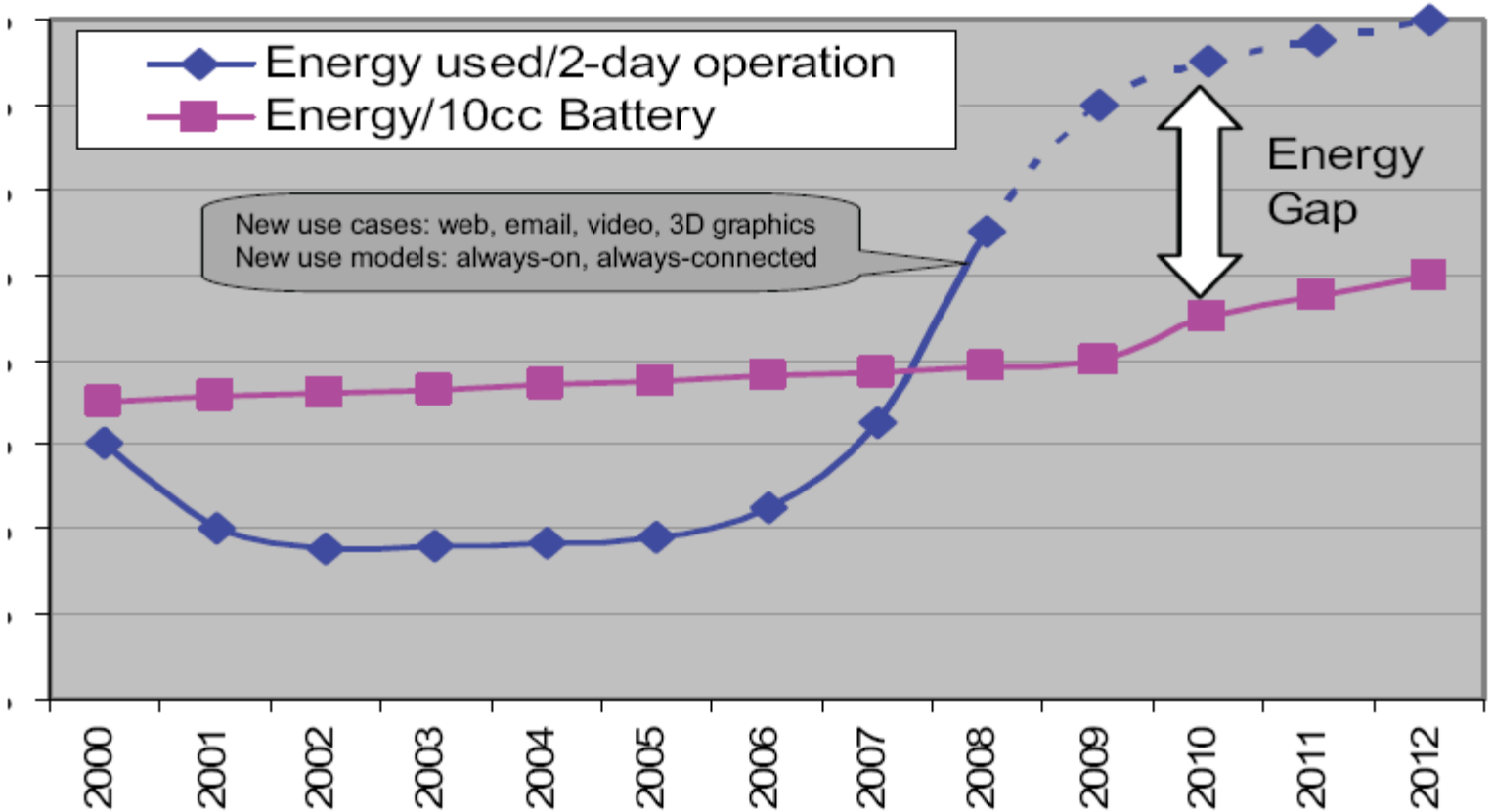


New Application Benchmarks



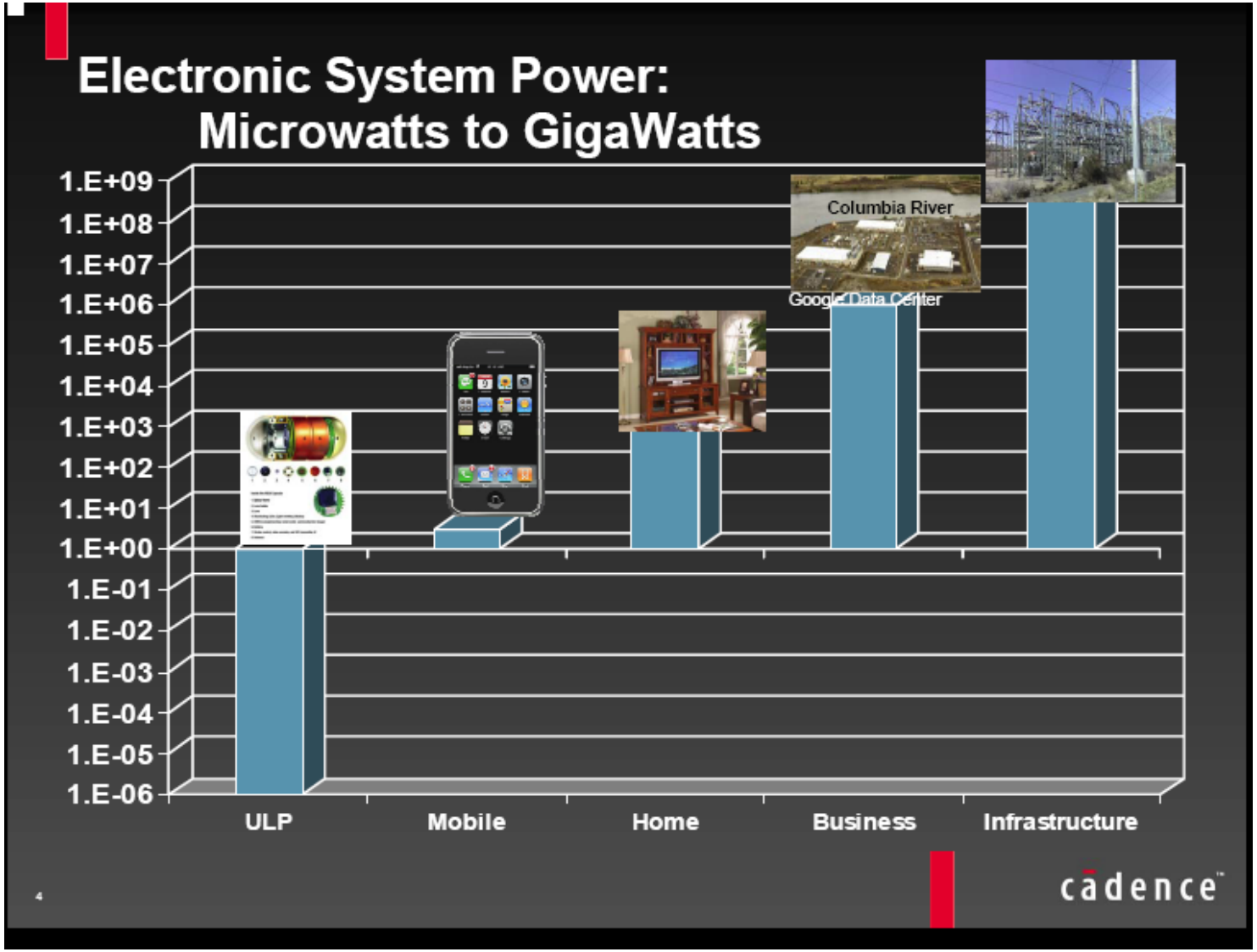
[Ref: Delagi ISSCC 2010]

Are Fueling an “Energy Gap”



[Ref: Delagi ISSCC 2010]

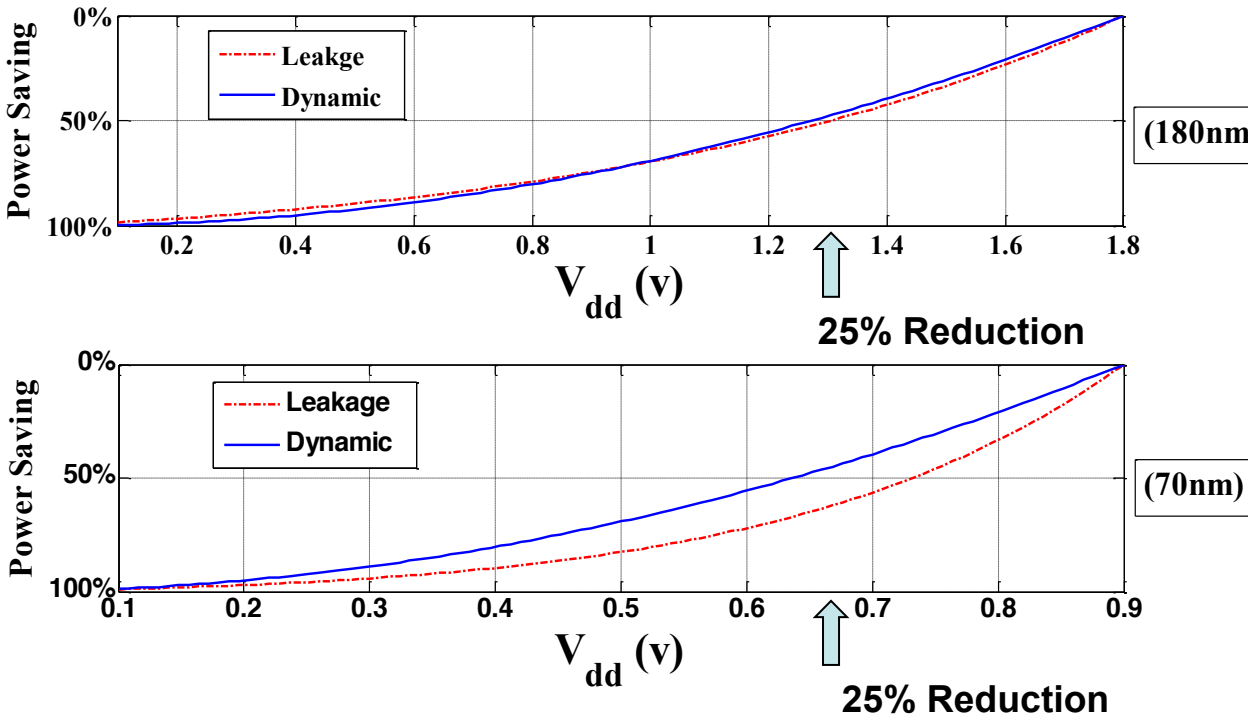
And also across the spectrum of electronics



[Ref: T. Vucurevich SAME 2008]

Reducing power through V_{dd} scaling

- We would like to lower Vdd as much as possible
  - Dynamic power has square relationship with supply voltage (V_{dd})
  - Leakage power has exponential relationship with V_{dd}
- However, lowering V_{dd} too much results in erroneous operations



Almost 50% power saving

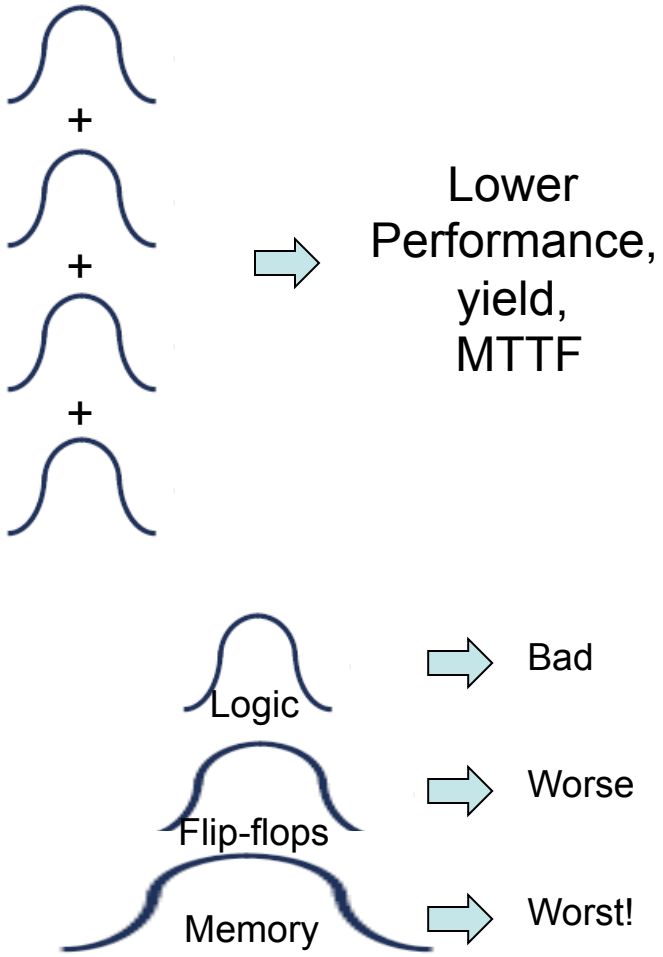
More than 50% power saving

If we were allow errors to occur, we could optimize both yield *and* power consumption

Another Challenge: Variability

- Transistors on a chip are no longer identical
- Smaller features, low supply voltage make circuits more susceptible to noise
- Models used are increasingly inaccurate
- Temperature, etc…

Becoming more acute with advanced fabrication technologies

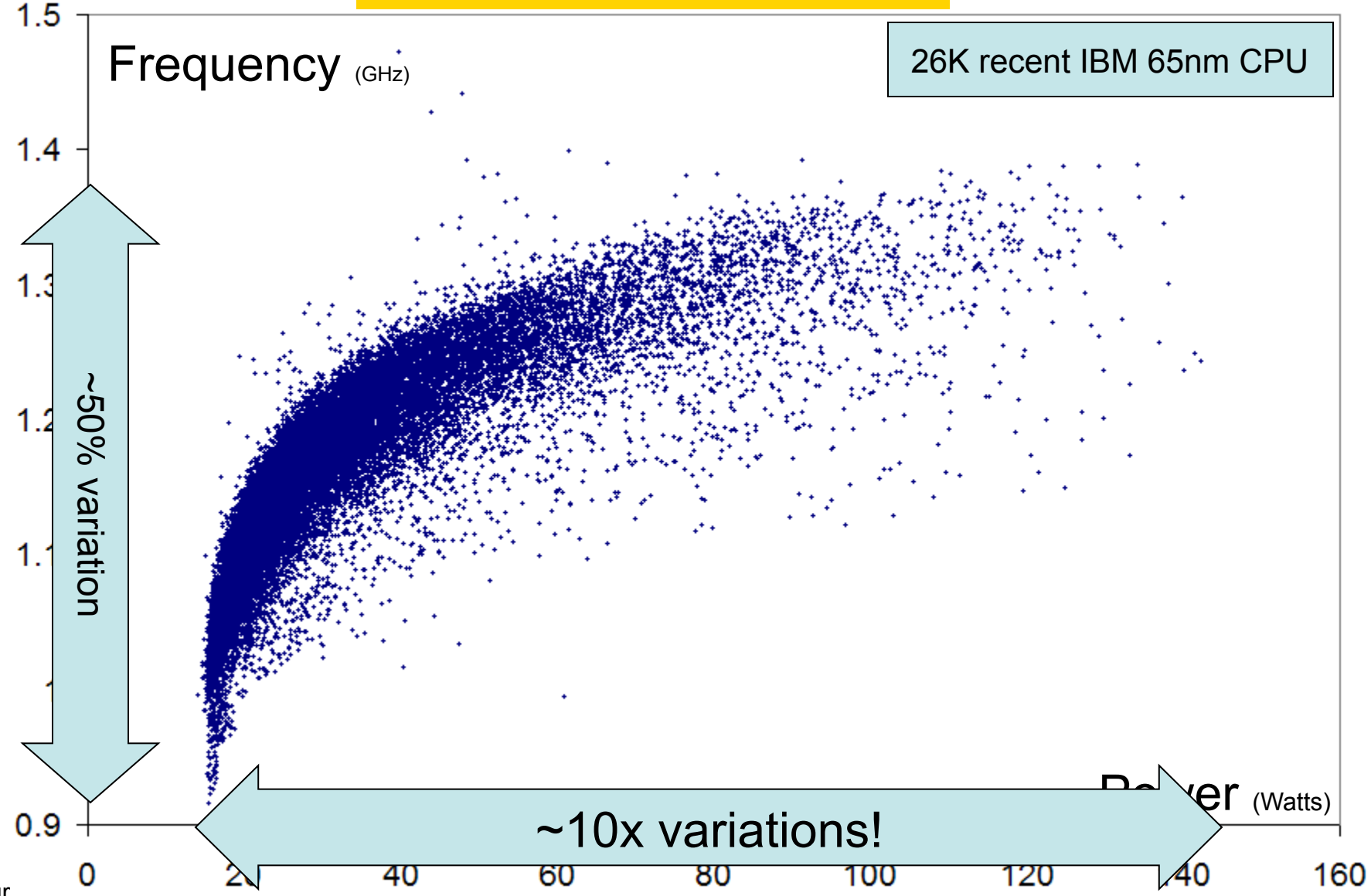


Straightforward solution:

Increase the supply voltage (V_{dd})

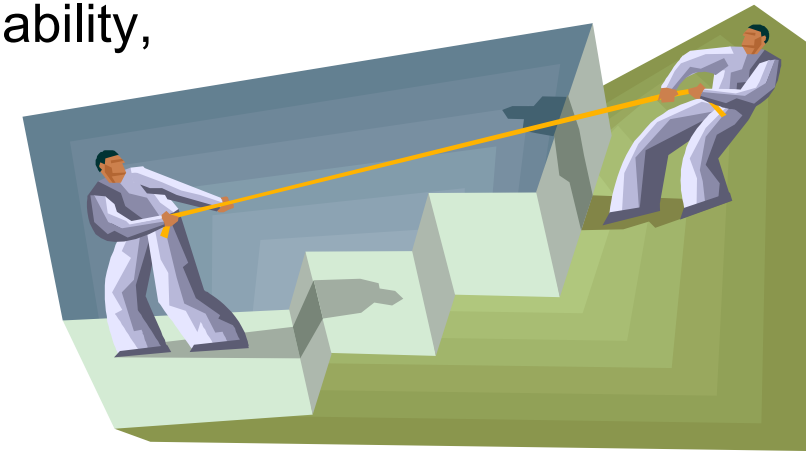
- More noise immunity
- Faster

Source: SR Nassif – IBM



So now we have a conflict!

- ***Increase*** V_{dd} to compensate for variability, reliability, and performance
- ***Decrease*** V_{dd} to reduce power

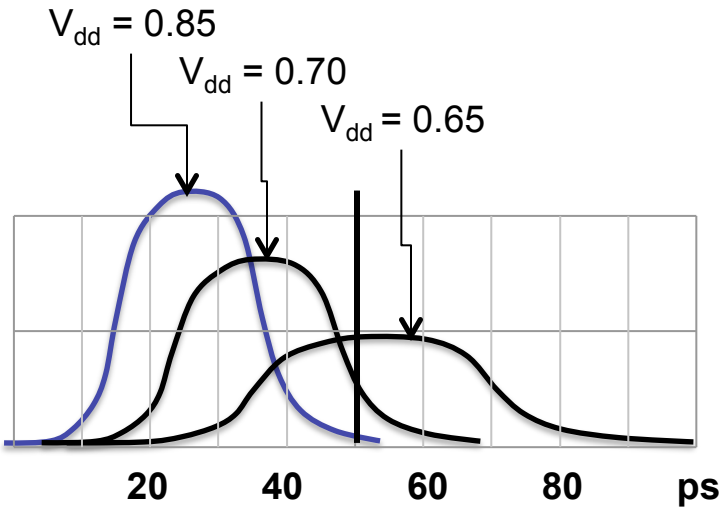


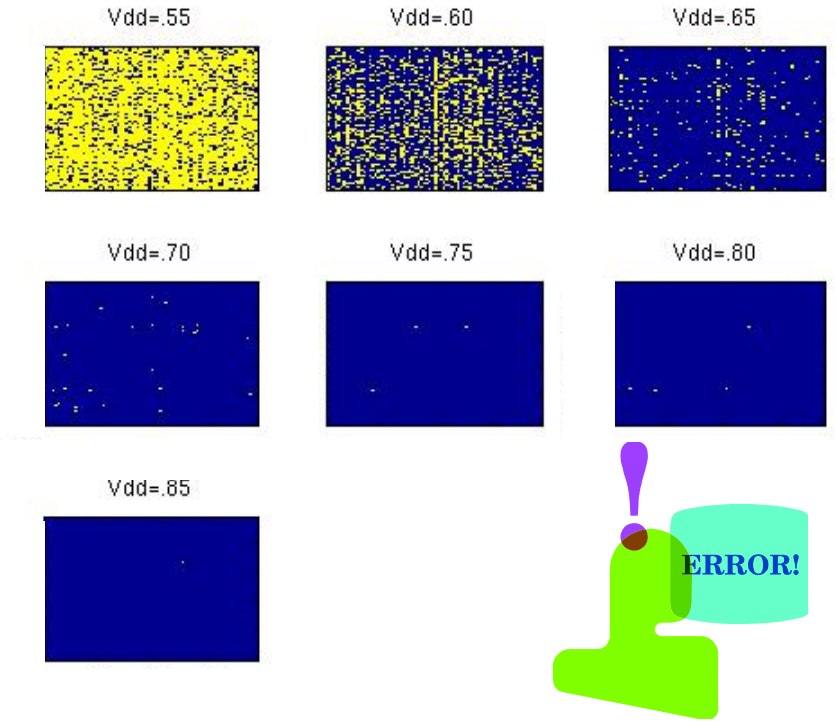
What happens if I try to have my cake and eat it too?

Conventional wisdom: when you reduce V_{dd}, you should run slower.
What happens if I reduce V_{dd} and keep running at the same frequency?

Take a memory array, for example:

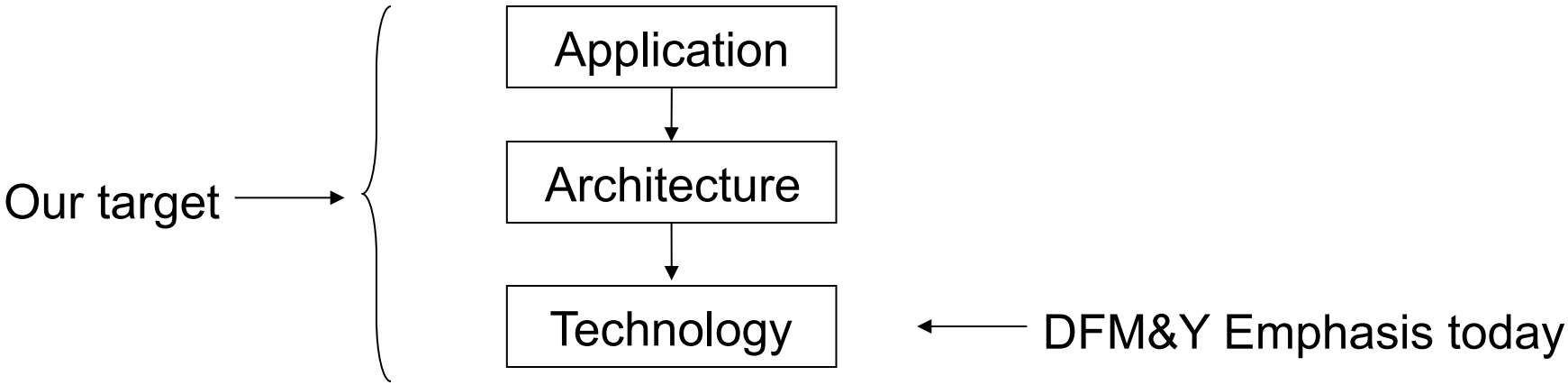
- Variability -> variable read, write delay across memory cells
- V_{dd} reduced -> increased probability of access failure/cell -> more cells failing



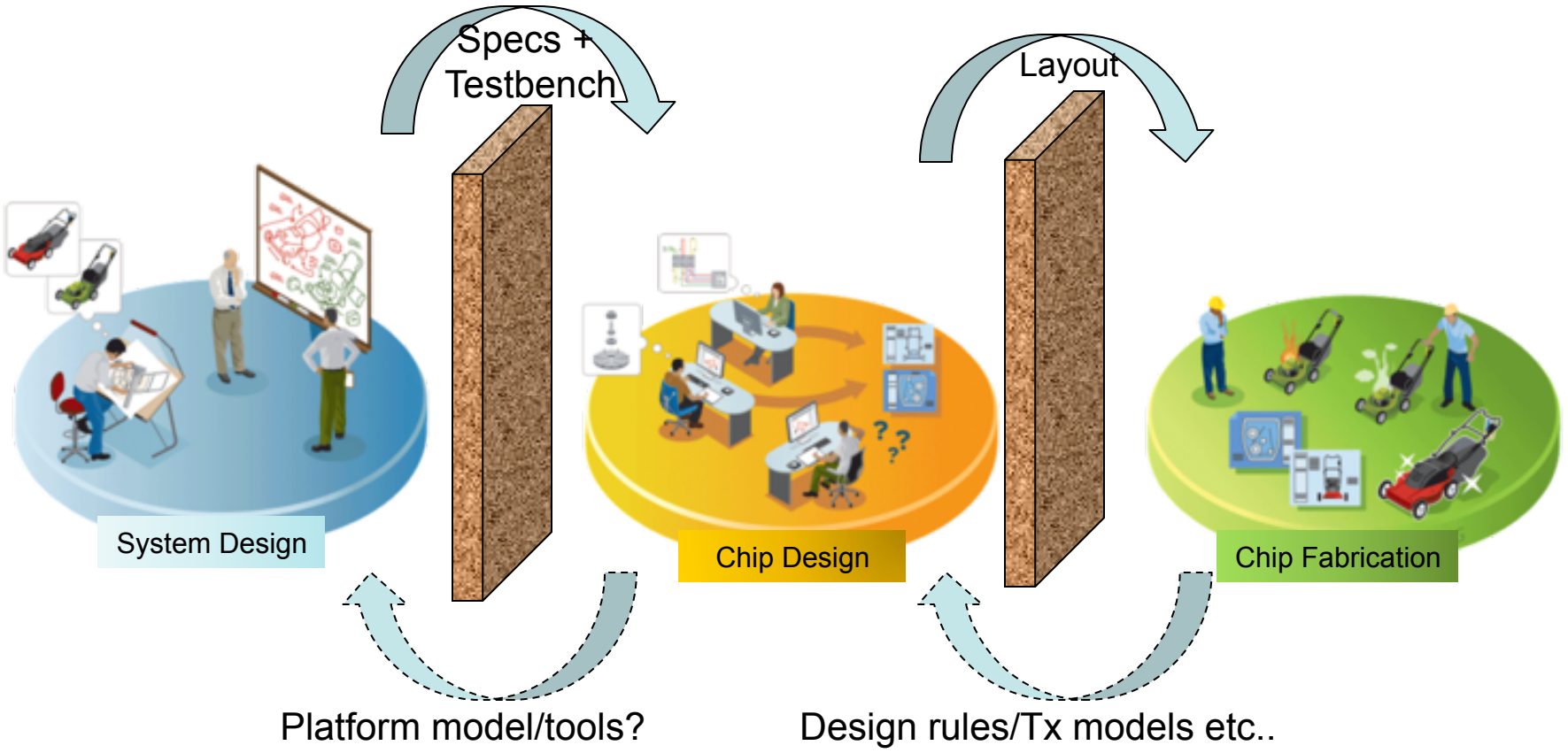


Concept of Yield?

- We are still insisting on error free status of accepted chips
- With decreasing device geometries this is becoming a hard requirement to keep while maintaining sufficiently high yield to offset NRE and RE costs
- “To maintain 100% error free chips is expensive, time consuming and soon will become impractical” (ITRS Roadmap)
- ⇒The earlier yield is considered in the design process, the more benefits can be reaped.



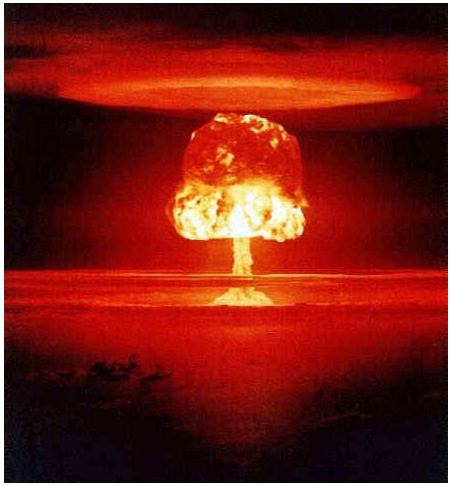
Challenge: Today's design process– The Walls



- Specs in Matlab or C
- Chip design must match specs ***bit-exactly***

How bad are errors?

- For system designers: depends:
  - Systems are designed for worst case conditions
    - …. Which occur quite infrequently
  - Many systems are designed to *inherently* tolerate faults, but up to a limit
    - Wireless
    - Multimedia
  - Yet the realizations of those systems must be 100% correct wrt specs!
- For chip designers: catastrophic!



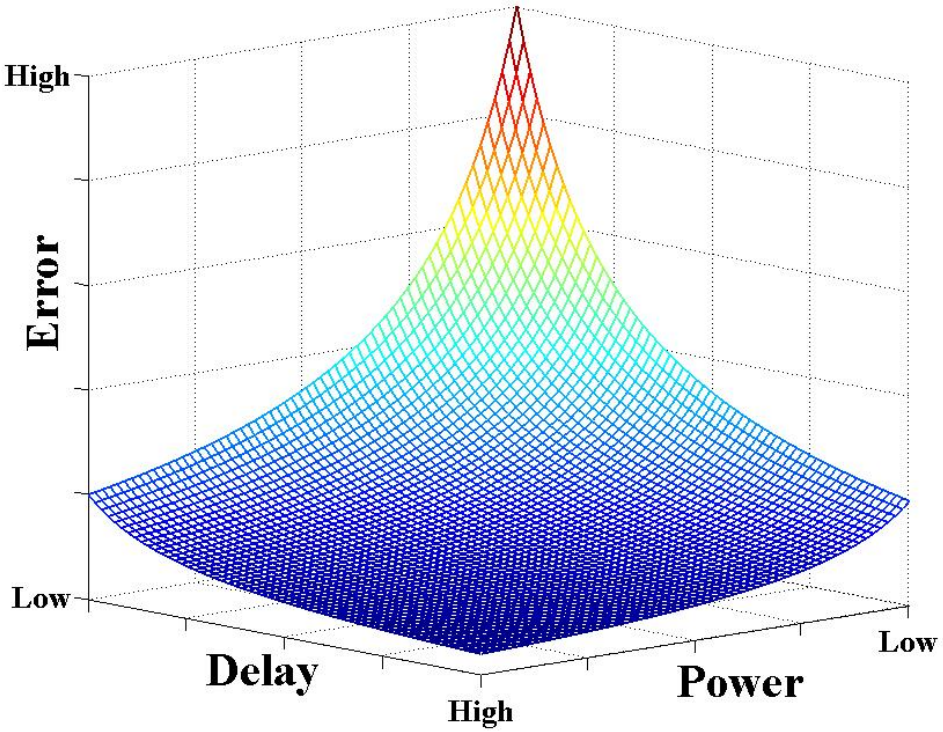
Idea

Co-design system and hardware to relax the 100% correctness requirement on the underlying hardware.

- If needed, fix the errors at the system/application level
- Improved power consumption by allowing aggressive voltage and frequency scaling
- Improved yield by relaxing the specifications defining a working chip.

Implications on the design space

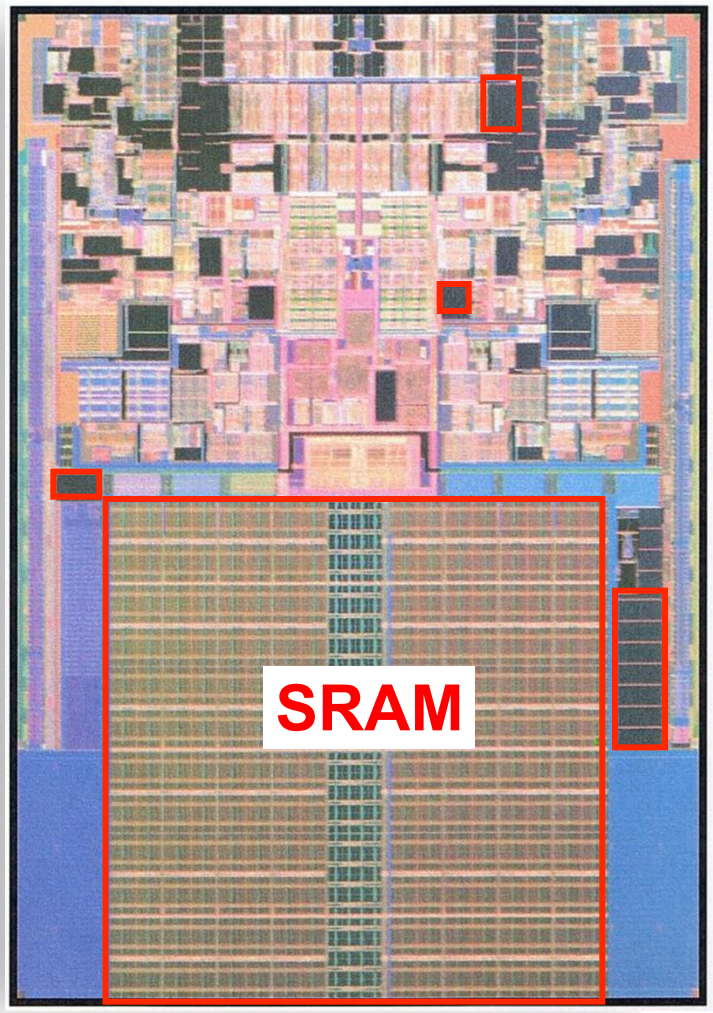
- Proposed paradigm allows a new dimension for system optimization by encapsulating fault tolerance
- Traditional design space trades off power for performance



Existing work

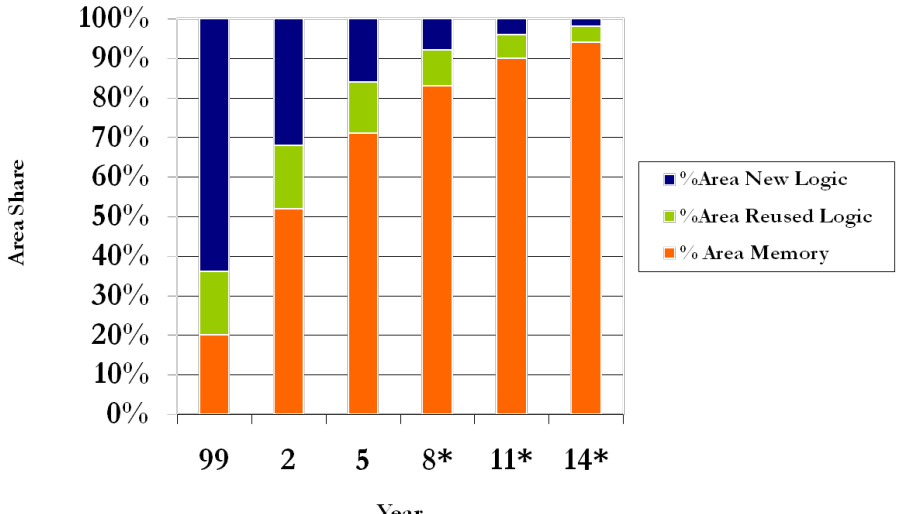
- Application-Aware Adaptation (Software level)[Mitra2010]
  - Balancing scheduling of operations to maintain constant current drain
  - Selective execution of threads
- Power-Aware Design (Hardware level)
  - Dynamic voltage scaling
  - Dynamic frequency scaling
  - Sub-threshold operation
- Large body of work on RAM BIST/BISR [D&T2004]
- Breuer et. al. [D&T2003]
- Shanbhag et. al. [TVLSI2003]
- Roy et. Al [TVLSI2005]
- Timing-aggressive, error tolerant NoC design [Benini, DeMicheli et a. 2007]
- ERSA [Mitra 2010]

Observation: SoC Becoming Memory Dominated



Intel Penryn™

(Picture courtesy of Intel)



- **On chip SRAM contains 50-90% of total transistor count**
  - Xeon: 48M/110M
  - Itanium 2: 144M/220M
- **SRAM is a major source of chip power dissipation**
  - Dominant in ultra-low power
  - Substantial fraction in others

Focus on memories
Logic later

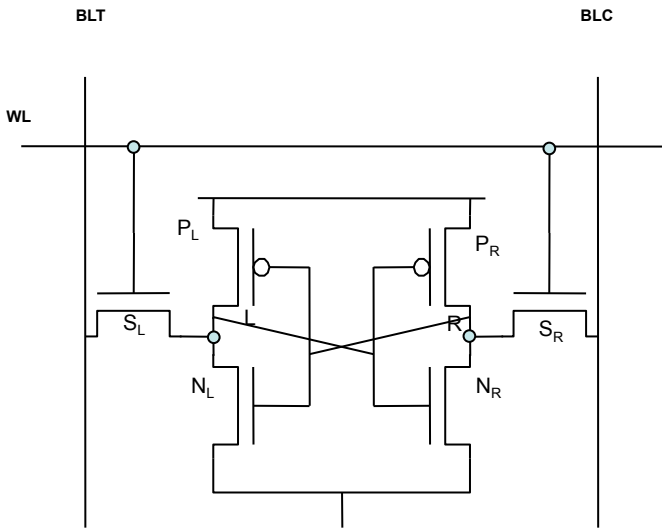
Critical Questions

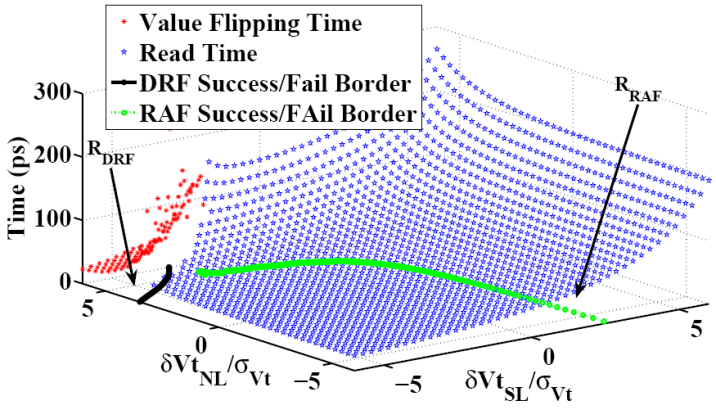
1. How do embedded memories behave under aggressive voltage scaling?
2. How to compensate for the memory's errors at the system level?
3. How much is the expected overall power saving?

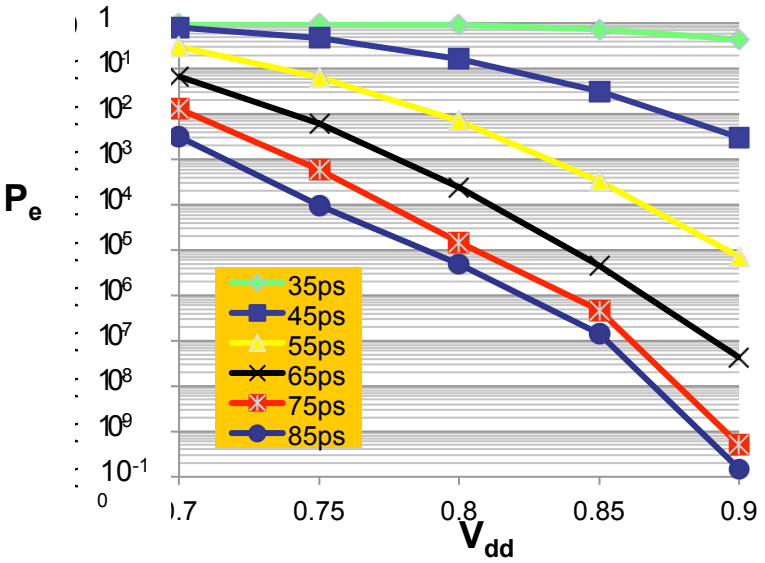
Nature of Memory Defects

- **Fixed**
  - Manufacturing errors
  - Predominant in above 100nm technologies
  - Redundancy solutions
- **Transient**
  - Alpha particles
- **Operating condition**
  - Voltage, frequency, temperature
  - Predominant in sub 100nm technologies
  - Defects are due to :
    - Gate Length Variation (GLV)
    - Random Dopant Fluctuation (RDF)
  - Manifest themselves at the circuit level as inter-die variation in Vt

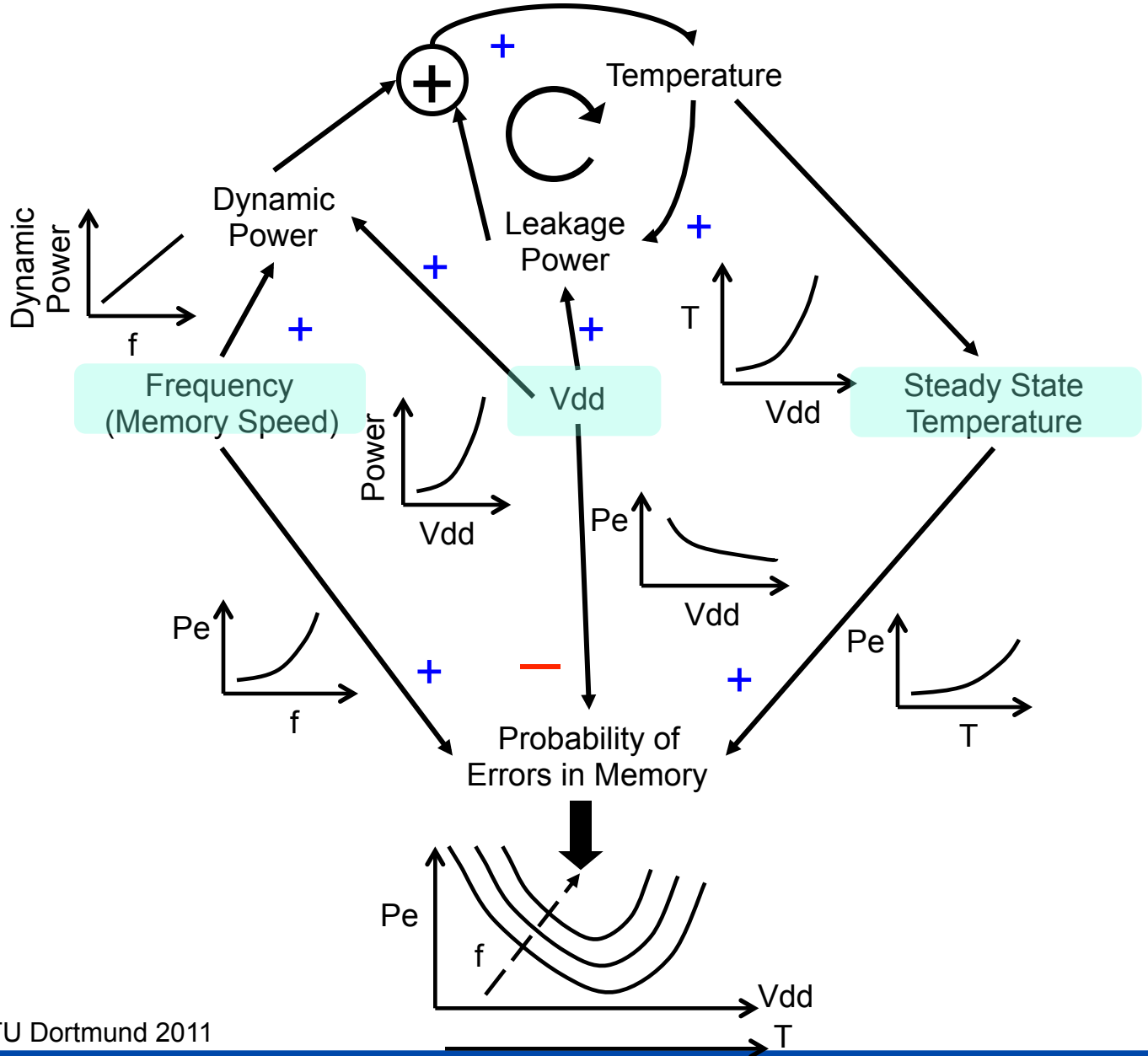
Understanding memory behavior



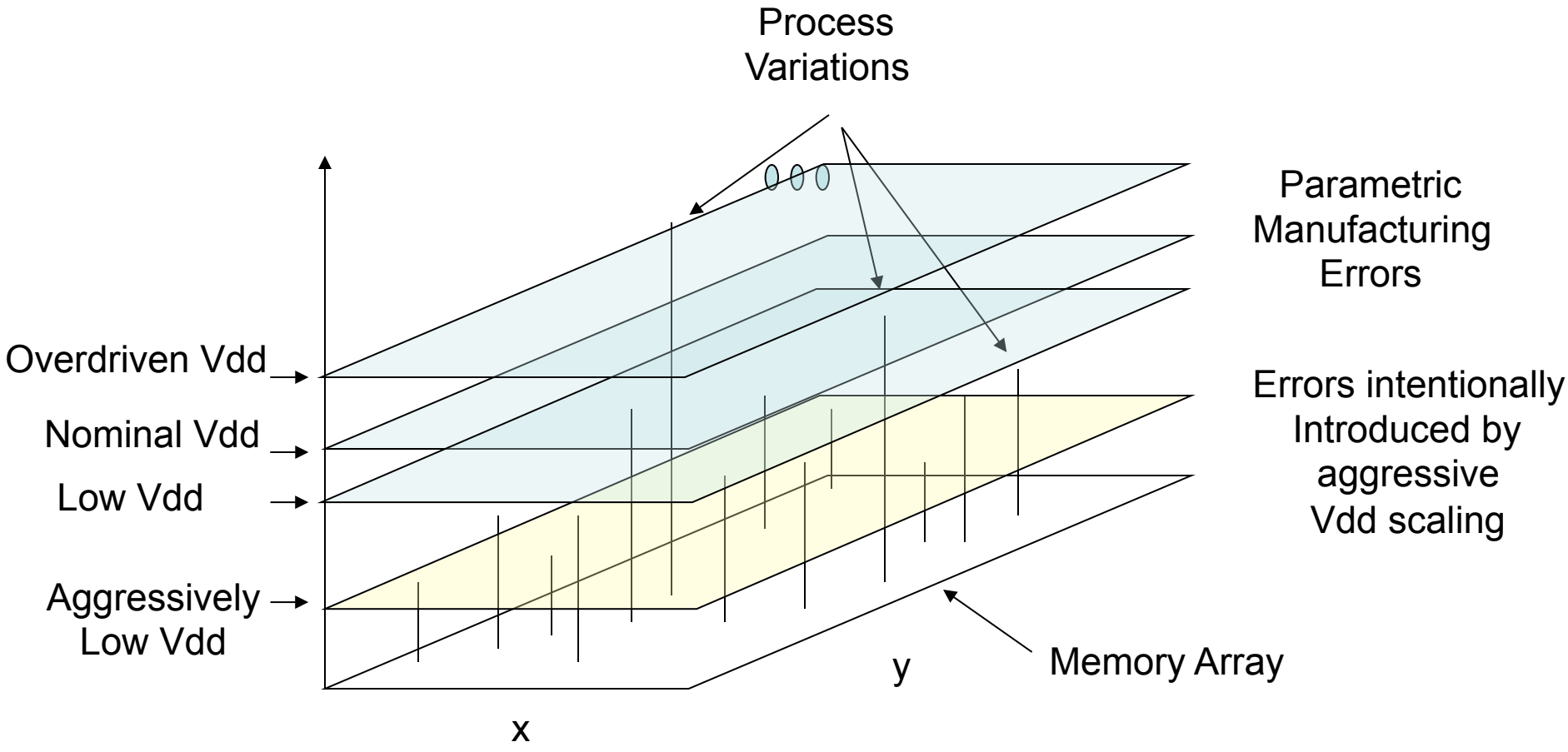




Interaction **between V, T and Errors**



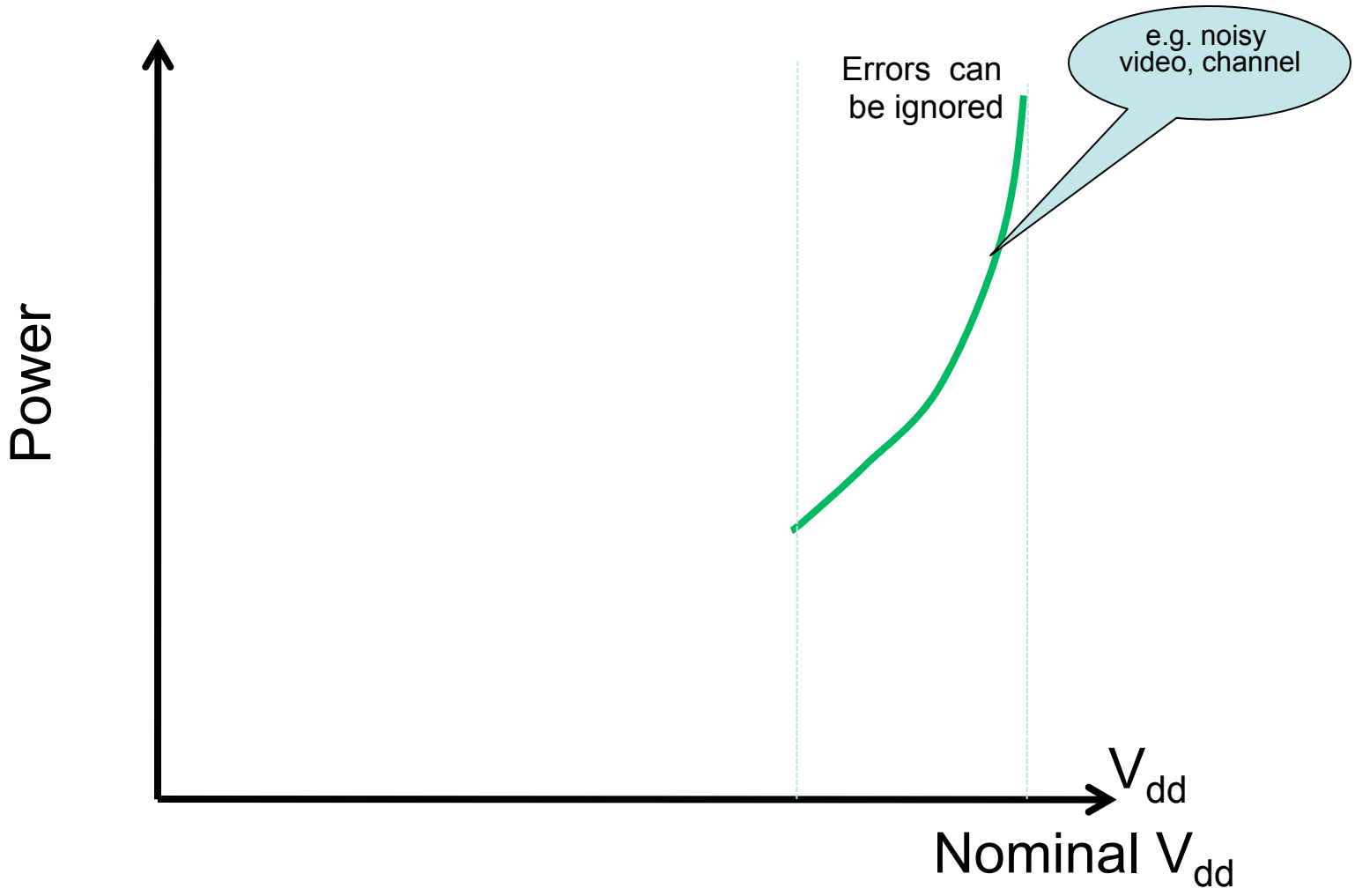
Visualizing the global memory behavior



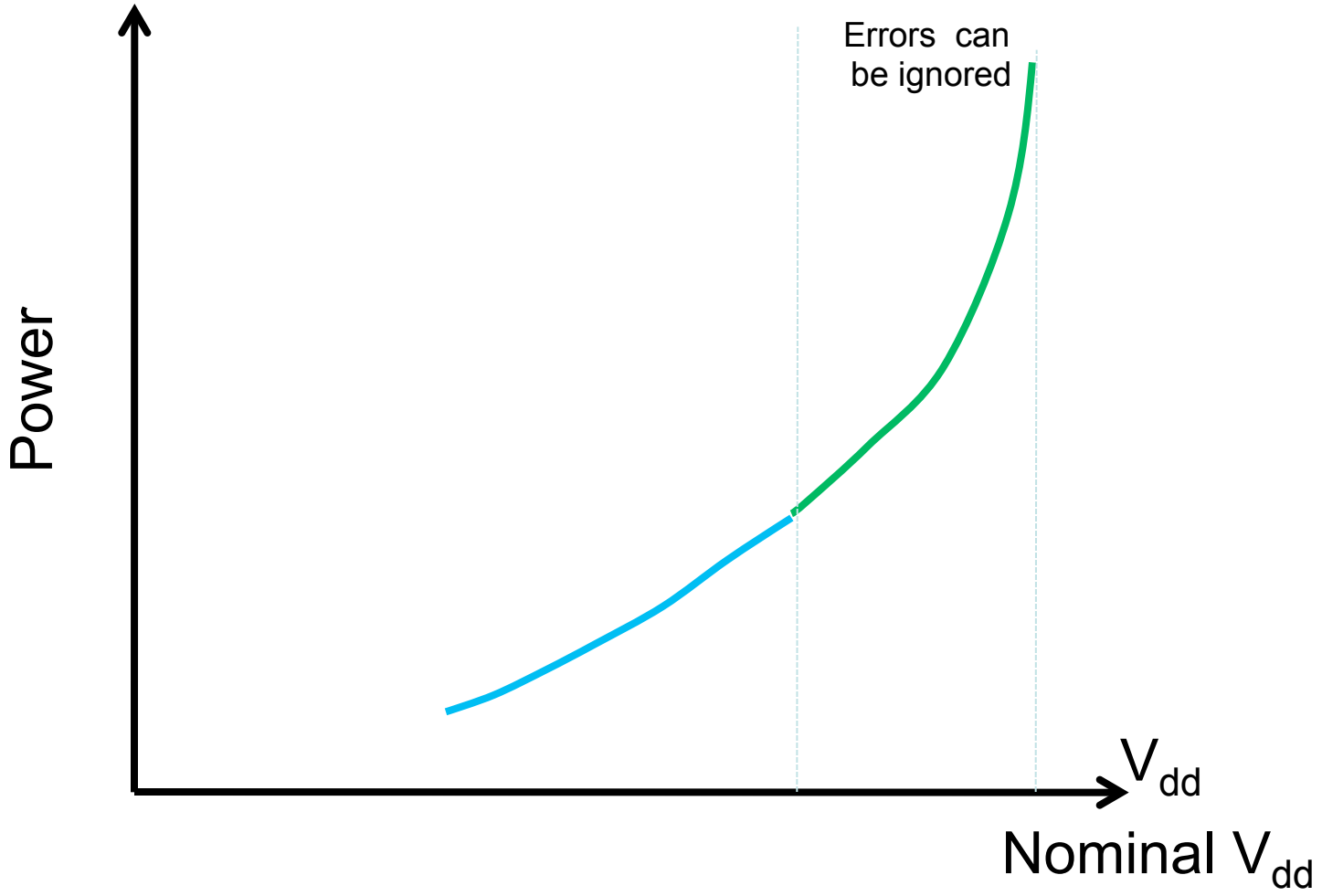
Critical Questions

1. How do embedded memories behave under aggressive voltage scaling?
2. How to compensate for the memory's errors at the system level?
3. How much is the expected overall power saving?

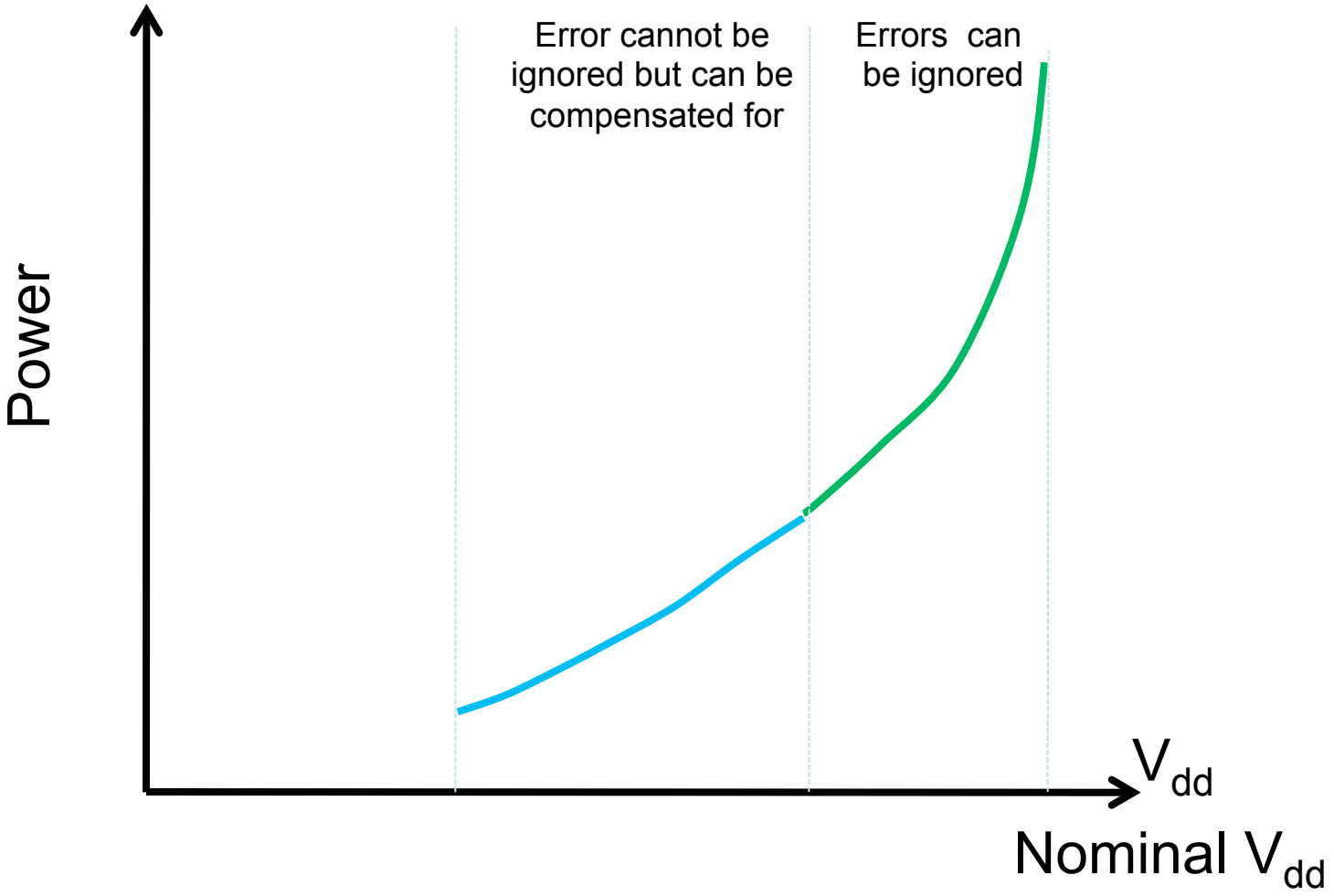
Dealing with Errors



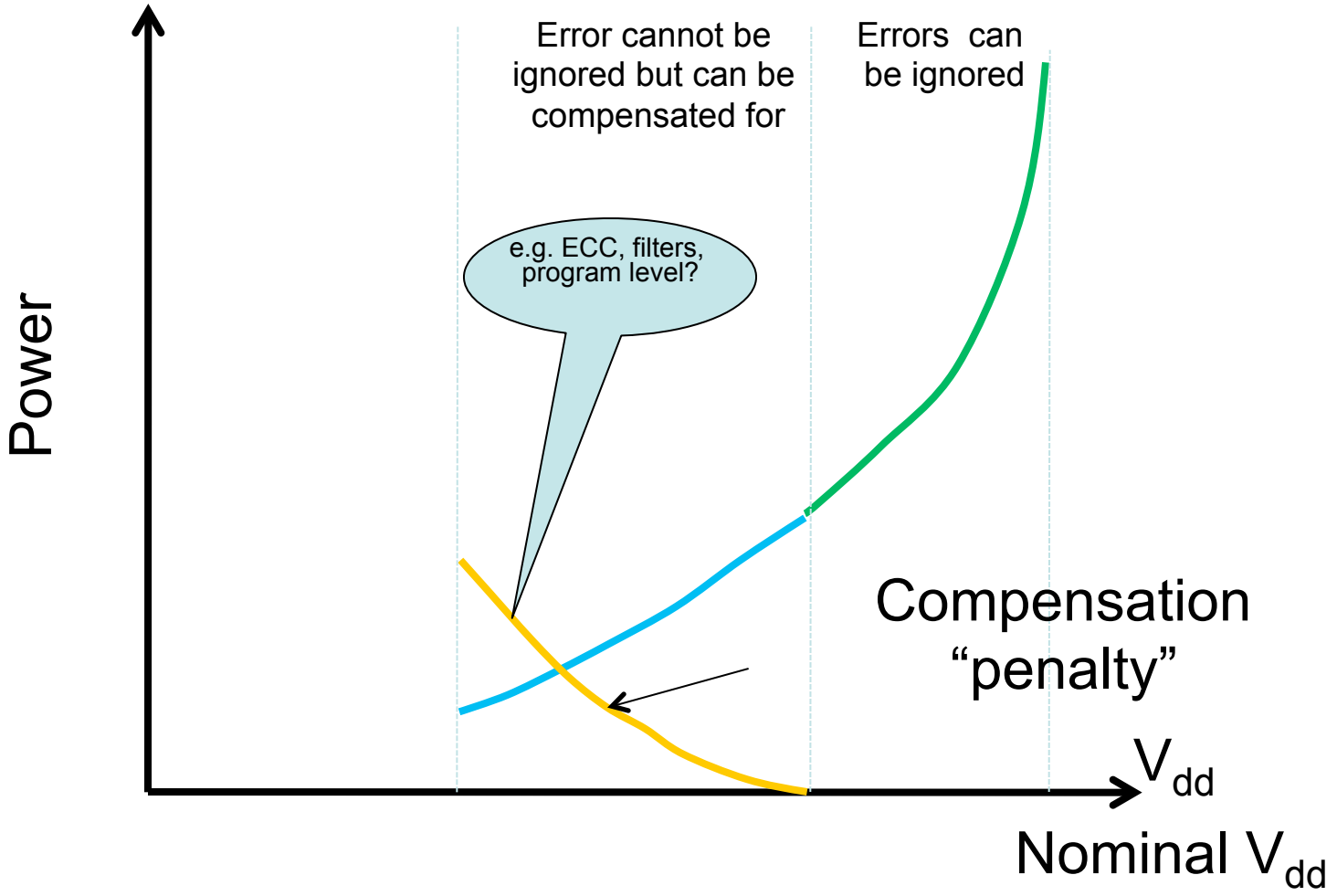
Dealing with Errors



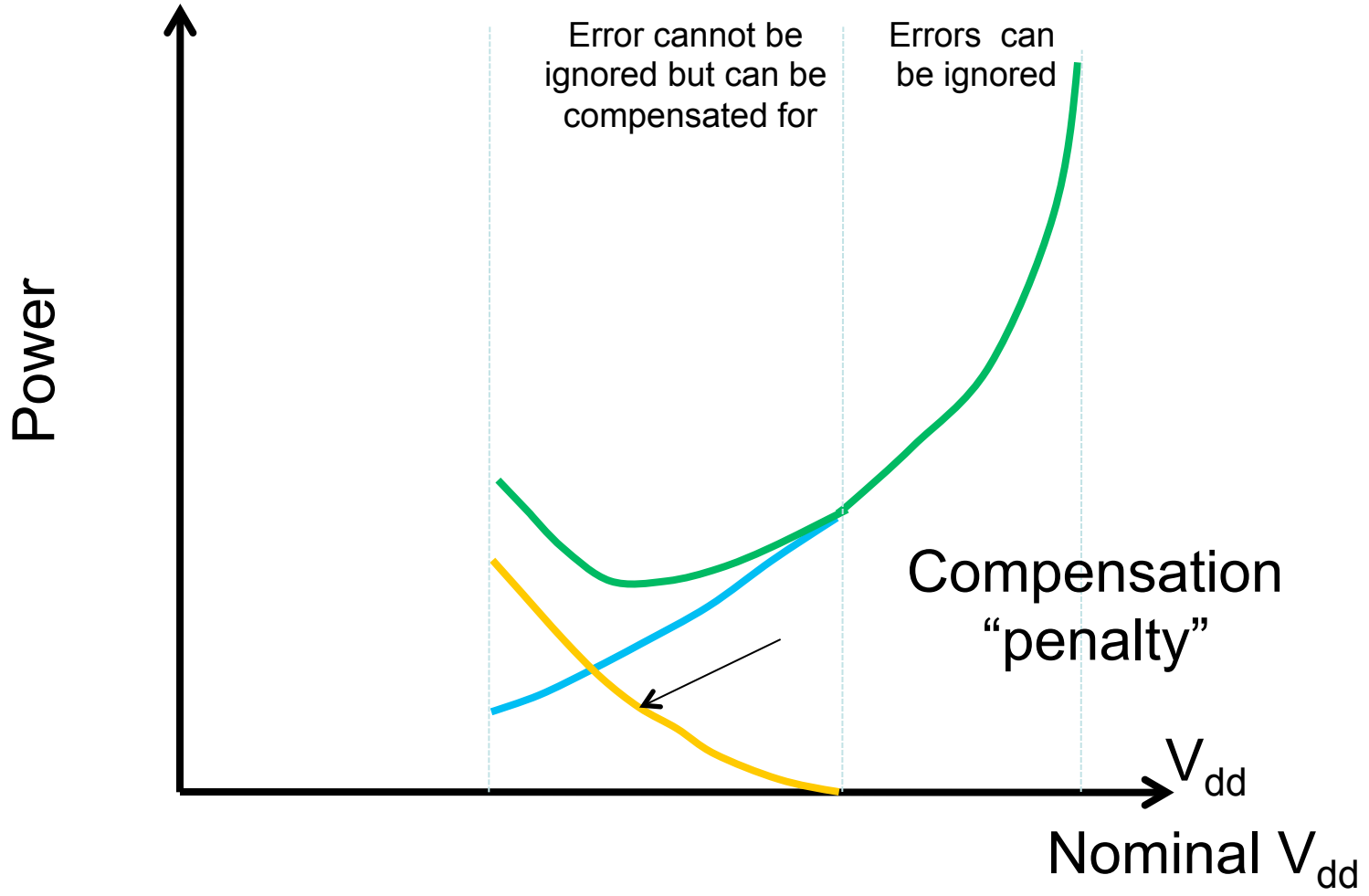
Dealing with Errors



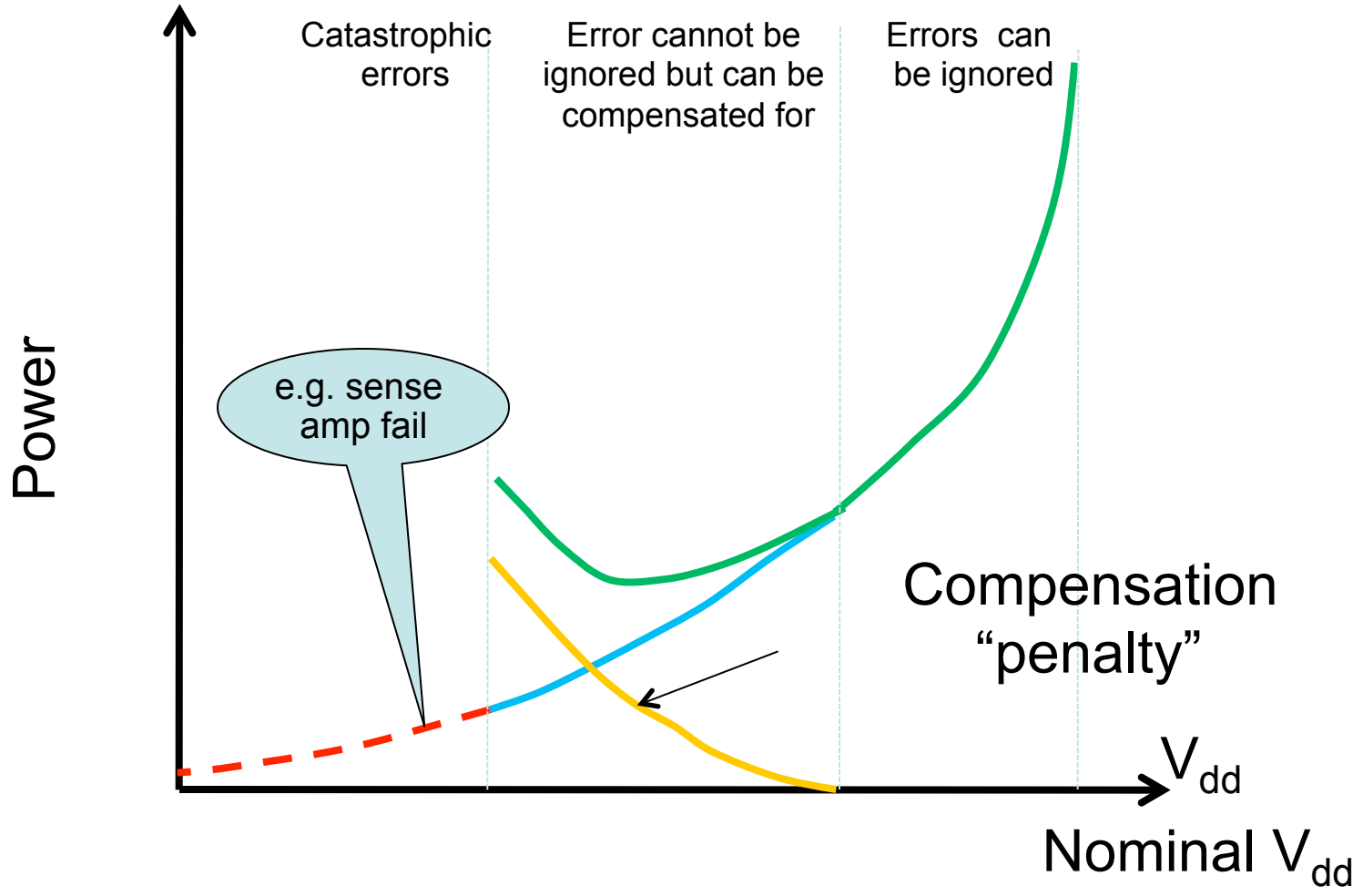
Dealing with errors



Dealing with Errors



Dealing with Errors

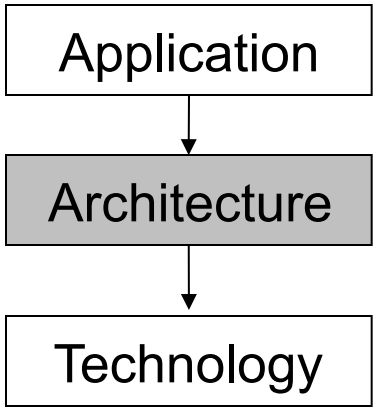


Critical Questions

1. How do embedded memories behave under aggressive voltage scaling?
2. How to compensate for the memory's errors at the system level?
3. How much is the expected overall power saving?

Case Studies:

- Processor Caches
- H.264 Decoder
- Wireless Link
- Cross Layer

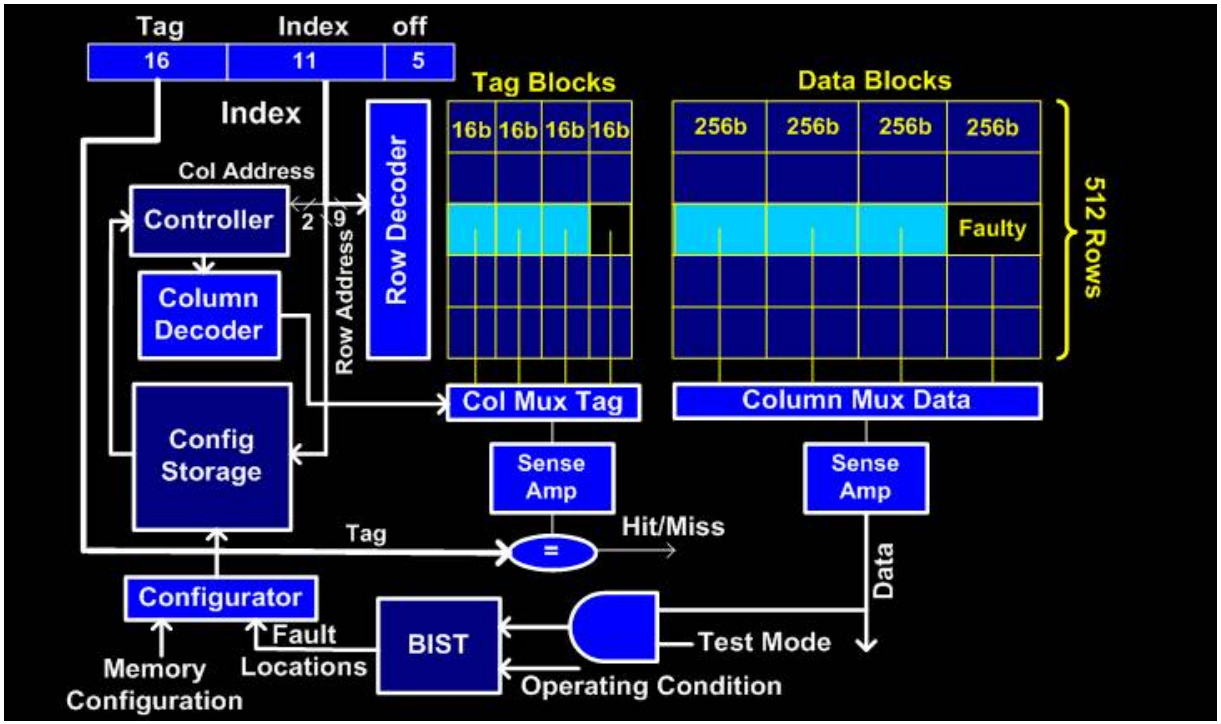


PROCESSOR MEMORIES

State of the Art Cache Architecture

- **Non voltage scalable**
  - Due to increase in access time.
  - Due to increase in defect rate.
- **Defects are diagnosed in manufacture time**
  - Only for nominal voltage
  - Redundant rows used to replace defective ones.
- **Pipelined (wave, latch), non-pipelined (mostly embedded designs)**
- **Dynamically generated errors**
  - BIST (startup, interval)
  - DTS

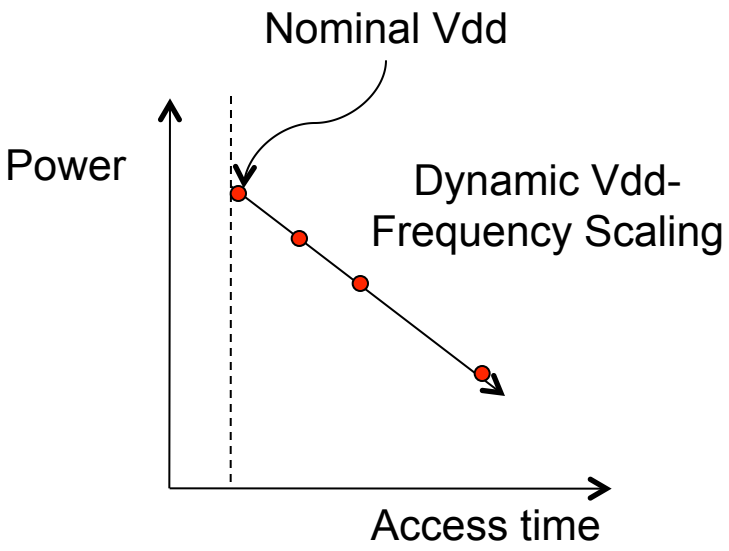
Fault-Tolerant Cache Architecture [Roy2006]



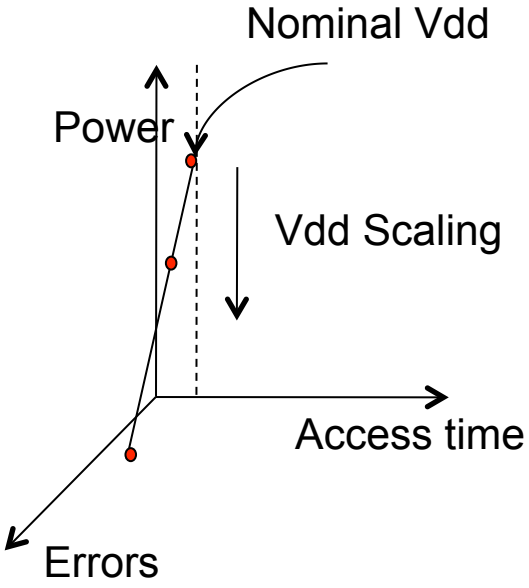
- **BIST detects the faulty blocks**
- **Config Storage stores the fault information**
- **Idea is to resize the cache to avoid faulty blocks during regular operation**
- **Works well for yield enhancement at nominal Vdd**

Source: Kaushik Roy

What if I want to reduce Cache power?



What is done today



Can we do this?

What happens when we start scaling Vdd of a Cache?

- Disable faulty blocks
- Assume BIST & Defect map are error-free

Voltage	Miss Rate
0.9	3.92
0.85	4.10
0.8	5.9
0.75	14.6
0.7	59.5

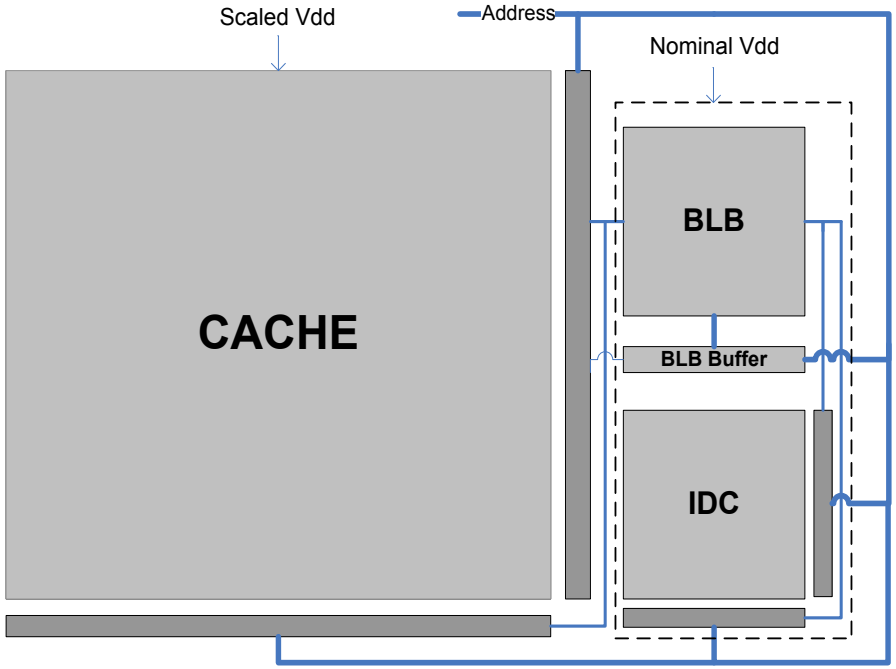
Can we improve miss rate and still lower Vdd?

A fault tolerant L1 cache architecture

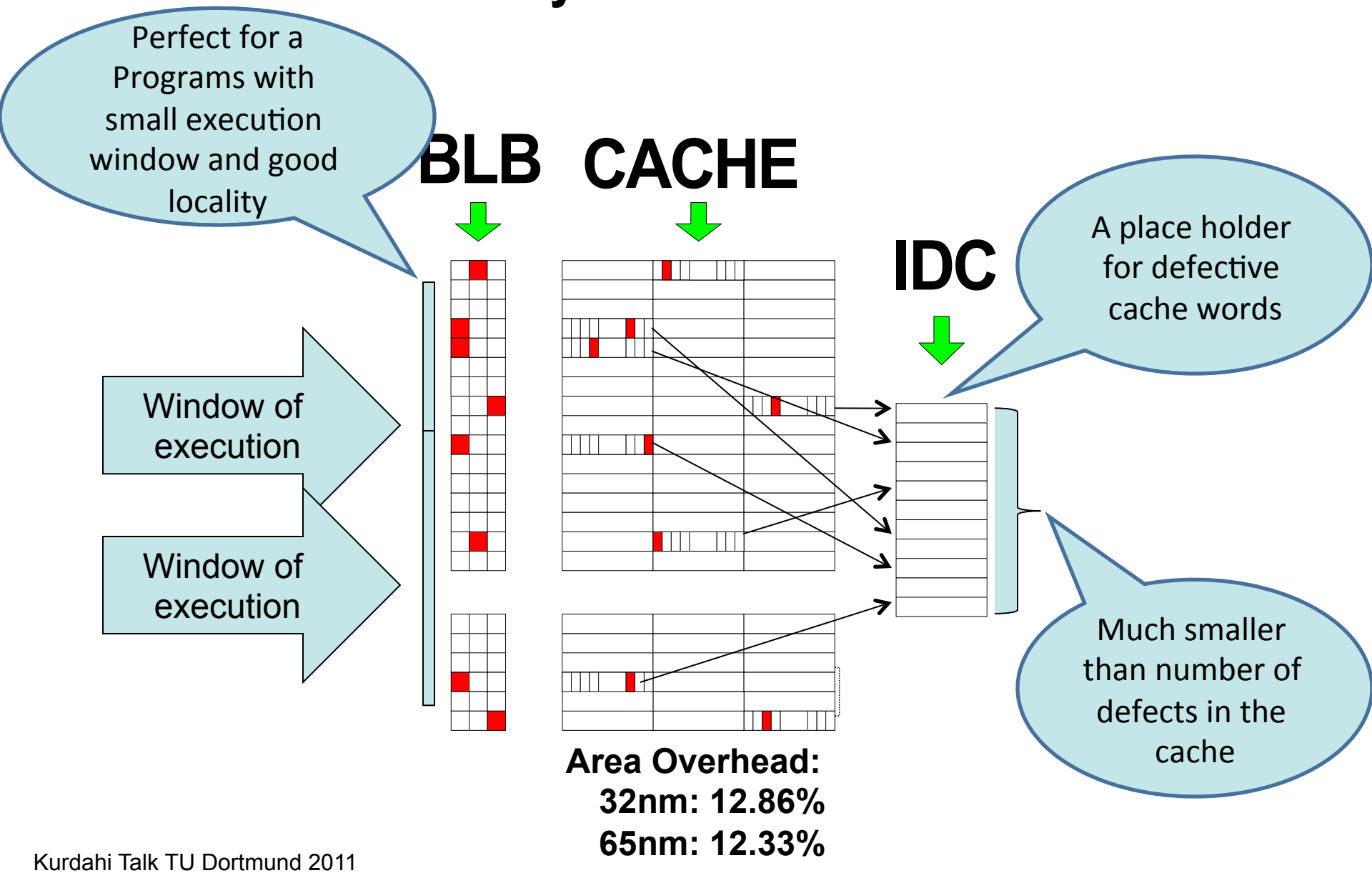
Bit Lock Block (BLB)
- Off chip defect map always operating at nominal voltage.
- Equipped with a buffer acting as a small cache for BLB

Inquisitive Defect Cache (IDC)
- Small direct or associative cache, acting as a place holder for defective words in window of execution

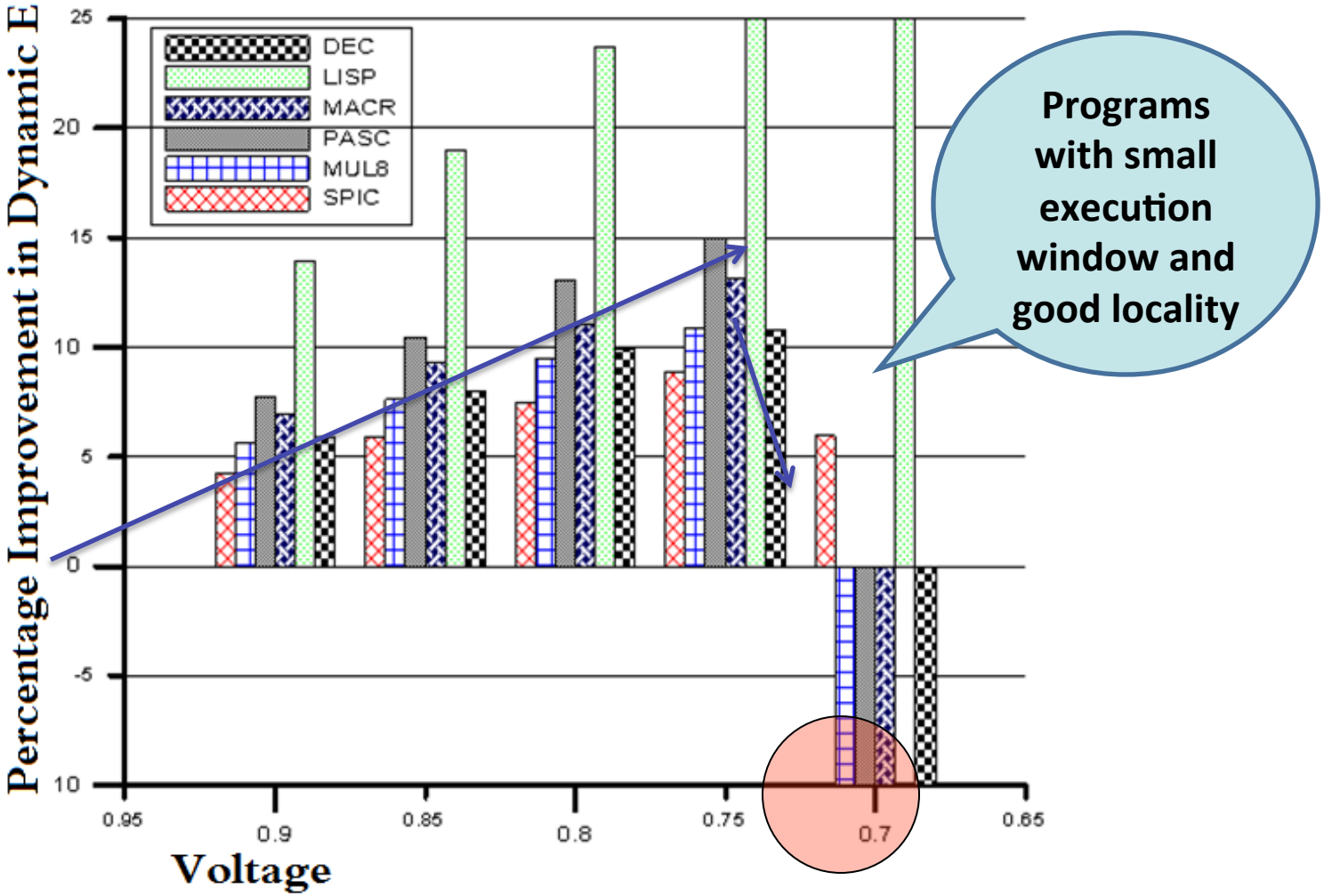
Voltage Scalable Cache
- All component are operated with a single scalable voltage.



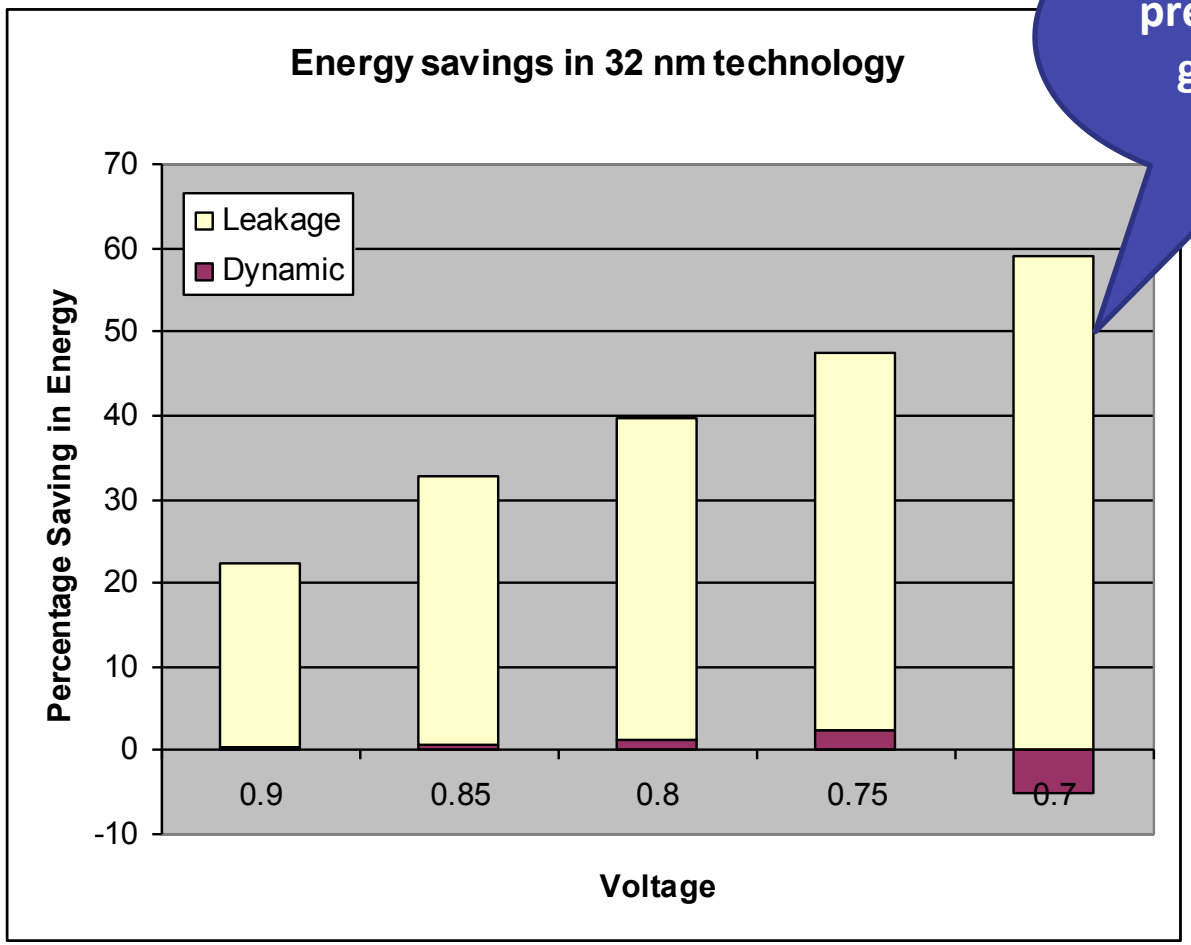
IDC System Architecture



Dynamic Energy Savings
32nm, 16KB cache with 16 row IDC



Dynamic and Leakage Energy Savings 32 nm, 16KB Cache, 16 Row IDC

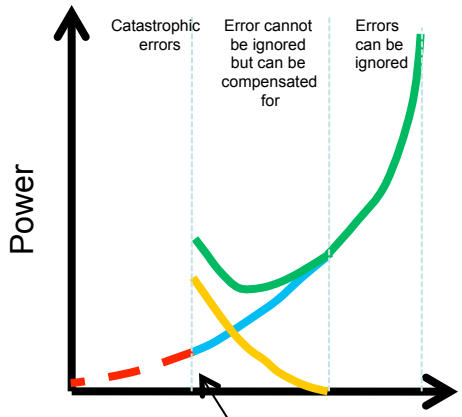


Leakage predominance gains huge savings

Comparison to block disabling architecture

Voltage	Miss rate in block disabling architecture	Miss rate in proposed architecture
0.9	3.90	3.90
0.85	4.10	3.90
0.8	5.9	3.90
0.75	14.6	3.92
0.7	59.5	6.45

Max power saving and min Vdd



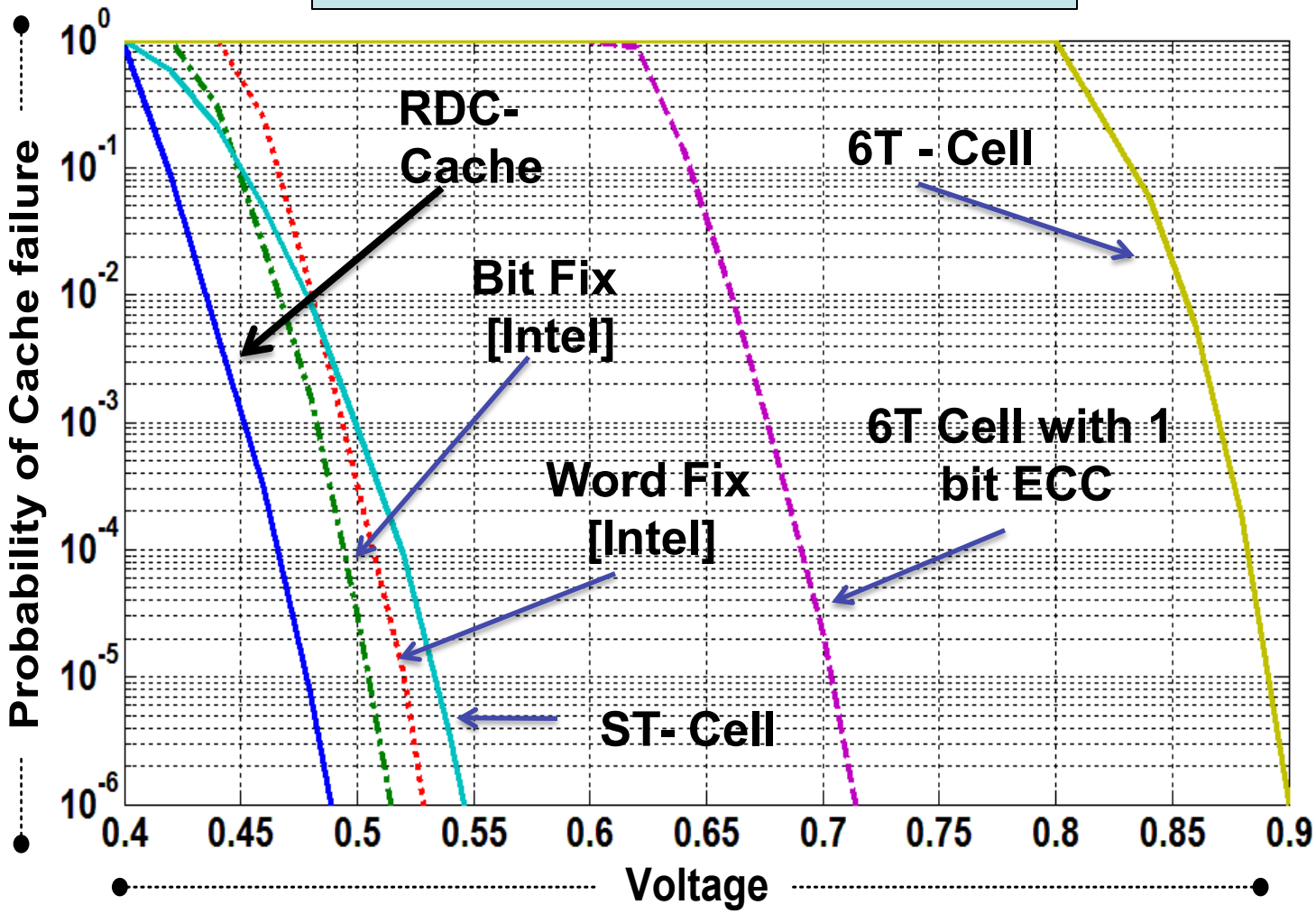
- LISP,SPIC min voltage bounded by logic min voltage
- DEC, MACR,PSC,MUL min voltage bounded by miss rate
- In 32nm leakage domination introduce better saving opportunities.

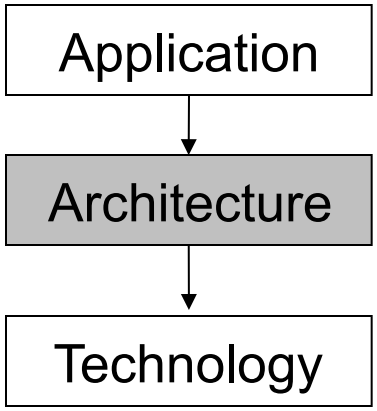
Address decoder fails: “catastrophic error”

Bench Mark	32 nm		65 nm	
	Min Voltage	Max Power saving	Min Voltage	Max Power Saving
DEC	0.72	43.28%	0.84	10.23%
LISP	0.72	49.38%	0.77	18.36%
MACR	0.72	44.38%	0.83	12.21%
PASC	0.72	45.60%	0.83	11.90%
MUL	0.75	43.12%	0.84	7.93%
SPIC	0.72	44.73%	0.77	15.21%

Change in the probability of cache failure

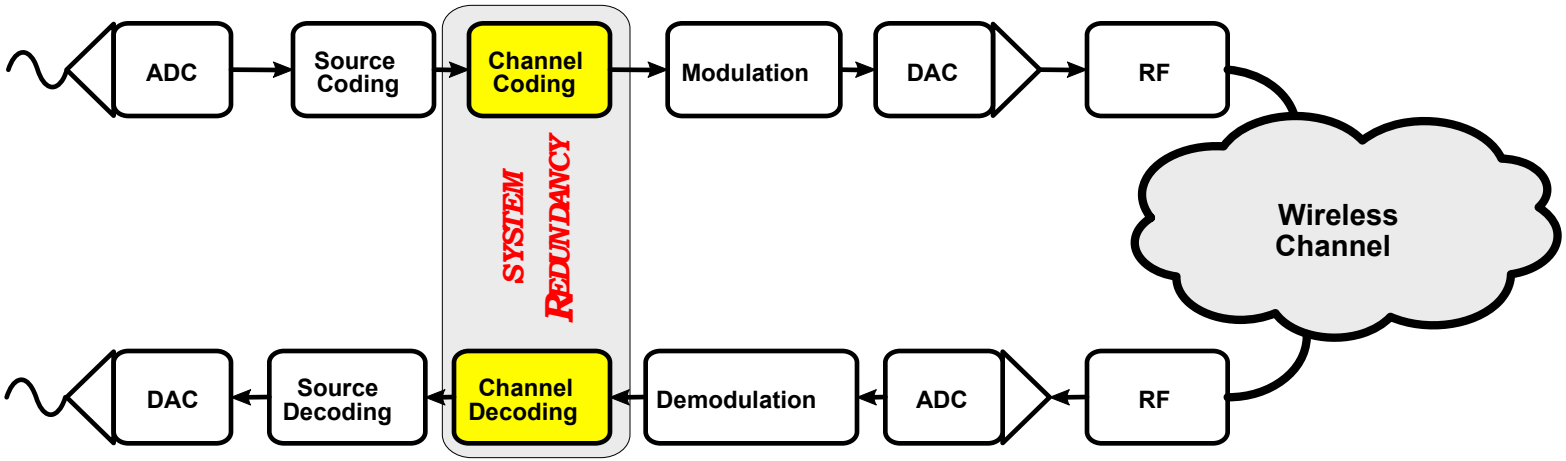
60-80% Power savings!





CASE STUDY: WCDMA (3GPP) RECEIVER

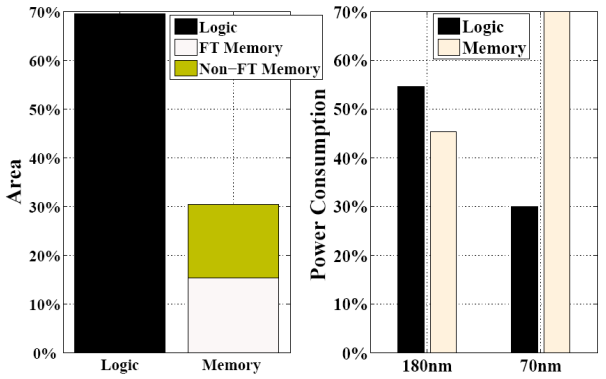
Application to communication systems

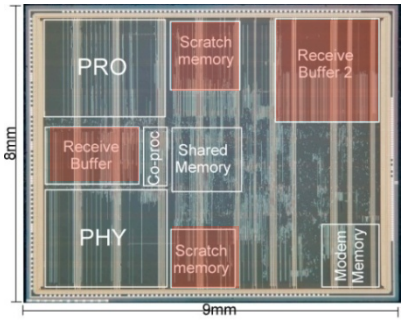


- What system parameters can be manipulated to allow a limited amount of hardware errors in memory?
- Channel coding techniques using Viterbi algorithm, Turbo codes, Convolutional codes, etc.
- FEC provides some degrees of freedom to compensate for varying degrees of data corruption.
  1. Depth of the interleaving memory
  2. Number of iterations required to converge to a target error performance.

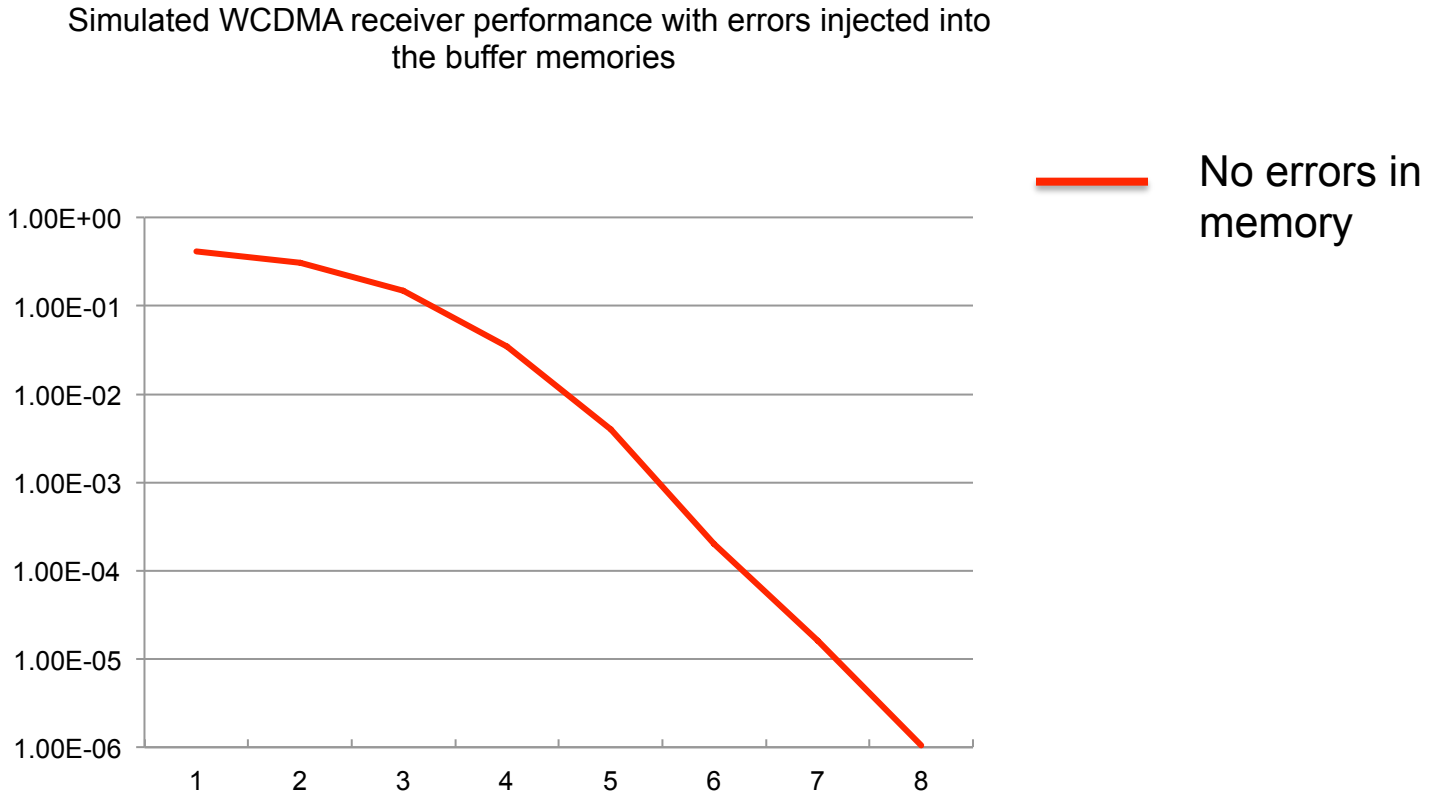
C3GPP modem -> receive mode

(noisy) memories at low Vdd



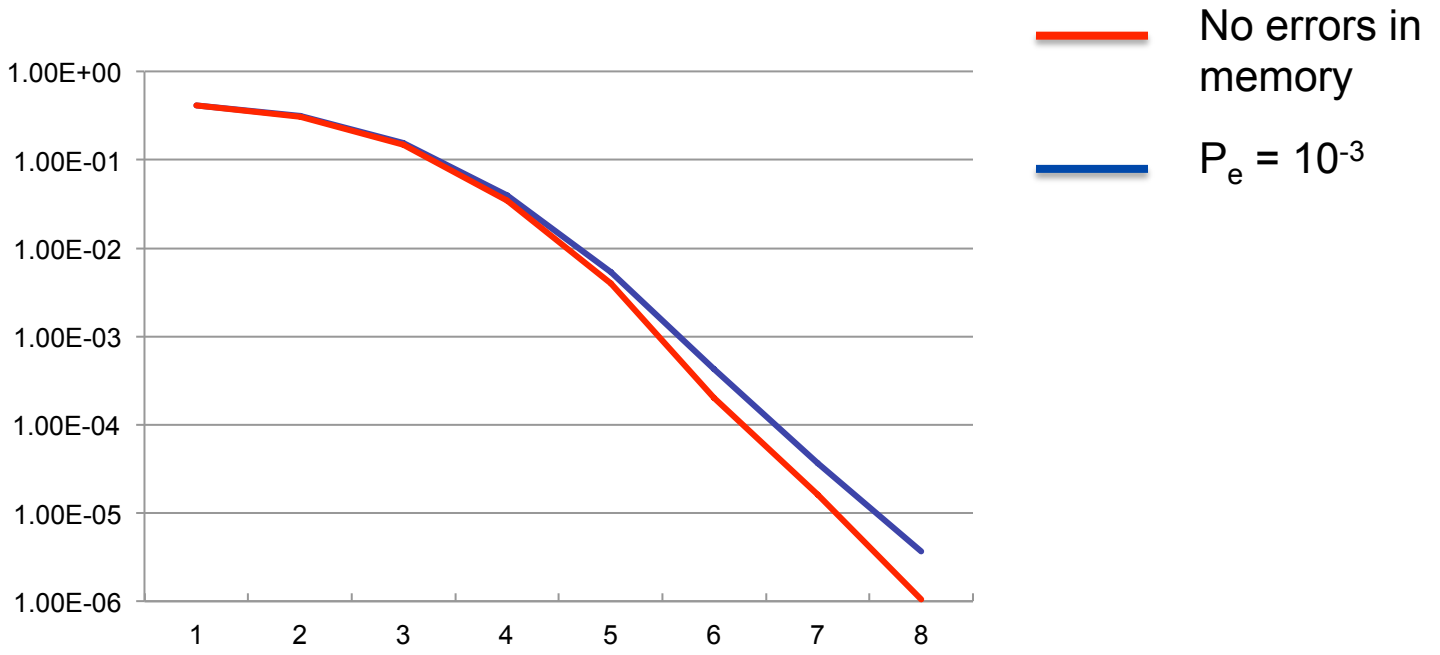


Comparing receiver performance (simulation)

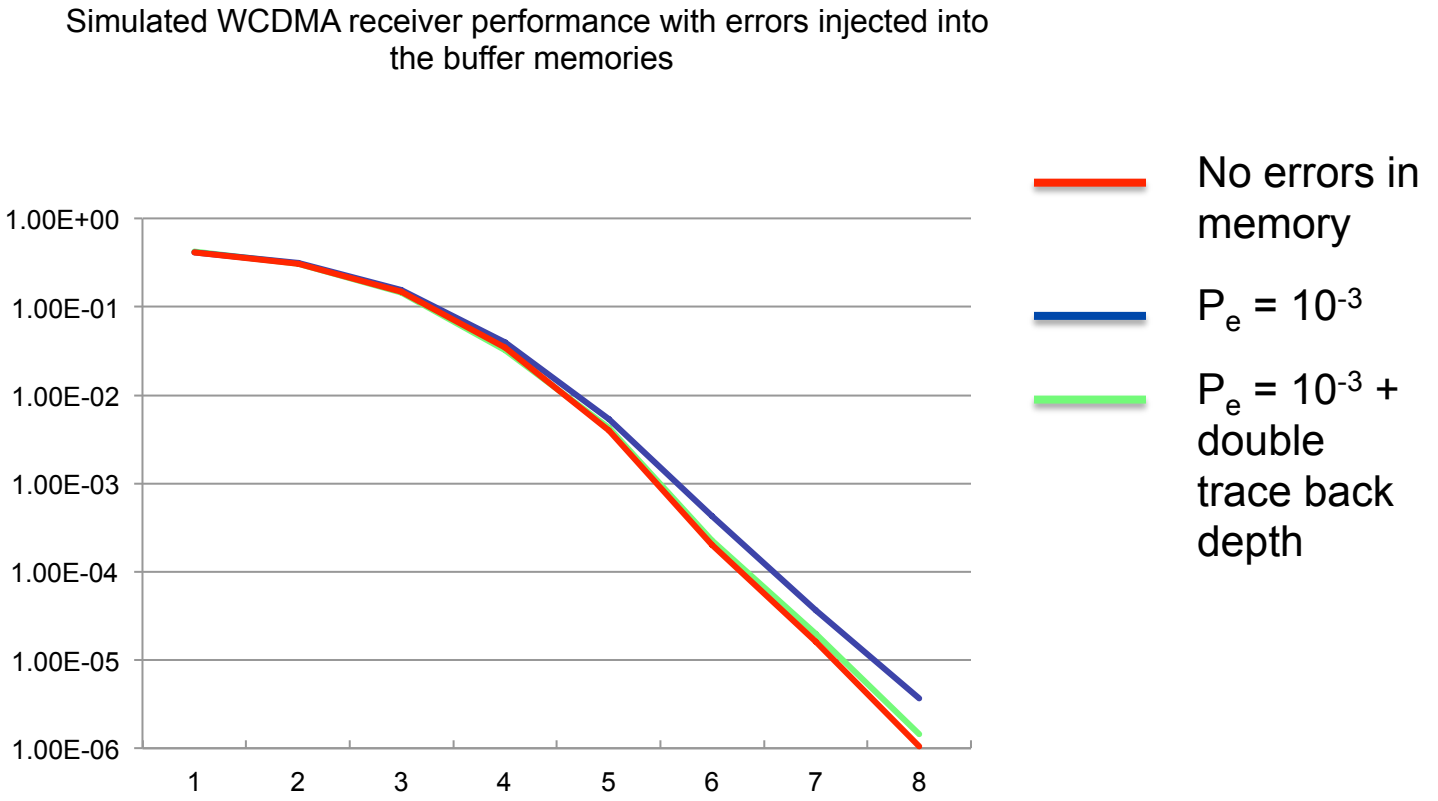


Comparing receiver performance (simulation)

Simulated WCDMA receiver performance with errors injected into the buffer memories

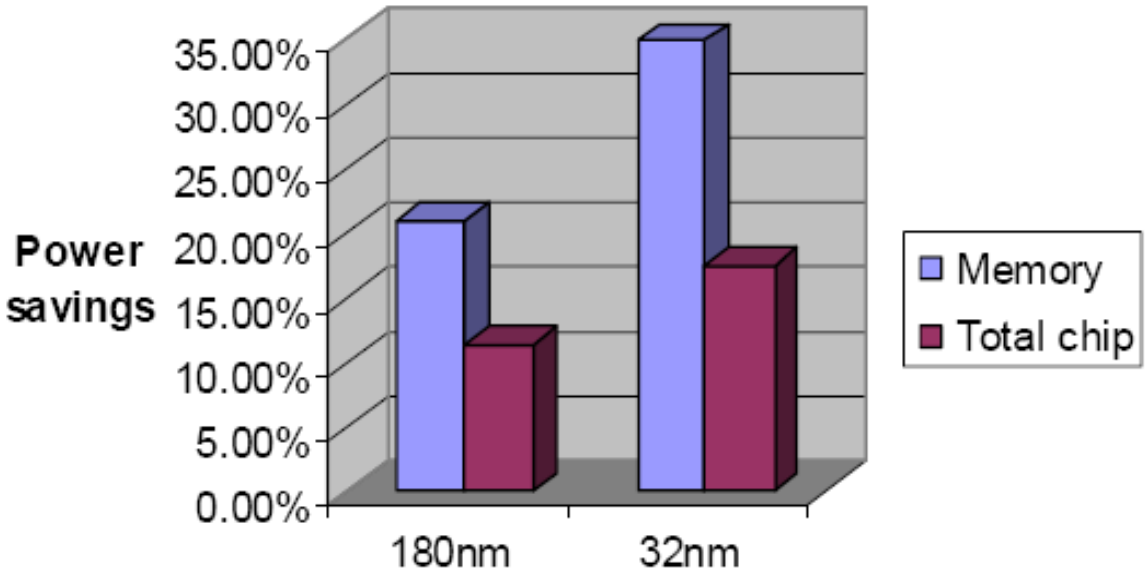


Comparing receiver performance (simulation)



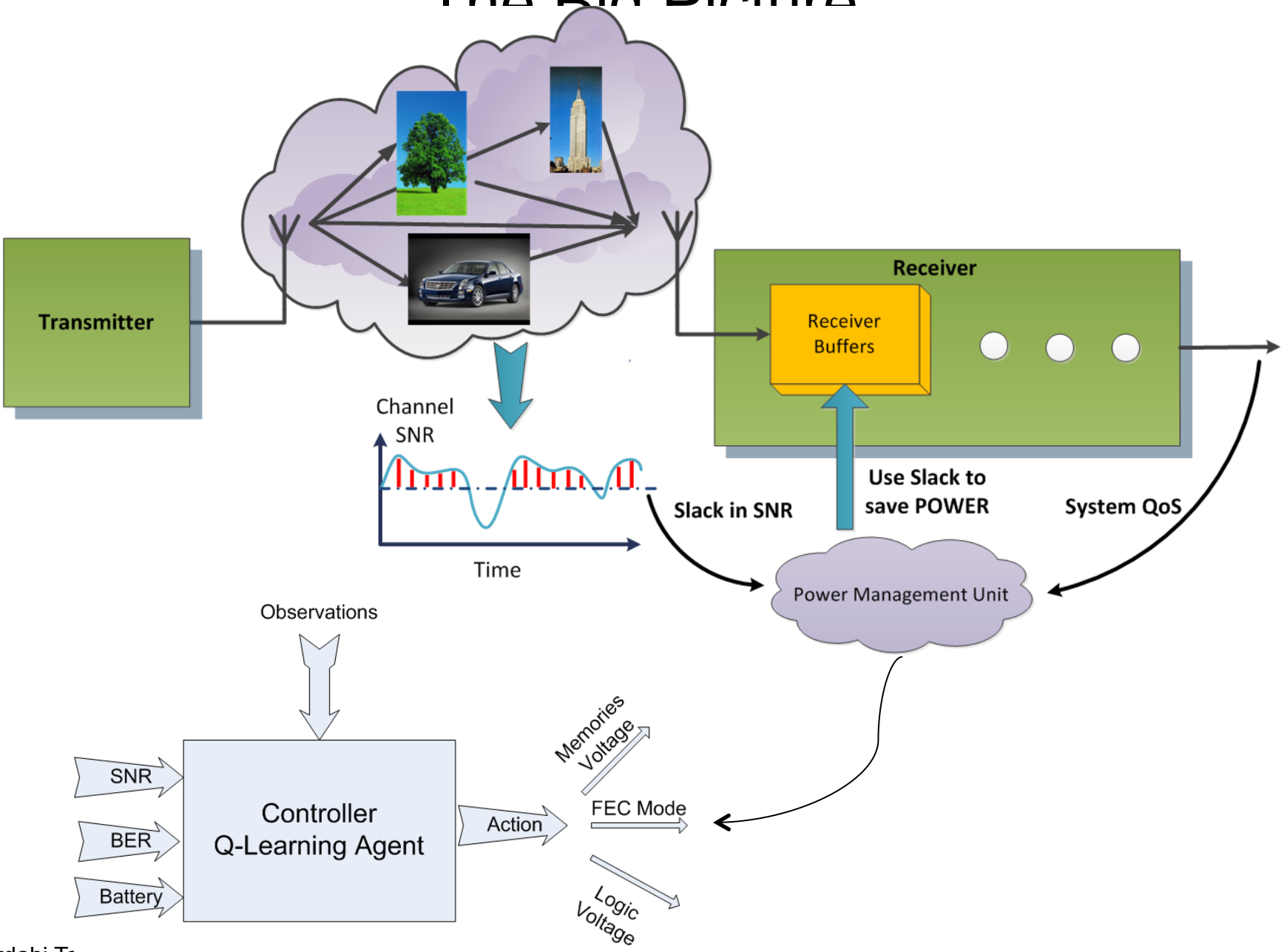
By doubling the trace back depth and introducing 0.1% errors in the buffer memories, the faulty system performs almost the same as the system with no errors in terms of BER.

Expected power savings



Up to 35% power saving can be achieved with no modifications at the system level.

The Big Picture



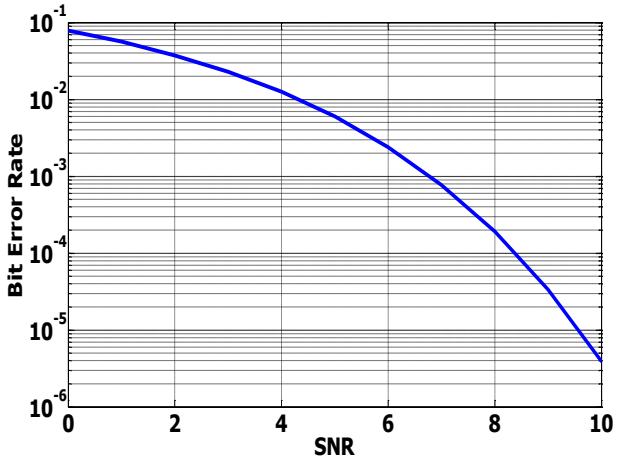
Power Manger for Ultra Low Power Mobile Device

How would the system perform when operated at low power mode ?

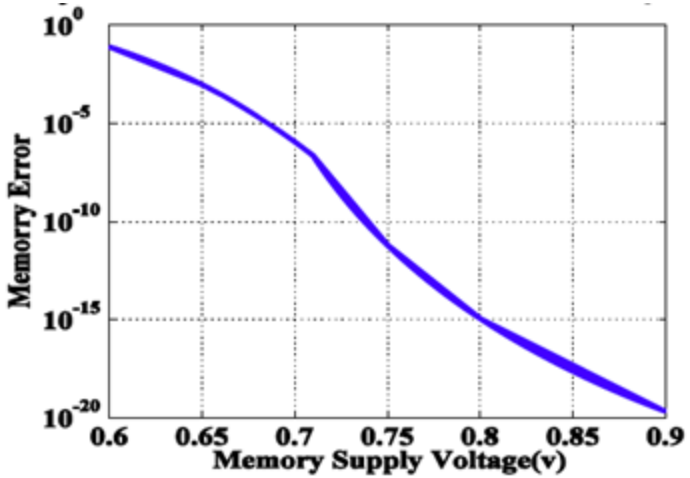
What are the parameters that control and affect the system metric (BER)?

- Simulation of the system
  - Low accuracy and simulation time
  - Lack of scalability : large combination of system mode and settings
- Mathematically Modeling
  - Obtain the distribution of the data after each block of the up to the slicer

Fault Tolerant Adaptation

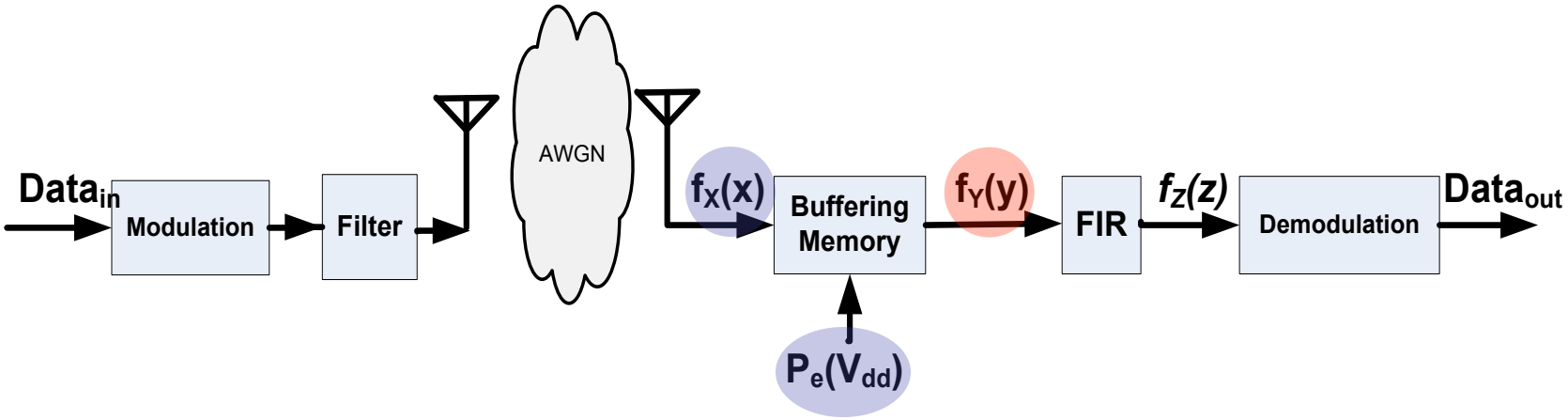


Dual ity

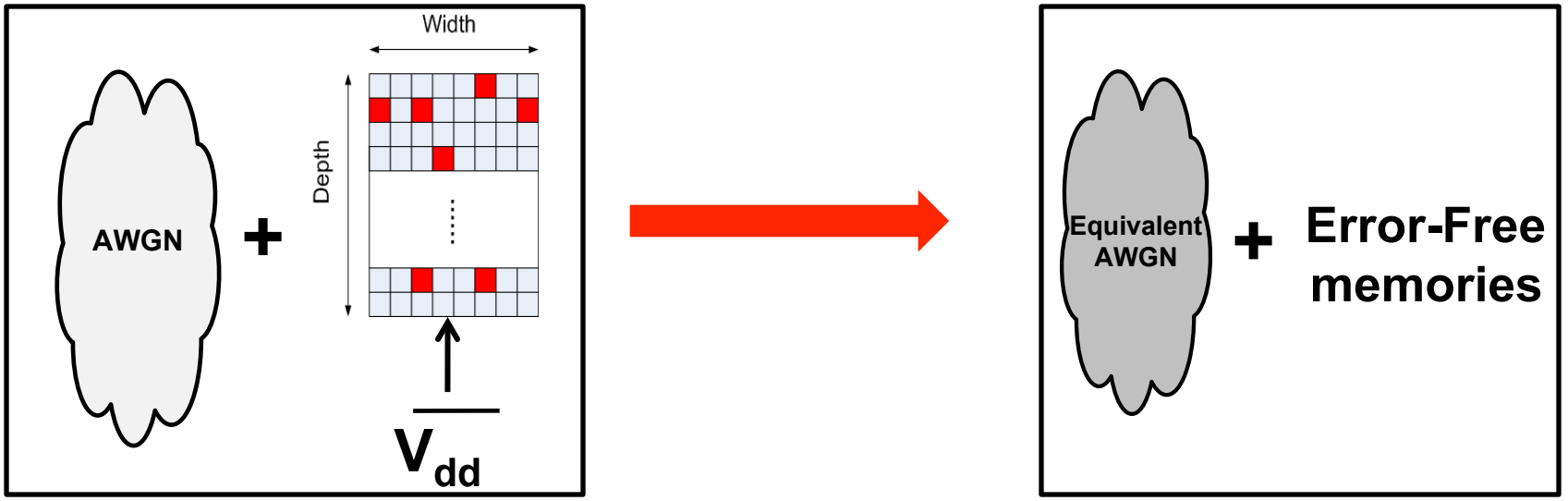


Memory can be considered as another channel that contributes supply voltage dependent

Simplified System Model



Equi-Noise



$$N\sim(\mu_0,\sigma_0) \longrightarrow N\sim(\mu(\mu_0,\overline{V_{dd}}),\sigma(\sigma_0,\overline{V_{dd}}))$$

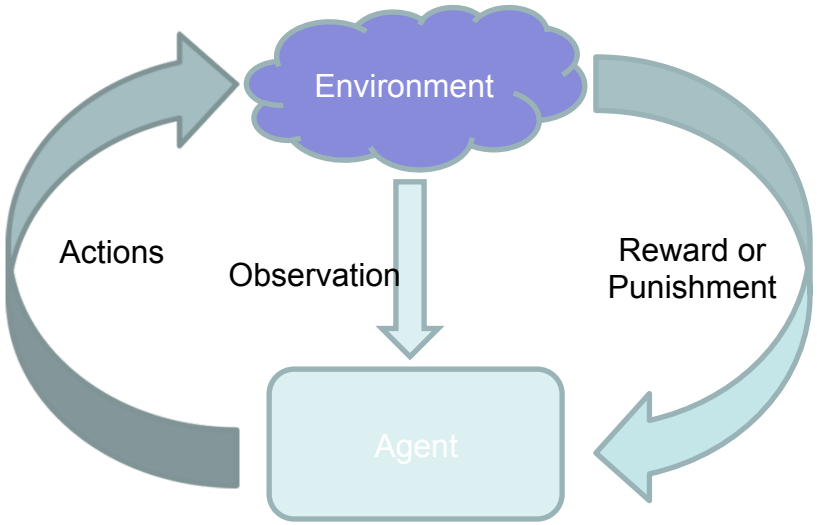
Reinforcement Learning

- The standard reinforcement-learning model consists of an agent that interacts with the environment through observations and actions.
- On each step of interaction with the environment:

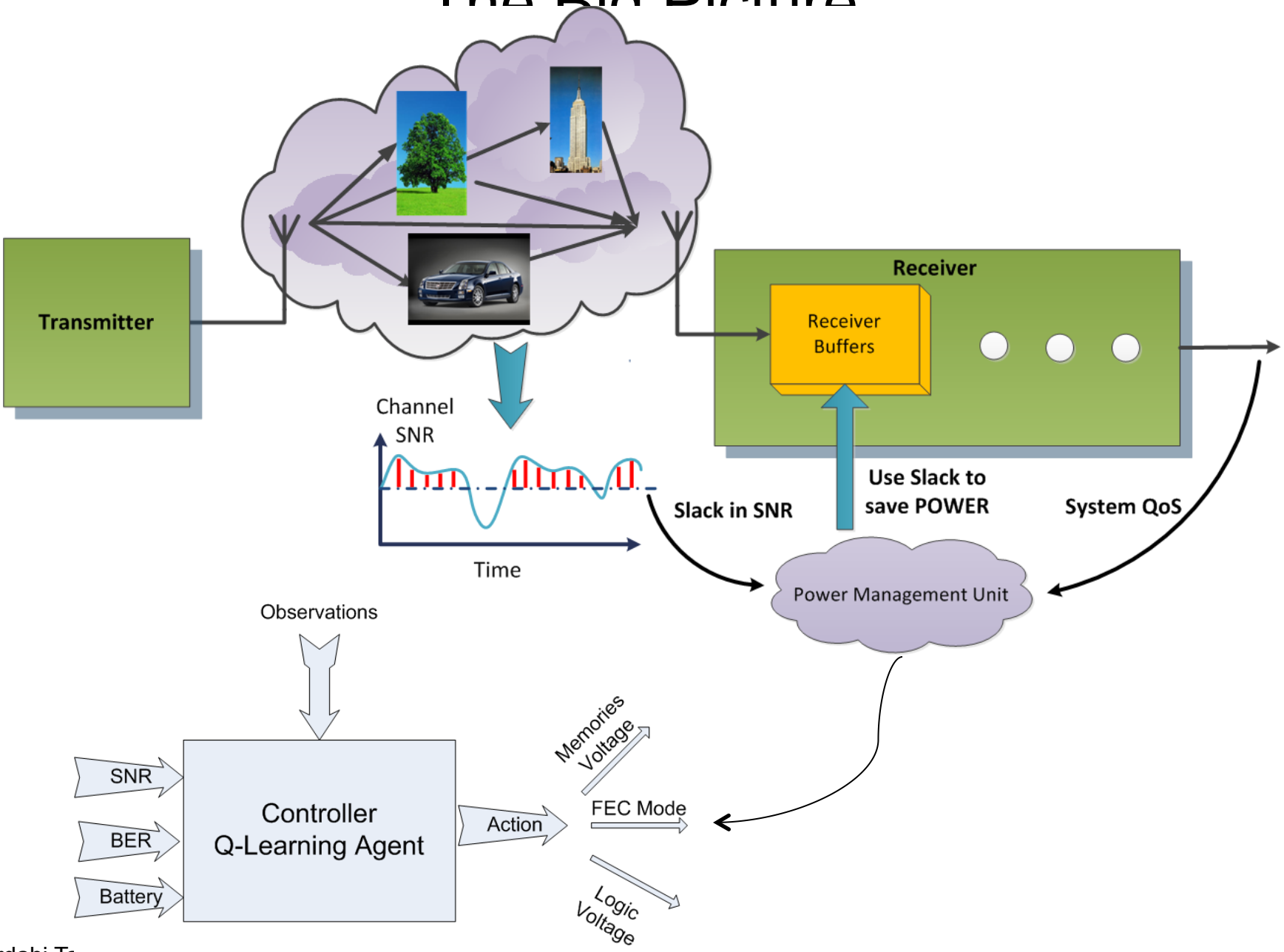
  1. The agent receives an input from the environment that describes the current state of the environment.

  2. The agent issue an action as an output which changes the state of the environment.

  3. The agent receives a scalar-valued reward which indicates the value of the state transition



The Big Picture



Q-Learner Algorithm

Initialize Q table entry for each Q(*s, a) pair*

Do

1. Observe the current state of the system (SNR,V1,V2) = S(t)
2. Determine whether follow the Q-learning algorithm or random next state
3. The agent takes an action, A(t), based on the current state S(t)
4. The agent observes the next state of the system S(t+1) and measures the BER
5. The agent determines the action a‘ that maximizes Q[S(t+1),a’]
6. The agent calculates the expected reward

$$reward = c(s,a) = P_c + P_{switching} + \lambda \times BER$$

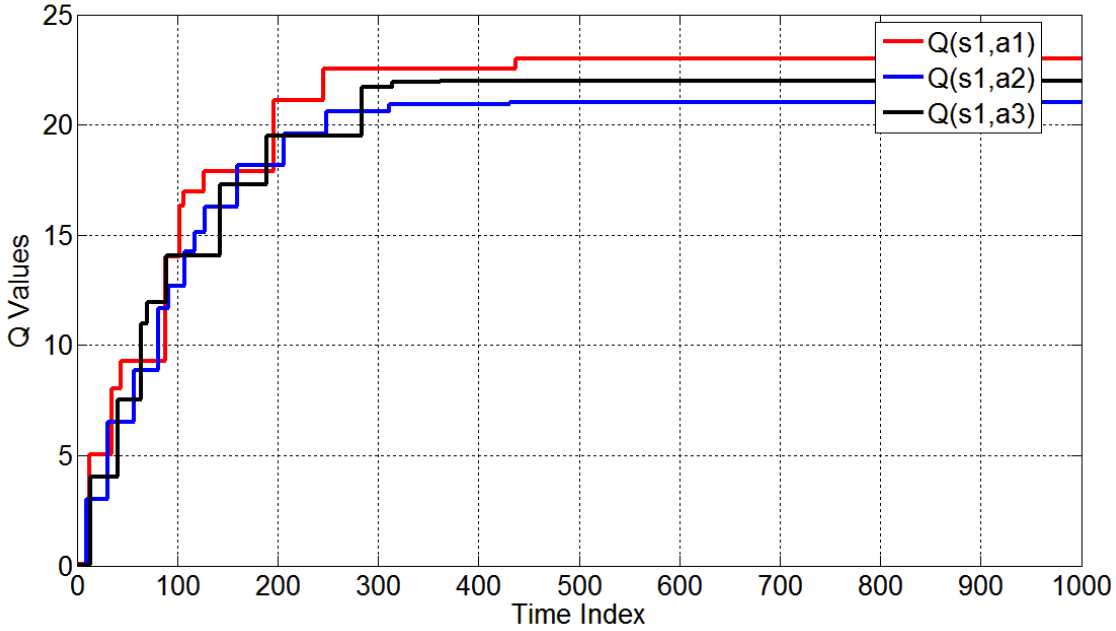
7. update the current Q-value

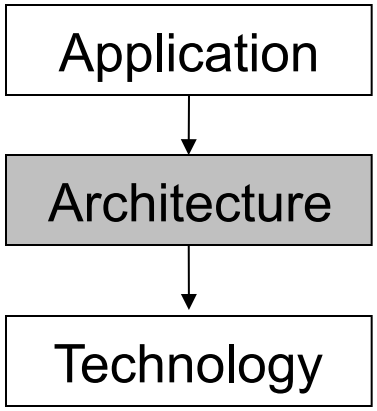
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \varepsilon \times (c(s,a) + \gamma \min_a Q(s_{t+1}, a) - Q(s_t, a_t))$$

end do loop

Training Results

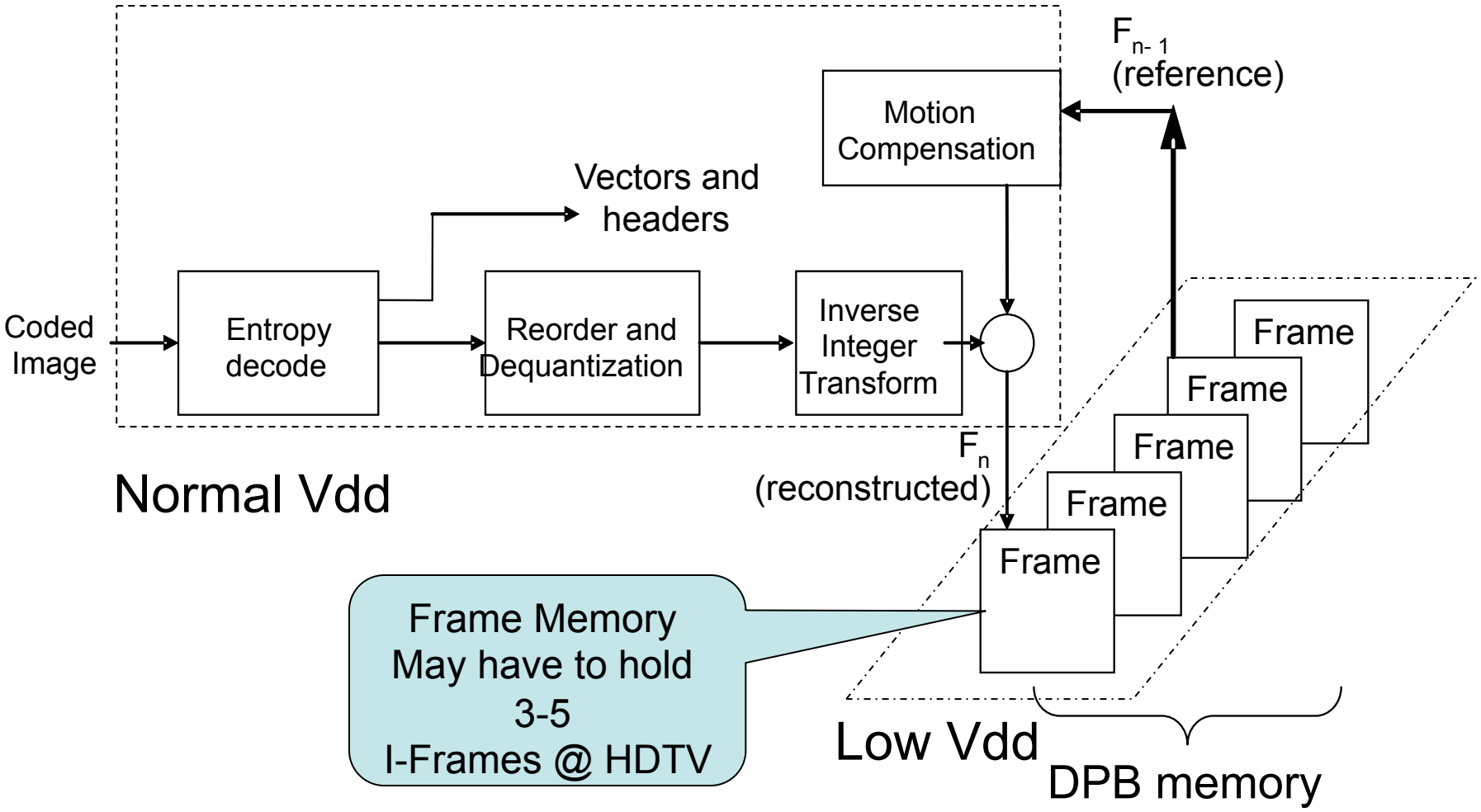
	A1	A2	A3
S1	23.128	21.154	22.183
S2	19.834	18.435	16.342
S3	22.838	23.893	20.838





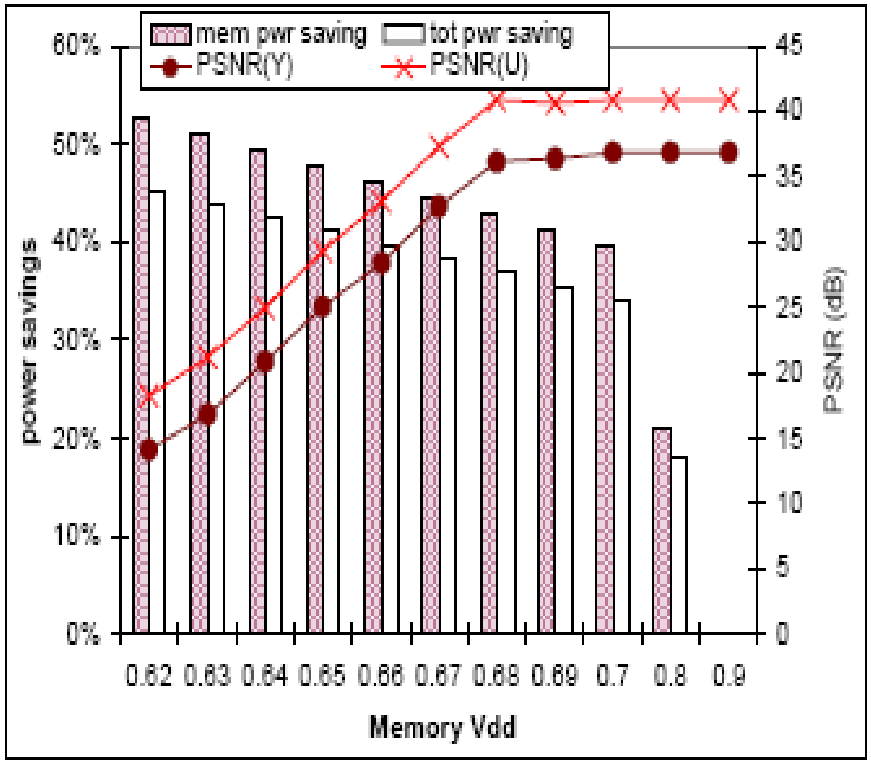
CASE STUDY: MULTIMEDIA (H. 264 DECODER)

video codecs (H.264 decoder)



How bad can the output get When V_{dd} is lowered on DPB?

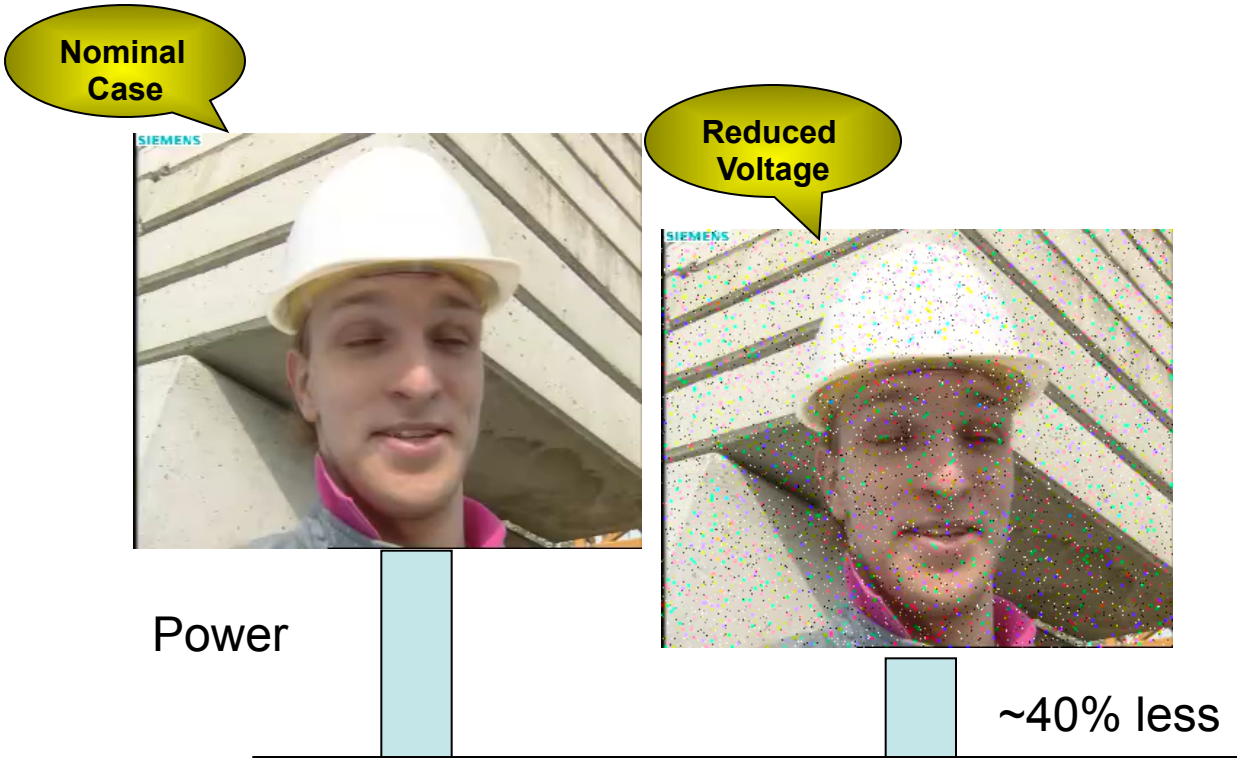
Fig. 7 Snap shot of CIF-sized "FOREMAN" video image (29th frame where an I-frame comes every 30th.)
Encoding Configuration : IPPPP, QP=28, Reference Frame#=2, I frame Period =30



But, we can save > 40% in power!

How bad can the output get When V_{dd} is lowered on DPB?

- Can save 40% power
- Serious degradation in image quality (20db drop in PSNR)



Can we improve the image quality?