Exploiting On-Chip Data Behavior for Delay Minimization

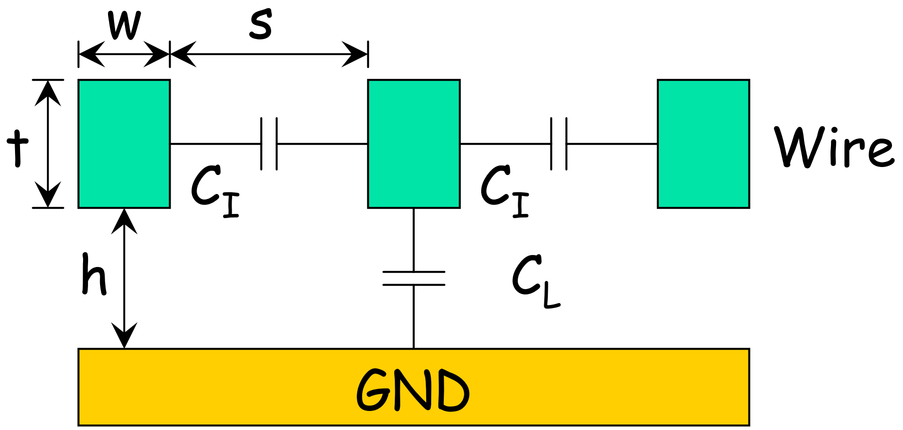
Nallamothu Satyanarayana
Adam's Engineering College
Palvoncha, Andhra Pradesh
India

Outline

- Motivation
- Analytical Model for Delay
- Crosstalk Classes
- Related Works
- Our Approach
- Experimental Results
- Conclusion

Motivation

- In deep sub-micron level technologies
  - Inter-wire capacitance (C_I) is higher compared to the wire-to-substrate capacitance (C_L).
  - Large Propagation delay due to opposite transitions and relative switching activity on adjacent wires.
  - As the technology shrinks, the Inter-Wire capacitance becoming more dominant, results it hurts the system performance.



Analytical Model for Delay

- Let $\lambda = C_I/C_L$, $\Delta_k = d^k{}_{t+1} - d^k{}_t$, and R_T be the total resistance.
- Let d_t be a *n*-bit data present on the bus.
- The propagation delay for transmitting a *n*-bit data d_{t+1} is calculated by Chandrakasan:

$$T(d_t,d_{t+1}) = \mathit{max}\{T_k(d_t,d_{t+1}) \mid 1 \le k \le n\},$$

$$T_k(d_t,d_{t+1}) = C_L R_T((1+2\lambda)\Delta_k{}^2 - \lambda(\Delta_{k-1}+\Delta_{k+1}))\Delta_k,\ 1 < k < n$$

Δk is the transition occurring on line k,

$\Delta k = 1$ for 0 to 1 transition

$\Delta k = 1$ for 1 to 0 transition

$\Delta k = 0$ for no transition

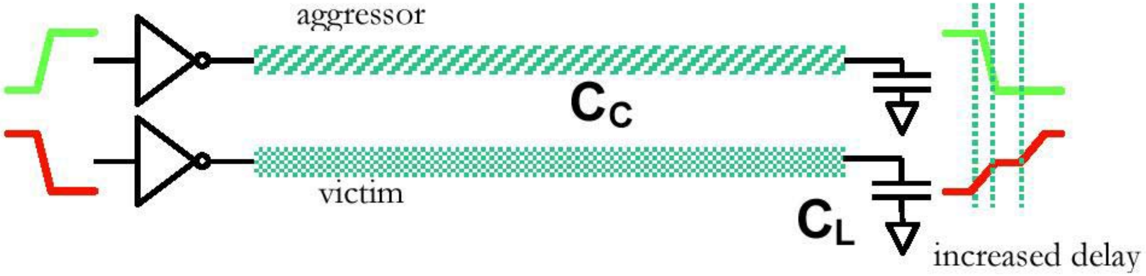
An Example

- If $d_t = 010$ and $d_{t+1} = 101$
    - $T(d_t, d_{t+1}) = C_L R_T(1+4\lambda)$

- If $d_t = 000$ and $d_{t+1} = 111$
    - $T(d_t, d_{t+1}) = C_L R_T$

Crosstalk

- Large propagation delay due to opposite transitions on adjacent wires.
- High power dissipation for driving on-chip buses.



Crosstalk Classes

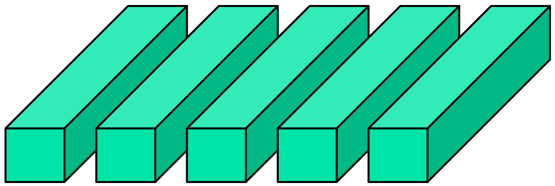
- Transition patterns are classified into six different crosstalk classes.

Crosstalk Class	Relative Delay on the middle wire	Transition Pattern
1	0	---,--↑,↑--,--↓,↓--,↑-↑, ↑-↓,↓-↑,↓-↓
2	C_LR_T	↑↑↑,↓↓↓
3	$C_LR_T(1+\lambda)$	-↑↑,↑↑-,↓↓-,-↓↓
4	$C_LR_T(1+2\lambda)$	-↑-,-↓-,↓↓↑,↑↓↓,↑↑↓,↓↑↑
5	$C_LR_T(1+3\lambda)$	-↑↓,-↓↑,↓↑-,↑↓-
6	$C_LR_T(1+4\lambda)$	↑↓↑,↓↑↓

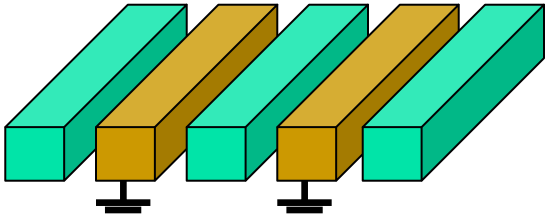
Related Works

- Existing techniques use large spatial redundancy.
  - Shielding techniques
  - Crosstalk Preventing Coding

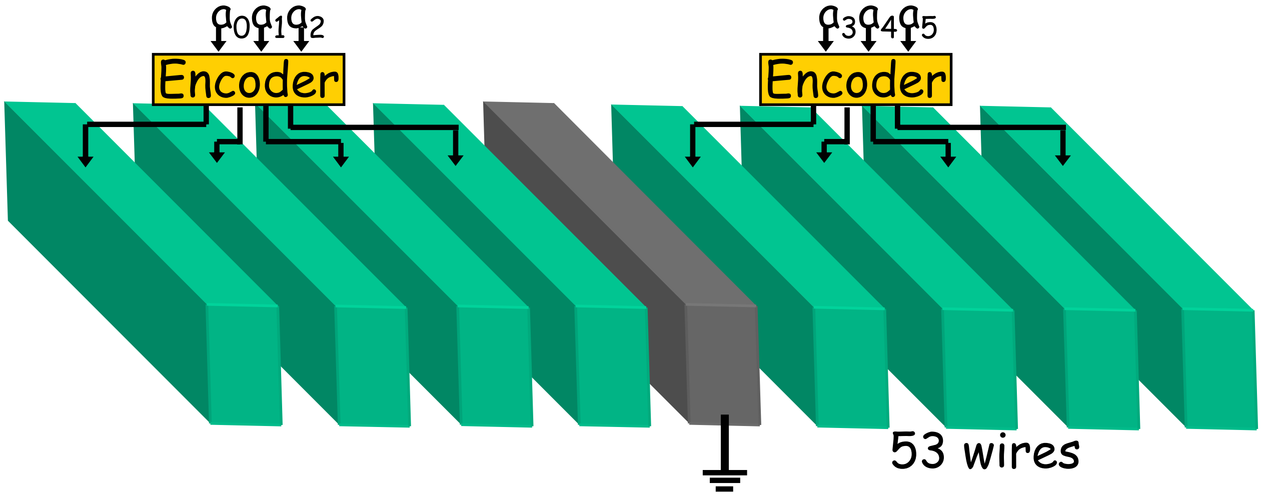
Original (32 Wires)



Shielding

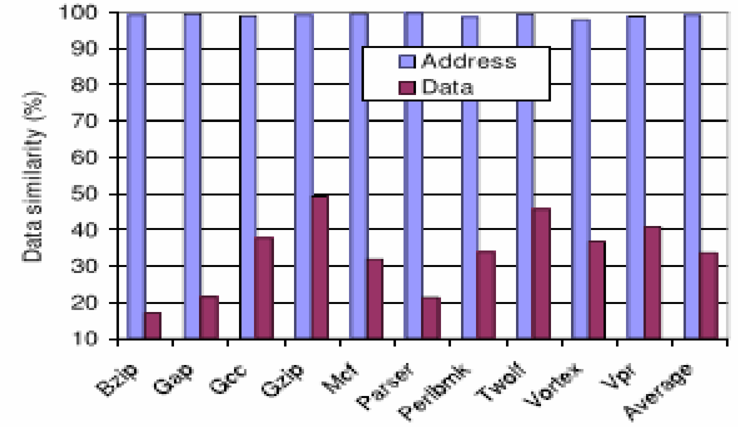


63 wires



Our Approaches

- Most of the times, the MSB 16 bits of a 32-bit to be transmitted is same as that of the present data on the bus.

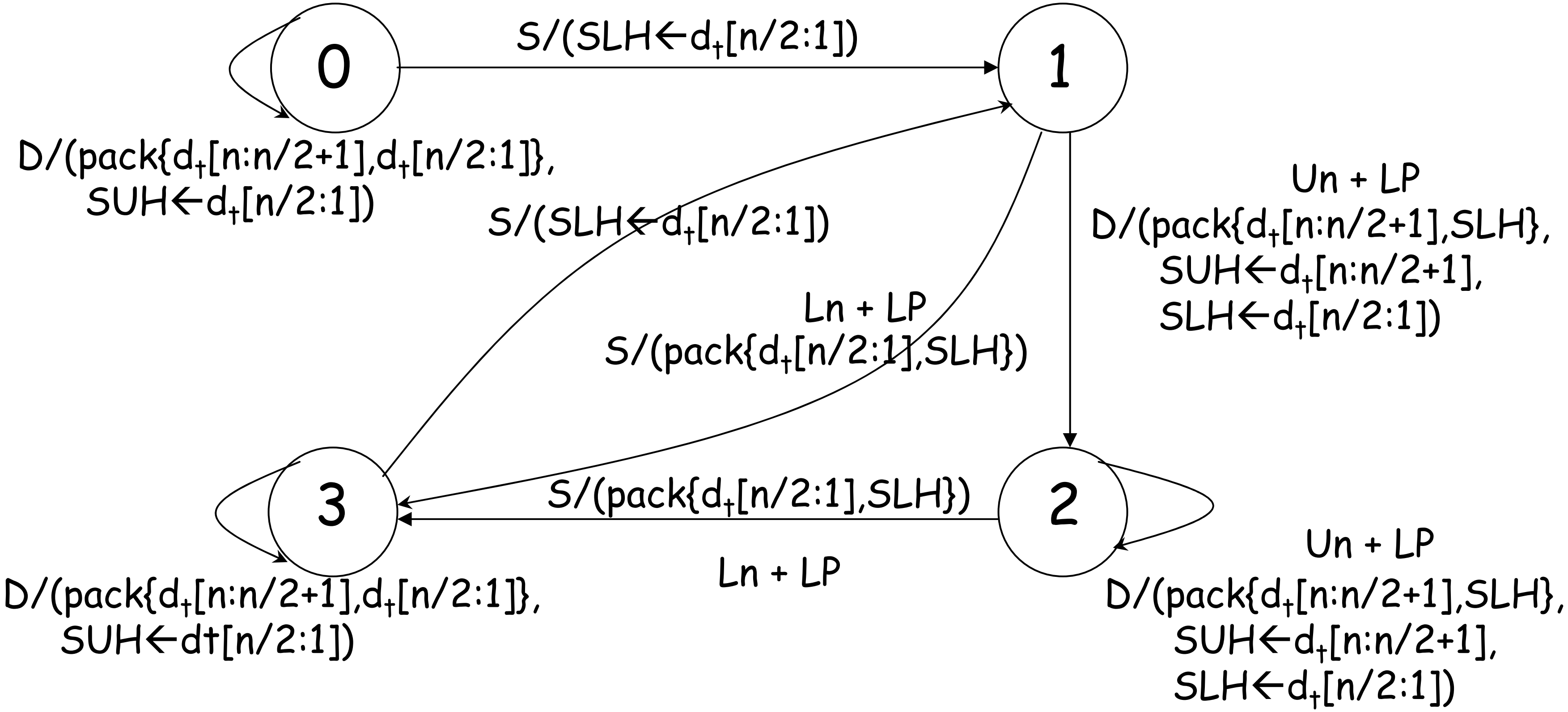


- 99% and 33% of times upper half of the data to be transmitted is same of that of present data on the bus for address and data buses.

Our Approaches

- To reduce the propagation delay, we are proposed two techniques:
  - **Data Packing (DPack):** Two different n/2 bit data can be packed together and transmitted.
  - **Data Permutation (DPerm):** MSB bit placed between every pair of LSB bits. MSB bits are transmitted can act as shield wires when similarity in MSB 16-bits.

Data Packing Technique



0

1

2

3

S/(SLH←d_t[n/2:1])

D/(pack{d_t[n:n/2+1],d_t[n/2:1]},
SUH←d_t[n/2:1])

S/(SLH←d_t[n/2:1])

Ln + LP
S/(pack{d_t[n/2:1],SLH})

Un + LP
D/(pack{d_t[n:n/2+1],SLH},
SUH←d_t[n:n/2+1],
SLH←d_t[n/2:1])

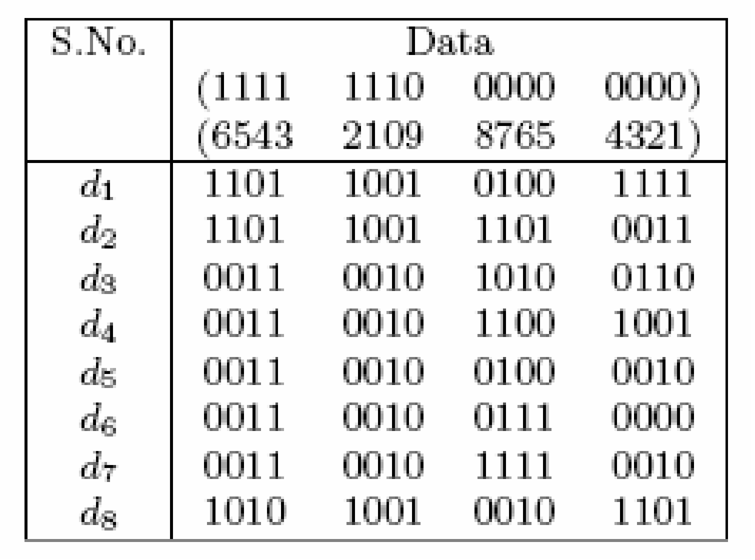
S/(pack{d_t[n/2:1],SLH})

Ln + LP

D/(pack{d_t[n:n/2+1],d_t[n/2:1]},
SUH←dt[n/2:1])

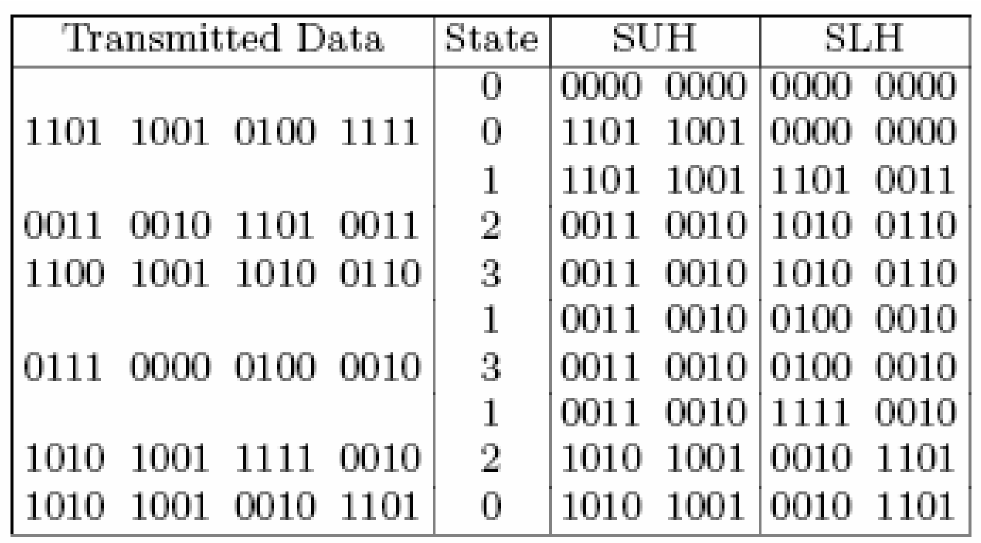
Un + LP
D/(pack{d_t[n:n/2+1],SLH},
SUH←d_t[n:n/2+1],
SLH←d_t[n/2:1])

Illustrating DPack Technique



S.No.	Data			
	(1111	1110	0000	0000)
	(6543	2109	8765	4321)
d_1	1101	1001	0100	1111
d_2	1101	1001	1101	0011
d_3	0011	0010	1010	0110
d_4	0011	0010	1100	1001
d_5	0011	0010	0100	0010
d_6	0011	0010	0111	0000
d_7	0011	0010	1111	0010
d_8	1010	1001	0010	1101

Original data to be transmitted



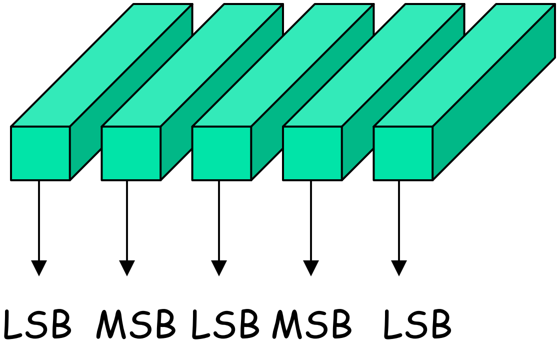
Transmitted Data	State	SUH	SLH
	0	0000 0000	0000 0000
1101 1001 0100 1111	0	1101 1001	0000 0000
	1	1101 1001	1101 0011
0011 0010 1101 0011	2	0011 0010	1010 0110
1100 1001 1010 0110	3	0011 0010	1010 0110
	1	0011 0010	0100 0010
0111 0000 0100 0010	3	0011 0010	0100 0010
	1	0011 0010	1111 0010
1010 1001 1111 0010	2	1010 1001	0010 1101
1010 1001 0010 1101	0	1010 1001	0010 1101

Data transmitted using DPack technique

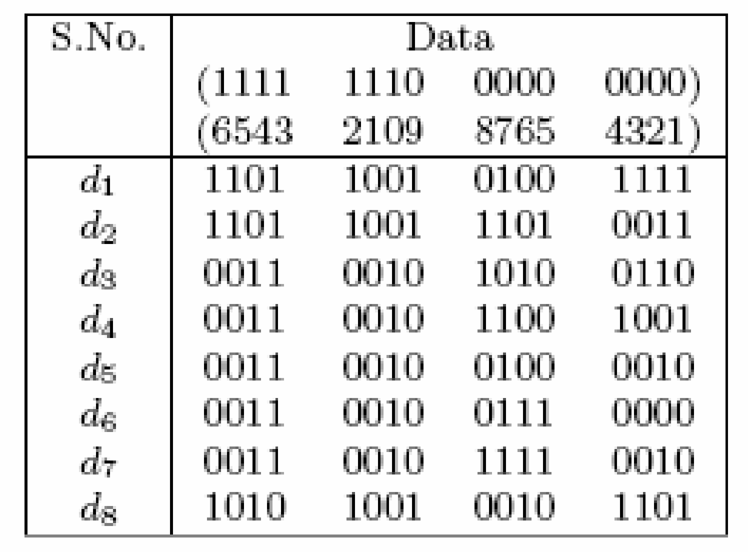
1bit ready signal + 1bit shield wire + 32 bits address / data lines + 1 bit shield wire + 3 bits status word = 38 wires required.

Data Permutation Technique

- For every data d:$a_n a_{n-1} \ldots a_2 a_1$, we transmit the permuted data d':$a_n a_{n/2} a_{n-1} \ldots a_2 a_{n/2+1} a_1$.
- If d_1:$a_n \ldots a_{n/2+1} a_{n/2} \ldots a_1$ & d_2:$a_n \ldots a_{n/2+1} a'_{n/2} \ldots a'_1$, then transmitting d'_1 & d'_2 can eliminate opposite transitions on adjacent wires.

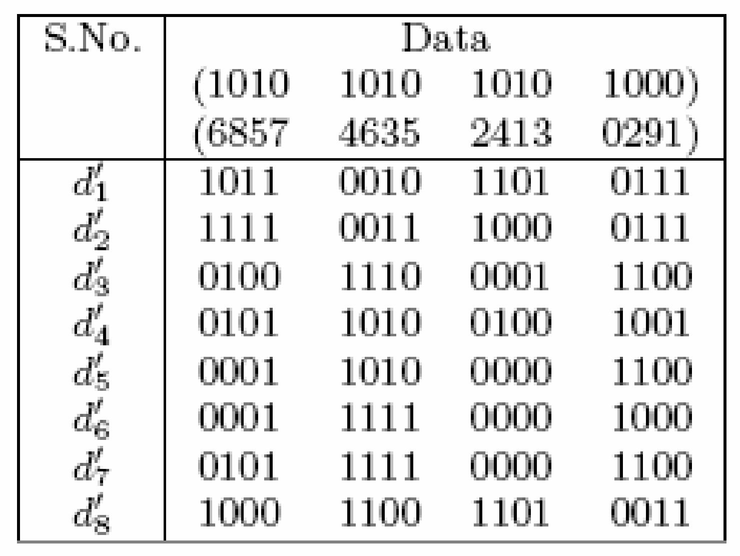


Illustrating DPerm Technique



S.No.	Data			
	(1111	1110	0000	0000)
	(6543	2109	8765	4321)
d_1	1101	1001	0100	1111
d_2	1101	1001	1101	0011
d_3	0011	0010	1010	0110
d_4	0011	0010	1100	1001
d_5	0011	0010	0100	0010
d_6	0011	0010	0111	0000
d_7	0011	0010	1111	0010
d_8	1010	1001	0010	1101

Original data to be transmitted



S.No.	Data			
	(1010	1010	1010	1000)
	(6857	4635	2413	0291)
d'_1	1011	0010	1101	0111
d'_2	1111	0011	1000	0111
d'_3	0100	1110	0001	1100
d'_4	0101	1010	0100	1001
d'_5	0001	1010	0000	1100
d'_6	0001	1111	0000	1000
d'_7	0101	1111	0000	1100
d'_8	1000	1100	1101	0011

Data transmitted using DPerm technique

32 bits address / data lines + 1 bit shield wire + 1 bit ready signal

= 34 wires required.

Data Transmission Using Variable Delay

- Analyze the crosstalk class of next data w.r.t. the data present on the bus.
- If the crosstalk class is from the set {1,2,3}, transmit the next data with a delay of $2[C_L R_T(1+4\lambda)/5]$.
- If the crosstalk class is $n \in \{4,5,6\}$, transmit the next data with a delay of $(n-1)[C_L R_T(1+4\lambda)/5]$.

Experimental Setup

- We designed the Crosstalk Class Analyzer, crossbar switch, and the codec in Verilog and synthesized them using the Synopsys Design Compiler with TSMC 90nm technology library.
- The Predictive Technology Model is used to calculate the ground and coupling capacitance of interconnects.

Experimental Setup

- Technology parameters used in the experiments are shown below.

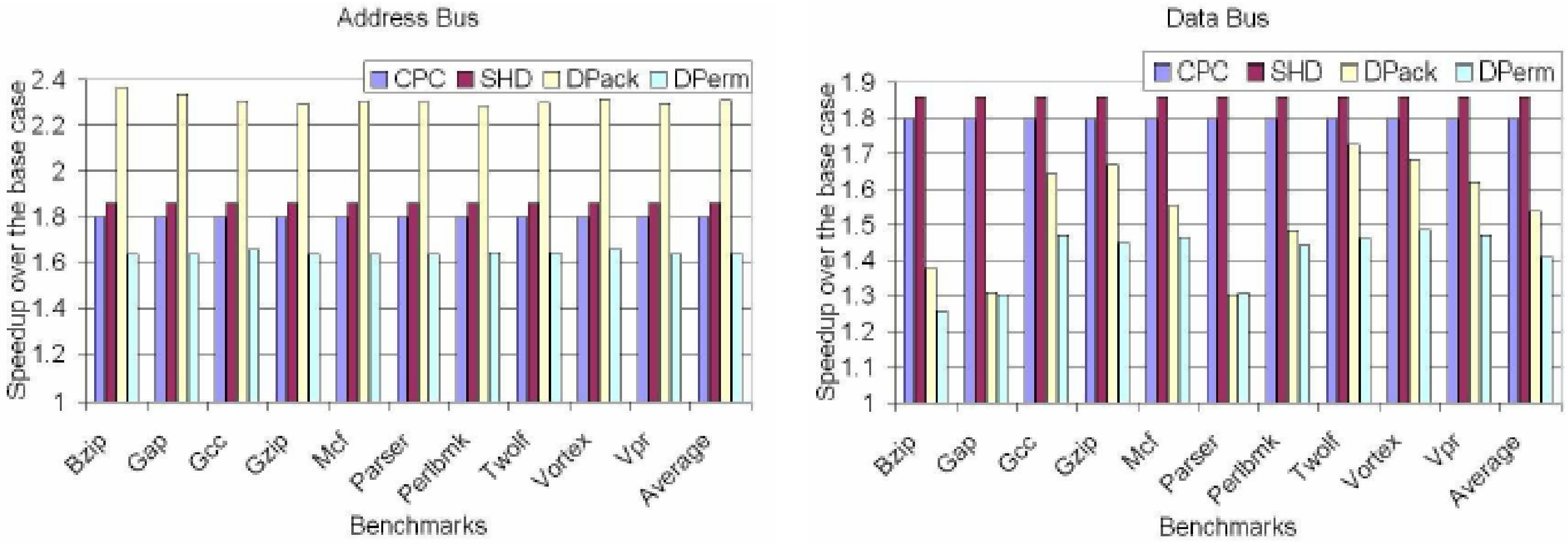
Parameter	Technology nodes			
	90nm	65nm	45nm	32nm
Wire width (nm)	205	145	102.5	70
Space (nm)	205	145	102.5	70
Thickness (nm)	430.5	319	235.75	168
Height of ILD (nm)	398.5	290	215.25	154
Dielectric constant	3.3	2.7	2.3	2.3

Experimental Setup

- We used Simplescalar 3.0 tool-set to perform experimental analysis and the SPEC-2000 CINT benchmark suite to simulate the performance of different on-chip buses between the processor datapath and L1 I-cache/D-cache.
- We present the results of IL1 address bus and DL1 data bus.

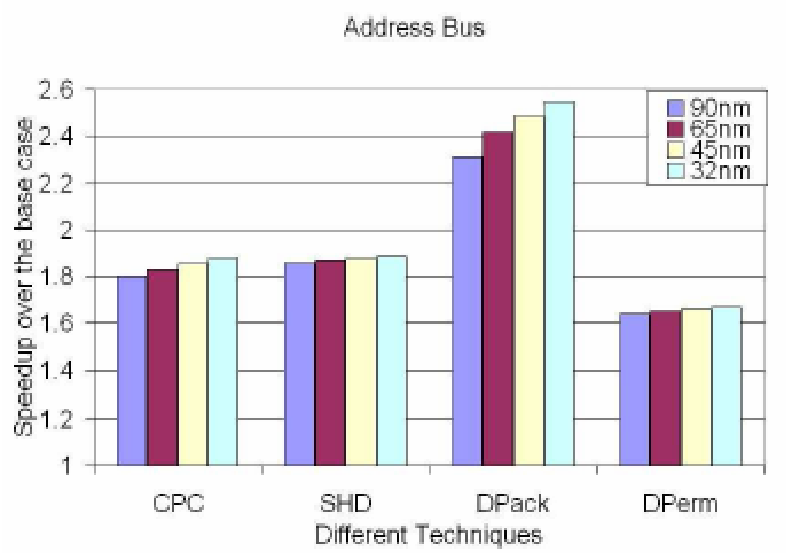
Experimental Results

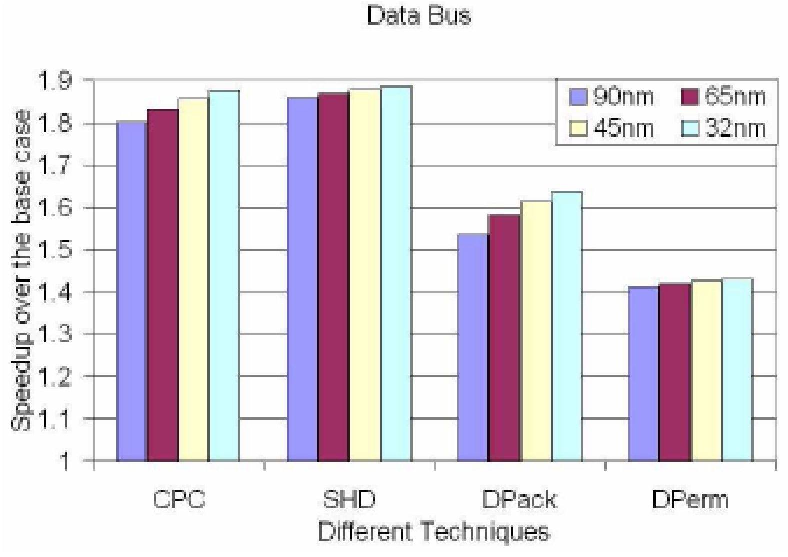
- Benchmark-wise speedup over the base case for 90nm process technology.



Experimental Results

- Average speedup over the base case for different process technologies.





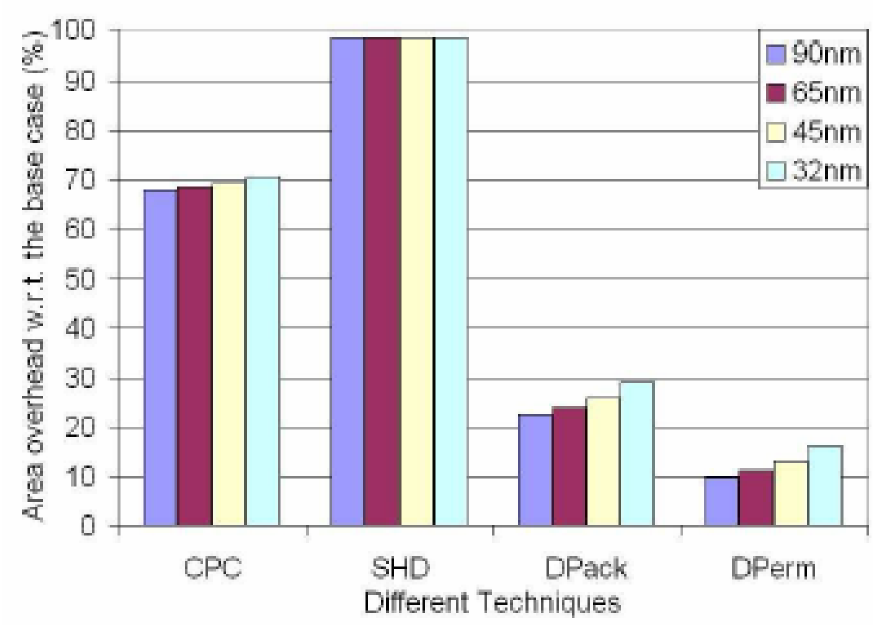
Experimental Results

- Codec Overhead

Method	# of wires	Codec overhead			
		Area (μm^2)	Delay (ps)	Effective delay (ps)	Energy (pJ)
Base	32	0	0	0	0
DPack	38	4395	950	560	2.13
DPerm	34	4361	540	150	1.61
CPC	53	1758	400	200	1.20
SHD	63	0	0	0	0

Experimental Results

- Area overhead over the base case for different process technologies.



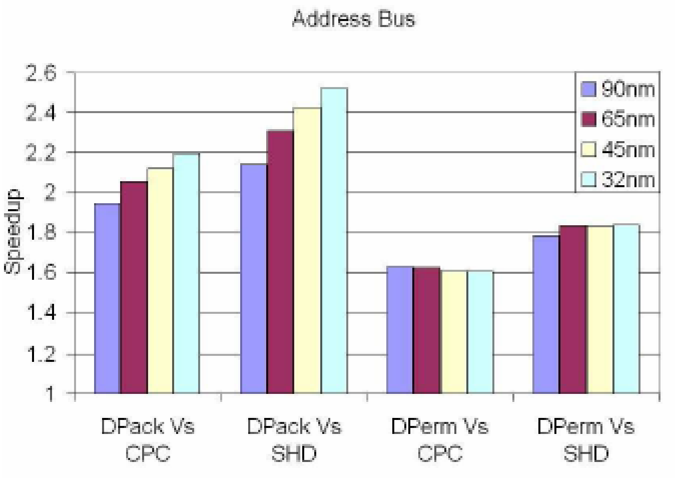
SHD has an overhead of 100% and CPC has 70%.

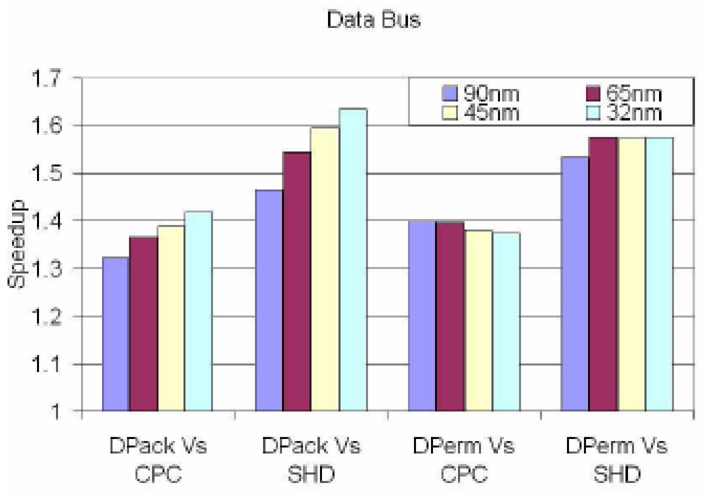
Dpack & Dperm significantly low overhead.

Area overhead increased with shrinking process.

Experimental Results

- Average speedup of our techniques over the CPC and SHD techniques under the area constraint.
  - Two techniques are compared only if they occupy the same area.





Conclusion

- DPack and DPerm techniques achieve more than 2.3x (1.5x) and 1.6x (1.4x) speedup, resp., in the address (data) bus over the unencoded bus.
- Though the CPC and SHD techniques are delay efficient in the data bus case, they require 21 and 31 extra wires, resp.
- Under the area constraint, our techniques outperform both CPC and SHD techniques.

Thank you!
Q & A