Fast Buffered Delay Estimation Considering Process Variations

Tien-Ting Fang and Ting-Chi Wang
National Tsing Hua University
Taiwan

ASPDAC 2007

Outline

- Introduction
  - Impact of Buffer Insertion
  - Effect of Process Variations
- Preliminaries
  - Problem Formulation
  - Delay Model
- Methodology
- Experimental Results
- Conclusion

Outline

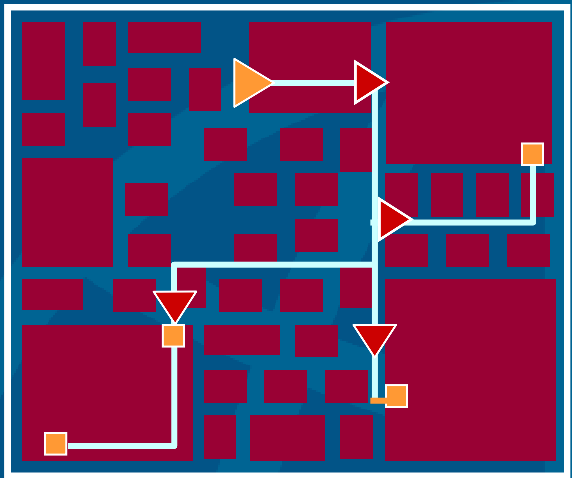
- Introduction
  - Impact of Buffers
  - Effect of Process Variations
- Preliminaries
- Methodology
- Experimental Results
- Conclusion

Impact of Buffers

- Buffer insertion is an essential technique for interconnect optimization.
- At 65nm process technology, 35% of the cells on a chip will be buffers. [1]
- One must be able to assess the impact of buffer insertion in earlier stages, such as floorplanning.
  - e.g., fast estimate the timing cost for a net

[1] P. Saxena, N. Menezes, P. Cocchini, and D. A. Kirkpatrick, "Repeater scaling and its impact on CAD," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 4, pp. 451-463, Apr. 2004.

Impact of Buffers 2

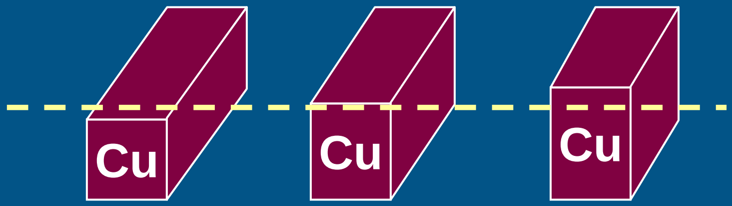


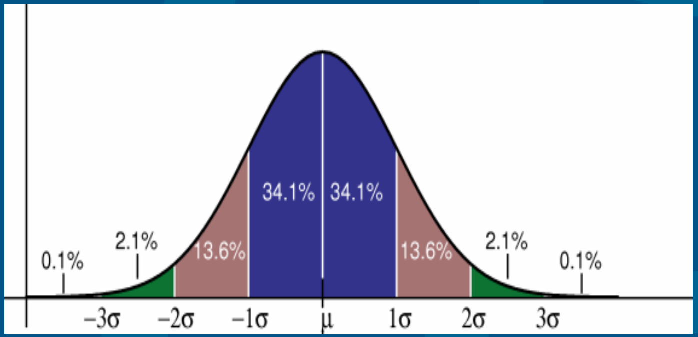
- A linear-time algorithm [2] was proposed to predict interconnect delay with optimal buffering.
  - Not actually perform buffer insertion
  - Consider the effect of buffer blockages
  - Based on a set of assumptions
  - Within 5% average error
  - 100x faster than van Ginneken’s algorithm

[2] C. J. Alpert, J. Hu, S. S. Sapatnekar, and C. N. Sze, “Accurate estimation of global buffer delay within a floorplan,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1140-1146, Jun. 2006.

Effect of Process Variations

- Technology beyond 90nm exhibits significant variations. [3]



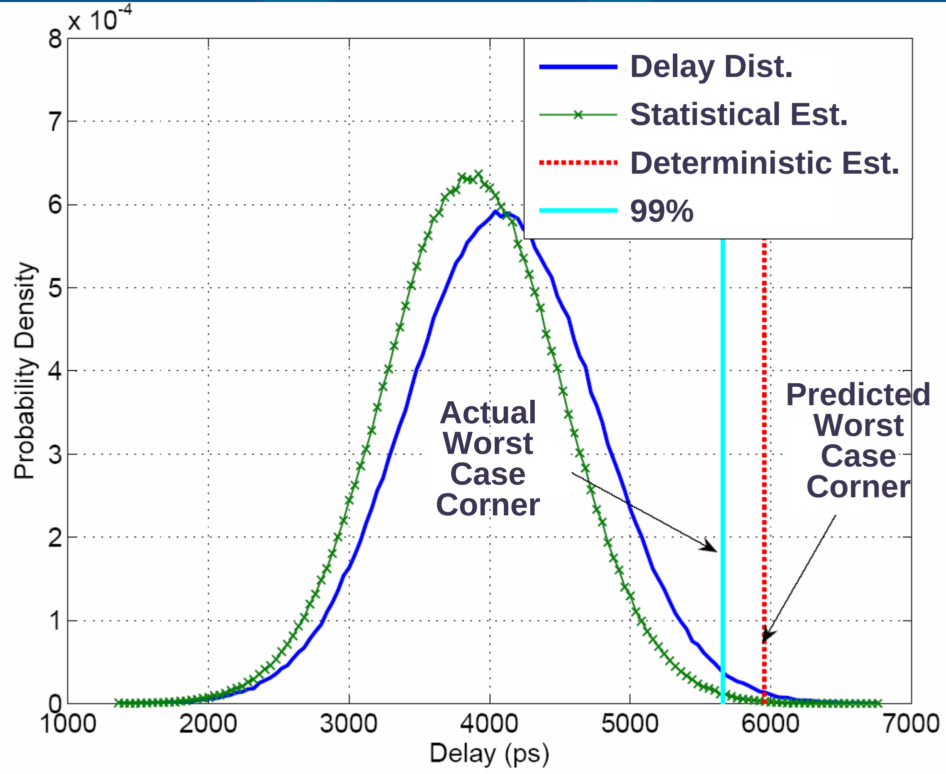


- Traditional analysis and optimization methods under nominal circuit parameters (μ) become too risky.
- Traditional analysis and optimization methods in the worst case corner (i.e., $\mu+3\sigma$) become too pessimistic.

[3] C. Visweswariah, "Death, taxes and failing chips," in *Proc. Design Automation Conf.*, pp. 343-347, 2003.

Effect of Process Variations [2]

- The delay estimated by a deterministic buffered delay estimation (*DBDE*) method in worst case corner (in red line) exceeds the actual worst case corner (in blue line)
  - force a designer to rollback design
  - but there is 99% probability to satisfy the given constraint
  - over-pessimistic!



Effect of Process Variations [3]

- In recent technology generations, variability was dominated by the Back-End-of-the-Line (BEOL) or interconnect metallization [3]
  - The number of cases or corners grows tremendously.
  - Traditional corner-based optimization are not applicable nowadays.

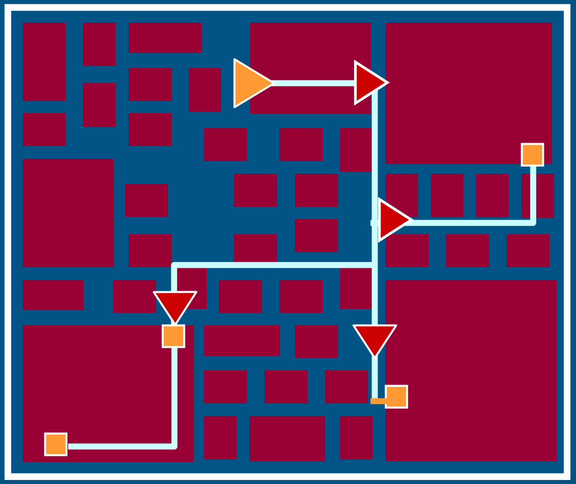
[3] C. Visweswariah, "Death, taxes and failing chips," in *Proc. Design Automation Conf.*, pp. 343-347, 2003.

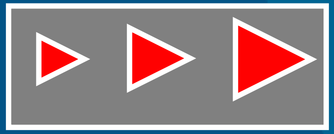
Outline

- Introduction
- Preliminaries
  - Problem Formulation
  - Delay Model
- Methodology
- Experimental Results
- Conclusion

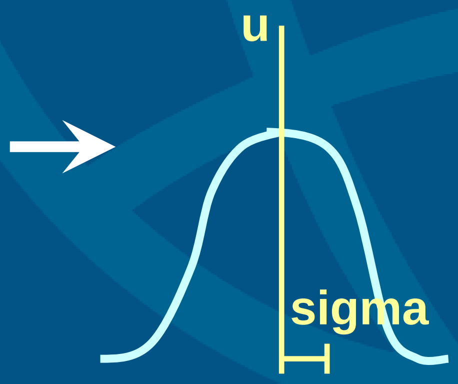
Problem Formulation

- Input
  - A routed topology of a net
  - A set of buffer blockages
  - A buffer library
  - Circuit parameters with variations
- Output
  - Fast statistical timing estimation on worst path delay among all paths from the driver to receivers



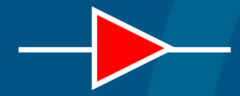


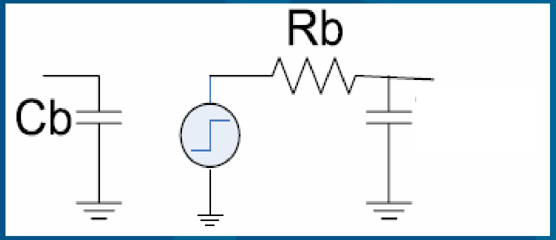
$C_w(u,sigma)$
$R_w(u,sigma)$
$C_b(u,sigma)$
$R_b(u,sigma)$
$D_b(u,sigma)$



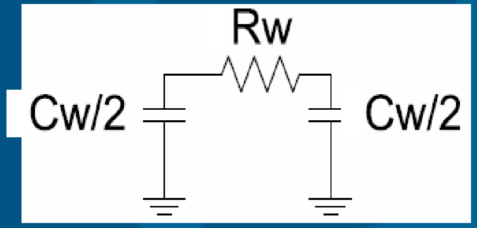
Delay Model

- Delay model for buffer
  - input capacitance C_b
  - output resistance R_b
  - intrinsic delay D_b





- π -model for interconnect
  - wire capacitance per unit length C_w
  - wire resistance per unit length R_w

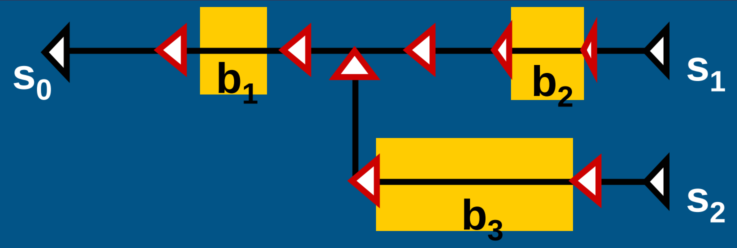


- Elmore delay model for delay computation
  - Is a linear function of wire length while optimal buffering
  - Is quadratic to wire length without buffer insertion

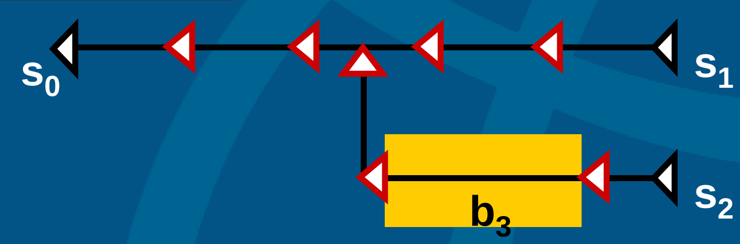
Outline

- Introduction
- Preliminaries
- Methodology
  - **D**eterministic **B**uffered **D**elay **E**stimation (***DBDE***)
    - Four Major Assumptions
    - Delay Calculation
  - **S**tatistical **B**uffered **D**elay **E**stimation (***SBDE***)
    - Process Variation Modeling
    - Key Operations for Variation Awareness
- Experimental Results
- Conclusion

DBDE [1]



a) Without assumptions

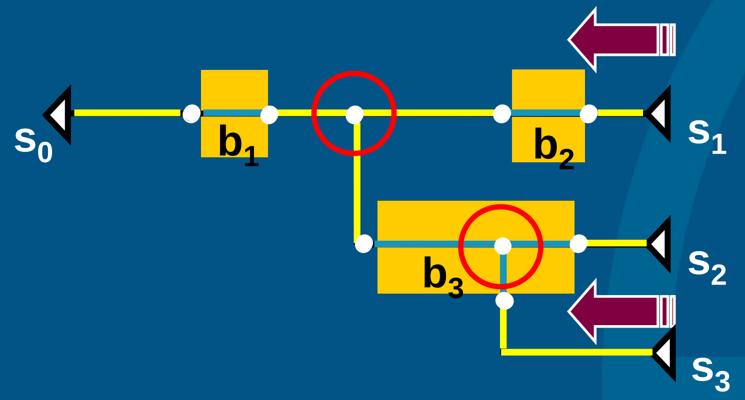


b) With assumptions

- Four assumptions are applied to simplify and accelerate the estimation.
  - Single buffer type
  - Infinitesimal decoupling buffers
  - Small buffer blockages ignored (e.g., b1 and b2)
  - Larger buffer blockage front-and-back buffering (e.g., b3)

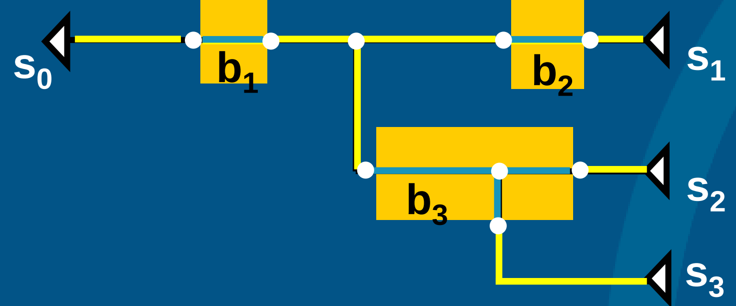
[2] C. J. Alpert, J. Hu, S. S. Sapatnekar, and C. N. Sze, “Accurate estimation of global buffer delay within a floorplan,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1140-1146, Jun. 2006.

DBDE 2



- Delay is accumulated in a single bottom-up tree traversal by decomposing edges as out-blockage edges and in-blockage edges.
- When reaching the merge point, *DBDE* picks the largest accumulated delay and propagates it.

DBDE 3



- Delay calculation:
  - ignore small blockages : $L < L_{opt} = \sqrt{\frac{2(R_b C_b + D_b)}{R_w C_w}}$
  - the delay of out-blockage edge is linear
    - $D(e) = L_e (R_w C_b + R_b C_w + \sqrt{2 R_w C_w (R_b C_b + D_b)})$
  - the delay of in-blockage edge is quadratic
    - $D(e) = R_w L_e \left(\frac{1}{2} C_b L_e + C_b \right) + R_b (C_b L_e + C_b)$

How to extend these formulas with variations ?

SBDE – Process Variation Modeling

- We represent all random variables in a first-order canonical form : $$A = A_0 + \sum_{i=1}^{k} a_i X_i = A_0 + \alpha^T X$$
  - k : number of variation sources
  - X_i: the ith variation source (e.g., inter-/intra-die variation)
  - a_i: sensitivity with respect to X_i
- Circuit parameters with variations are represented in the canonical form:

$$R_w = R_{w0} + \gamma_w{}^T X \qquad C_w = C_{w0} + \varepsilon_w{}^T X$$

$$R_b = R_{b0} + \gamma_b{}^T X \qquad C_b = C_{b0} + \varepsilon_b{}^T X \qquad D_b = D_{b0} + \eta_b{}^T X$$

- We assume all X_i are in the standard Gaussian distribution ~N(0,1) and mutually independent.

SBDE – Key Operations

- Associate each node *v* with (*d*(*v*), *c*(*v*))
  - Accumulated worst delay : *d(v)*
  - Downstream loading capacitance : *c(v)*
- Represent the *d*(*u*) and *c*(*u*) of the child node *u* of node *v* in the first-order canonical forms:
  - $$d(u) = d_{u0} + \alpha_u^T X$$
    $$c(u) = c_{u0} + \beta_u^T X$$
- Derive *d(v)* and *c(v)* from node *u* by edge *e*:
  - If *e* is an out-blockage edge

    $$d(v) = d(u) + L_e(R_w C_b + R_b C_w + \sqrt{2R_w C_w (R_b C_b + D_b)})$$
  - If *e* is an in-blockage edge

    $$d(v) = d(u) + R_w L_e (C_w L_e / 2 + c(u))$$

SBDE – Key Operations 2

- Quadratic delay calculation
  - $$d(v) = d_{u0} + \frac{1}{2} l(u,v)^2 R_{w0} C_{w0} + l(u,v) R_{w0} c_{u0}$$
    $$+ \left[\alpha_u + \frac{1}{2} l(u,v)^2 (R_{w0}\varepsilon_w + C_{w0}\gamma_w) + l(u,v)(R_{w0}\beta_u + c_{u0}\gamma_w) \right]^T X$$
    $$+ X^T \left[\frac{1}{2} l(u,v)^2 \gamma_w \varepsilon_w^T + l(u,v) \gamma_w \beta_u^T \right] X$$
    $$= d_0 + \lambda^T X + X^T \Omega X$$

Not in the first-order canonical form

  - a) calculate the first and the second moments of $d(v)$
    - $$E(d(v)) = d_0 + \lambda^T \cdot E(X) + E(X^T \Omega X) = d_0 + tr(\Omega)$$
    - $$E(d(v)^2) = d_0^2 + E(X^T \lambda \lambda^T X) + E((X^T \Omega X)^2) + 2d_0 \lambda^T E(X)$$
      $$+ 2E(X^T \Omega X \lambda^T X) + 2d_0 E(X^T \Omega X)$$
      $$= (d_0^2 + tr(\Omega))^2 + \lambda^T \lambda + 2tr(\Omega^2)$$

SBDE – Key Operations [3]

$$E(d(v)) = d_0 + \lambda^T \cdot E(X) + E(X^T \Omega X) = d_0 + tr(\Omega)$$

$$E\left(d(v)^2\right) = d_0^2 + E\left(X^T \lambda \lambda^T X\right) + E\left(\left(X^T \Omega X\right)^2\right) + 2d_0 \lambda^T E(X)$$
$$+ 2E\left(X^T \Omega X \lambda^T X\right) + 2d_0 E\left(X^T \Omega X\right)$$
$$= \left(d_0^2 + tr(\Omega)\right)^2 + \lambda^T \lambda + 2tr\left(\Omega^2\right)$$

– b) calculate the mean and variance by the first and the second moments

- $$\mu(d(v)) = E(d(v)) = d_0 + tr(\Omega)$$
  $$\sigma(d(v))^2 = E\left(d(v)^2\right) - E(d(v))^2 = \lambda^T \lambda + 2tr\left(\Omega^2\right)$$

– c) approximate *d*(*v*) to the first-order canonical form by matching the mean and variance

- $$d(v) \approx \left(d_0 + tr(\Omega)\right) + \sqrt{1 + \frac{2tr\left(\Omega^2\right)}{\lambda^T \lambda}} \lambda^T X$$

SBDE – Key Operations [4]

- Linear delay calculation
  - $$d(v) = d_{u0} + L_e\left(R_{w0}C_{b0} + R_{b0}C_{w0}\right) + \left(\alpha_u + L_e\left(R_{w0}\varepsilon_b + C_{w0}\gamma_b + R_{b0}\varepsilon_w + C_{b0}\gamma_w\right)\right)^T X + X^T L_e\left(\gamma_w \varepsilon_b{}^T + \gamma_b \varepsilon_w{}^T\right) X + L_e f(X)$$

Not in the first-order canonical form

where $f(X) = \sqrt{A + BX + CX^2 + DX^3 + EX^4}$

Square-root ?!

$$A = 2R_{w0}C_{w0}\left(R_{b0}C_{b0} + D_{b0}\right)$$

$$B = 2\left[R_{w0}C_{w0}R_{b0}C_{b0}\left(\frac{\gamma_w}{R_{w0}} + \frac{\varepsilon_w}{C_{w0}} + \frac{\gamma_b}{R_{b0}} + \frac{\varepsilon_b}{C_{b0}}\right) + R_{w0}C_{w0}D_{b0}\left(\frac{\gamma_w}{R_{w0}} + \frac{\varepsilon_w}{C_{w0}} + \frac{\eta_b}{D_{b0}}\right)\right]$$

$$C = 2\left[\begin{array}{l} R_{w0}C_{w0}R_{b0}C_{b0}\left(\frac{\gamma_w\varepsilon_w}{R_{w0}C_{w0}} + \frac{\gamma_w\gamma_b}{R_{w0}R_{b0}} + \frac{\gamma_w\varepsilon_b}{R_{w0}C_{b0}} + \frac{\varepsilon_w\gamma_b}{C_{w0}R_{b0}} + \frac{\varepsilon_w\varepsilon_b}{C_{w0}C_{b0}} + \frac{\gamma_b\varepsilon_b}{R_{b0}C_{b0}}\right) \\ + R_{w0}C_{w0}D_{b0}\left(\frac{\gamma_w\varepsilon_w}{R_{w0}C_{w0}} + \frac{\gamma_w\eta_b}{R_{w0}D_{b0}} + \frac{\varepsilon_w\eta_b}{C_{w0}D_{b0}}\right) \end{array}\right]$$

$$D = 2\left[\begin{array}{l} R_{w0}C_{w0}R_{b0}C_{b0}\left(\frac{\gamma_w\varepsilon_w\gamma_b}{R_{w0}C_{w0}R_{b0}} + \frac{\gamma_w\varepsilon_w\varepsilon_b}{R_{w0}C_{w0}C_{b0}} + \frac{\gamma_w\gamma_b\varepsilon_b}{R_{w0}R_{b0}C_{b0}} + \frac{\varepsilon_w\gamma_b\varepsilon_b}{C_{w0}R_{b0}C_{b0}}\right) \\ + \gamma_w\varepsilon_w\eta_b \end{array}\right]$$

$$E = 2\gamma_w\varepsilon_w\gamma_b\varepsilon_b$$

SBDE – Key Operations [5]

- a) apply the Taylor series expansion on $f(x)$ with respect to X=0 and truncate it until the second order
  - $$f(X) \approx \sqrt{A} + \left(\frac{B}{2\sqrt{A}}\right)^T X + X^T \left(\frac{C}{2\sqrt{A}} - \frac{B^2}{8\sqrt{A^3}}\right) X$$
- b) then use the same matching technique to approximate $d(v)$ to the first-order canonical form.
  - $$d(v) \approx \left(d_0 + tr(\Omega)\right) + \sqrt{1 + \frac{2tr\left(\Omega^2\right)}{\lambda^T \lambda}} \lambda^T X$$

$$d_{v0} = R_{w0} C_{b0} + R_{b0} C_{w0} + \sqrt{A}$$

$$\lambda = R_{w0} \varepsilon_b + C_{w0} \gamma_b + R_{b0} \varepsilon_w + C_{b0} \gamma_w + \frac{B}{2\sqrt{A}}$$

$$\Omega = \gamma_w \varepsilon_b{}^T + \gamma_b \varepsilon_w{}^T + \frac{C}{2\sqrt{A}} - \frac{B^2}{8\sqrt{A^3}}$$

SBDE – Key Operations [6]

- Maximum delay determination
  - given two random variables in first-order canonical form
    - $A = A_0 + \sigma_A{}^T X$
    - $B = B_0 + \sigma_B{}^T X$
  - first calculate the tightness probability $T_{A,B}$ (the probability of A larger than B) and $T_{B,A}$ according to [4]
    - $$T_{A,B} = \Phi\left(\frac{A_0 - B_0}{\theta}\right), \; T_{B,A} = \Phi\left(\frac{B_0 - A_0}{\theta}\right) = 1 - T_{A,B}$$
      $$\text{where } \theta = \sqrt{\sigma_A{}^2 + \sigma_B{}^2 - 2\rho\sigma_A\sigma_B}$$
  - then compute the mean and variance via the moment generating function provided in [5]
    - $$\max(A,B) = T_{A,B}A_0 + T_{B,A}B_0 + \theta\,\phi\left(\frac{A_0 - B_0}{\theta}\right) + \left(T_{A,B}\sigma_A + T_{B,A}\sigma_B\right)^T X$$

[4] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, “Frist-order incremental block-based statistical timing analysis,” in *Proc. Design Automation Conf.*, pp. 331-336, 2004.
[5] M. Cain, “The moment-generating function of the minimum of bivariate normal random variables,” in *The American Statistician*, vol. 48, May 1994.

Outline

- Introduction
- Preliminaries
- Methodology
- Experimental Results
- Conclusion

Experimental Results

- We implemented the following three algorithms in C++ on Linux x86_64 machine with 2G Processor/4GB RAM.
  - **D**eterministic **B**uffered **D**elay **E**stimation (***DBDE***) [1]
  - **S**tatistical **B**uffered **D**elay **E**stimation (***SBDE***)
  - **S**tatistical **B**uffer **I**nsertion (***SBI***) [6]
- We used the largest buffer in our estimation and forced the parameters of drivers and receivers to be equal to the parameters of the buffer we chose.

[1] C. J. Alpert, J. Hu, S. S. Sapatnekar, and C. N. Sze, “Accurate estimation of global buffer delay within a floorplan,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1140-1146, Jun. 2006.

[6] J. Xiung, and L. He, “Fast buffer insertion considering process variations,” in *Proc. Intl. Symp. on Physical Design*, pp. 128-135, 2006.

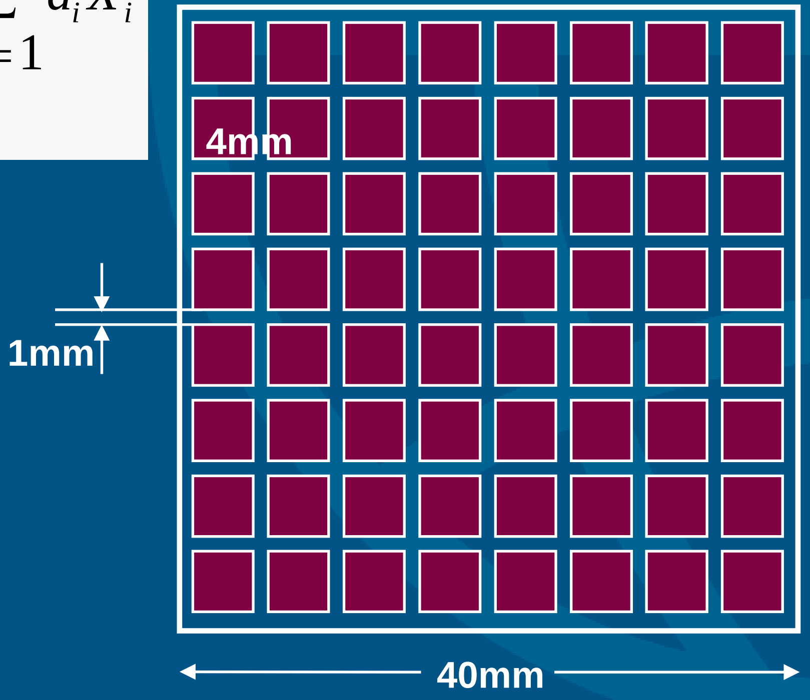
Experimental Results 2

- For each circuit parameters, the sensitivity a_i to each variation source X_i is set to 5% of its nominal value A_0.
  - e.g.,

$$A = A_0 + \alpha^T X = A_0 + \sum_{i=1}^{k} a_i X_i$$

$$\text{where } a_i = 0.05 A_0$$

- A same blockage configuration was applied on each test case.



Experimental Results [3]

net	DBDE		SBDE			SBI				SBDE vs. SBI		
	delay	runtime	delay	sd	runtime	delay	sd	#buf	runtime	delay	sd	runtime
r3	1550.75	0.0009	1560.02	246.43	0.0024	1605.33	255.05	12	0.3644	97.18%	96.62%	149x
r4	1771.43	0.0013	1782.11	282.17	0.0039	1837.98	292.77	16	0.4037	96.96%	96.38%	104x
r5	1497.81	0.0013	1507.15	241.56	0.0039	1603.84	260.84	13	0.2901	93.97%	92.61%	75x
r6	1627.52	0.0014	1637.19	258.88	0.0037	1667.90	265.55	20	0.4652	98.16%	97.49%	127x
r7	1632.03	0.0016	1641.93	261.30	0.0051	1725.14	278.69	18	0.4014	95.18%	93.76%	79x
r8	1946.99	0.0019	1958.55	310.58	0.0057	2040.27	328.48	23	0.4943	95.99%	94.55%	87x
r9	1561.62	0.0020	1570.75	247.17	0.0061	1658.93	267.96	21	0.5266	94.68%	92.24%	86x
r10	1745.86	0.0021	1756.23	278.64	0.0063	1880.92	304.91	25	0.5247	93.37%	91.38%	83x
										95.69%	94.38%	99x

- In comparison with SBI:
  - the average mean delay error of SBDE is 4.31%
  - the error of standard deviation is within 5.62% on the average
  - the speedup is 99x on the average

net	#sink	wirelength	%blk
r3	3	28849	55.50
r4	4	36727	66.42
r5	5	33210	72.53
r6	6	39967	48.49
r7	7	42645	62.72
r8	8	51207	59.29
r9	9	49267	62.93
r10	10	57807	65.30

Experimental Results 4

net	DBDE		SBDE			SBI				SBDE vs. SBI		
	delay	runtime	delay	sd	runtime	delay	sd	#buf	runtime	delay	sd	runtime
mcu0	1149.70	0.0020	1157.39	190.32	0.0070	1165.96	190.98	21	0.7889	99.26%	99.65%	113x
mcu1	1849.71	0.0020	1861.91	304.38	0.0080	1917.39	311.12	18	0.3329	97.11%	97.83%	42x
n107	639.99	0.0010	643.36	102.66	0.0060	720.07	131.80	13	0.5789	89.35%	77.89%	96x
n189	1727.20	0.0030	1738.50	285.20	0.0110	1768.46	316.06	30	0.5858	98.31%	90.24%	53x
n313	1801.14	0.0030	1812.44	293.51	0.0080	2034.04	367.97	22	0.4369	89.11%	79.77%	55x
n786	3842.54	0.0050	3867.33	628.67	0.0140	4082.36	677.60	20	0.3050	94.73%	92.78%	22x
n869	3137.13	0.0030	3156.68	506.37	0.0090	3271.70	528.52	15	0.2440	96.48%	95.81%	27x
n873	1418.88	0.0020	1427.59	227.48	0.0080	1455.53	262.77	25	0.4456	98.08%	86.57%	56x
poi3	3516.91	0.0030	3537.35	554.27	0.0090	3578.88	561.88	39	0.0905	98.84%	98.64%	10x
										95.70%	91.02%	53x

- In comparison with *SBI*:
  - the average mean delay error of *SBDE* is 4.3%
  - the error of standard deviation is within 8.98% on the average
  - the speedup is 53x on the average

net	#sink	wirelength	%blk
mcu0	18	39920	65.53
mcu1	19	41380	78.13
n107	17	12790	41.75
n189	29	58100	59.86
n313	19	55840	82.65
n786	32	54520	83.64
n869	21	43270	81.42
n873	20	49720	37.05
poi3	20	63960	46.75

Results Summary

- In the presence of process variations, *SBDE* tightens the lower bound and gives a more accurate estimation than *DBDE* can do.
- *SBDE* can achieve 10x~149x faster than *SBI* while only 2.5x~6x slower than *DBDE*.

Outline

- Introduction
- Preliminaries
- Methodology
- Experimental Results
- Conclusion

Conclusion

- We propose a statistical buffered delay estimation method which considers the effect of process variations and the presence of buffer blockages.
- We show that the deterministic buffered delay estimation using the worst case corner, i.e., $\mu+3\sigma$, will be over-pessimistic.
- The experimental results show the efficiency and accuracy of our statistical estimation technique.
- Useful for earlier stages such as floorplanning

Q & A

Thank You