

Mónica Jorge Carvalho Figueiredo

Sincronização em Sistemas Integrados a Alta Velocidade

Synchronisation in High-Performance Integrated Circuits



Mónica Jorge Carvalho Figueiredo

Sincronização em Sistemas Integrados a Alta Velocidade

Synchronisation in High-Performance Integrated Circuits

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Doutor em Engenharia Electrotcnica, realizada sob a orientação científica do Doutor Rui Luís Andrade Aguiar, Professor Associado com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Esta tese foi realizada com o apoio financeiro da Fundação para a Ciência e Tecnologia, sob a forma de uma bolsa de Doutoramento (BD/30654/2006).

agradecimentos

Muitas pessoas, de um ou outro modo, deram contribuições úteis para o progresso da investigação apresentada nesta dissertação, muito embora apenas me seja possível mencionar algumas dessas contribuições.

Primeiramente quero manifestar os meus agradecimentos ao meu orientador, o Professor Doutor Rui Luís Andrade Aguiar, não só por todo o seu apoio na elaboração desta tese com a sua valiosa orientação e crítica científica, mas também pela sua amizade e orientação no meu percurso académico na Universidade de Aveiro. Merece também um agradecimento especial o Doutor Luís Nero Alves, pela sua amizade, pela constante disponibilidade e por toda a contribuição que deu ao longo da pesquisa e na revisão deste documento.

Quero também prestar o meu reconhecimento à Fundação para a Ciência e Tecnologia, que me proporcionou o apoio financeiro para a realização desta tese sob a forma de uma bolsa de Doutoramento (BD/30654/2006), bem como às instituições de acolhimento onde este trabalho foi desenvolvido, a Universidade de Aveiro e o Instituto de Telecomunicações, por todos os recursos disponibilizados.

Finalmente, mas no menos importante, um agradecimento especial aos meus amigos, à minha família, e em especial aos meus pais e ao Carlos, por todo o apoio, carinho e compreensão ao longo destes anos.

o júri / the jury

presidente / president

Doutor Fernando Joaquim Fernandes Tavares Rocha Professor Catedrático do Departamento de Geociências da Universidade de Aveiro

Doutor José Alfredo Ribeiro da Silva Matos Professor Catedrático da Faculdade de Engenharia da Universidade do Porto

Doutor Jorge Filipe Leal Costa Semião

Professor Adjunto do Instituto Superior de Engenharia da Universidade do Algarve

Doutor Luis Filipe Mesquita Nero Moreira Alves

Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Doutor Rui Luís Andrade Aguiar

Professor Associado com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

palavras-chave

resumo

Sincronização, Incerteza temporal, Ruído, Circuitos integrados.

A distribuição de um sinal relógio, com elevada precisão espacial (baixo *skew*) e temporal (baixo *jitter*), em sistemas síncronos de alta velocidade tem-se revelado uma tarefa cada vez mais demorada e complexa devido ao escalonamento da tecnologia. Com a diminuição das dimensões dos dispositivos e a integração crescente de mais funcionalidades nos Circuitos Integrados (CIs), a precisão associada às transições do sinal de relógio tem sido cada vez mais afectada por variações de processo, tensão e temperatura. Esta tese aborda o problema da incerteza de relógio em CIs de alta velocidade, com o objetivo de determinar os limites do paradigma de desenho síncrono.

Na prossecução deste objectivo principal, esta tese propõe quatro novos modelos de incerteza com âmbitos de aplicação diferentes. O primeiro modelo permite estimar a incerteza introduzida por um inversor estático CMOS, com base em parâmetros simples e suficientemente genéricos para que possa ser usado na previsão das limitações temporais de circuitos mais complexos, mesmo na fase inicial do projeto. O segundo modelo, permite estimar a incerteza em repetidores com ligações RC e assim otimizar o dimensionamento da rede de distribuição de relógio, com baixo esforço computacional. O terceiro modelo permite estimar a acumulação de incerteza em cascatas de repetidores. Uma vez que este modelo tem em consideração a correlação entre fontes de ruído, é especialmente útil para promover técnicas de distribuição de relógio e de alimentação que possam minimizar a acumulação de incerteza. O quarto modelo permite estimar a incerteza temporal em sistemas com múltiplos domínios de sincronismo. Este modelo pode ser facilmente incorporado numa ferramenta automática para determinar a melhor topologia para uma determinada aplicação ou para avaliar a tolerância do sistema ao ruído de alimentação.

Finalmente, usando os modelos propostos, são discutidas as tendências da precisão de relógio. Conclui-se que os limites da precisão do relógio são, em última análise, impostos por fontes de variação dinâmica que se preveem crescentes na actual lógica de escalonamento dos dispositivos. Assim sendo, esta tese defende a procura de soluções em outros níveis de abstração, que não apenas o nível físico, que possam contribuir para o aumento de de-sempenho dos CIs e que tenham um menor impacto nos pressupostos do paradigma de desenho síncrono.

Synchronisation, Time uncertainty, Noise, Integrated circuits.

Distributing a the clock simultaneously everywhere (low skew) and periodically everywhere (low jitter) in high-performance Integrated Circuits (ICs) has become an increasingly difficult and time-consuming task, due to technology scaling. As transistor dimensions shrink and more functionality is packed into an IC, clock precision becomes increasingly affected by Process, Voltage and Temperature (PVT) variations. This thesis addresses the problem of clock uncertainty in high-performance ICs, in order to determine the limits of the synchronous design paradigm.

In pursuit of this main goal, this thesis proposes four new uncertainty models, with different underlying principles and scopes. The first model targets uncertainty in static CMOS inverters. The main advantage of this model is that it depends only on parameters that can easily be obtained. Thus, it can provide information on upcoming constraints very early in the design stage. The second model addresses uncertainty in repeaters with RC interconnects, allowing the designer to optimise the repeater's size and spacing, for a given uncertainty budget, with low computational effort. The third model, can be used to predict jitter accumulation in cascaded repeaters, like clock trees or delay lines. Because it takes into consideration correlations among variability sources, it can also be useful to promote floorplan-based power and clock distribution design in order to minimise jitter accumulation. A fourth model is proposed to analyse uncertainty in systems with multiple synchronous domains. It can be easily incorporated in an automatic tool to determine the best topology for a given application or to evaluate the system's tolerance to power-supply noise.

Finally, using the proposed models, this thesis discusses clock precision trends. Results show that limits in clock precision are ultimately imposed by dynamic uncertainty, which is expected to continue increasing with technology scaling. Therefore, it advocates the search for solutions at other abstraction levels, and not only at the physical level, that may increase system performance with a smaller impact on the assumptions behind the synchronous design paradigm.

keywords

abstract

Contents

Li	List of Figures iv				
Li	st of [Tables	ix		
A	bbrev	iations and Acronyms	xiii		
Li	st of]	Frequent Symbols	xvi		
1	Intr	oduction	1		
	1.1	Motivation	1		
	1.2	Goals	4		
	1.3	Original Contributions	5		
	1.4	Software Support	8		
	1.5	Thesis Organisation	9		
2	Tim	ing in Synchronous Systems	11		
	2.1	The Synchronous Paradigm	11		
		2.1.1 Synchronous Operation	12		
		2.1.2 Clock Uncertainty	14		
	2.2	Sources of Clock Uncertainty	16		
		2.2.1 Intrinsic Variations	17		
		2.2.2 Environmental Variations	19		
	2.3	Timing Analysis	25		
		2.3.1 Timing Models	25		
		2.3.2 Jitter Models	29		
		2.3.3 Simulation Tools	33		
	2.4	Clocking Systems	35		
		2.4.1 Clock Generation	35		
		2.4.2 Clock Distribution	36		
	2.5	Final Remarks	42		
3	Unc	ertainty in Clock Repeaters	45		
	3.1	Clock Repeaters	45		
		3.1.1 Static and Tunable Delay Repeaters	46		
		3.1.2 Uncertainty in Basic Inverters	50		
		3.1.3 Performance Comparison	56		

	3.2	Reference Inve	rter Jitter Model	60
		3.2.1 Circuit	Parameters	60
		3.2.2 Intrinsi	c Variability Sources	64
		3.2.3 Enviror	mental Variability Sources	67
	3.3	Scalable Jitter M	Model	72
		3.3.1 Equival	lent Circuit Model	
		3.3.2 litter M	odel for Symmetric Repeaters	
		3.3.3 Litter M	odel for Asymmetric Repeaters	83
		334 Model I	Fvaluation	
	34	Conclusions		
	0.1	conclusions .		00
4	Unc	ertainty in Cloc	king Structures	89
	4.1	Delay Lines an	d Clock Trees	89
		4.1.1 Digitall	y Controlled Delay Lines	90
		4.1.2 Clock D	Distribution Trees	92
		4.1.3 Perform	nance Analysis	
	4.2	Jitter Accumula	ation Model	99
		4.2.1 Dvnam	ic litter in Cascaded Repeaters	100
		4.2.2 Bounds	for litter Accumulation	102
		4.2.3 Simulat	tion Results	104
	4.3	Clock Deskewi	ing Systems	107
	1.0	4.3.1 Deskew	ving Uncertainty Model	107
		4.3.2 Impact	of Circuit Floorplanning	112
		433 Impact	of Synchronisation Topologies	115
		434 Compa	rativo Analycic	121
	44	Conclusions		121
	1.1	conclusions .		••• 121
5	Exp	erimental Resul	lts	127
	5.1	Experimental S	Setup	127
		5.1.1 Hardwa	are and Equipment	127
		5.1.2 Supply	Noise Generator	130
	5.2	Uncertainty an	d Jitter Evaluation	133
		5.2.1 Uncerta	ainty in SDRs	134
		5.2.2 PSN Jitte	er Evaluation	136
		5.2.3 CRT litte	er Evaluation	140
	5.3	Scalable litter N	Model Validation	142
		5.3.1 litter In	sertion	142
		5.3.2 litter Ad	ccumulation	144
	5.4	Conclusions .		147
6	Lim	its and Trends i	n Synchronous Clocking	149
	6.1	Clock Repeater	rs	149
		6.1.1 Scaling	and Circuit Parameters	150
		6.1.2 Trends	in Jitter Sensitivity	152
	6.2	Clocking Struc	tures	157
		6.2.1 Člock T	rees	157

	6.2.2 Trends in Jitter Accumulation	.62
6.	3 Discussion	.66
	6.3.1 Jitter Trends in Clock Repeaters	.66
	6.3.2 Jitter Trends in Synchronous Systems	.70
	6.3.3 The Synchronous Paradigm	.74
6.	4 Conclusions	.79
7 C	onclusions and Future Directions 1	183
7.	l Conclusions	.83
7.	2 Future Directions	.85
Refe	rences 1	80
nerei		.07

List of Figures

1.1	Scaling trends in: a) transistor intrinsic speed; and b) transistor density, clock speed, power and instruction-level parallelism in Intel CPUs	2
2.1	The synchronous paradigm: a) concept of a finite-state machine; and b) clock signal's timing parameters.	12
2.2	Synchronous operation constraints: a) sequential structure; b) setup time constraint; and c) hold time constraint.	13
2.3	Skew and jitter definitions, as components of clock uncertainty: a) clock distribution network with two clock paths; and b) absolute jitter in clock	
	edges vs. skew between different clock signals.	15
2.4	Clock uncertainty as a percentage of cycle time vs. processor clock fre-	
	quency: a) clock skew; and b) clock jitter.	16
2.5	Partition of process variation in inter-die and intra-die variations	17
2.6	PDN with on-chip, package and board components.	21
2.7	Simplified circuit model for a typical PDN with bump-bond packaging	22
2.8	Electromagnetic coupling in neighbouring interconnects	24
2.9	Loaded gate π -model and its equivalent effective capacitance model	28
2.10	Sample clock distribution for uncertainty accumulation model	32
2.11	Generic block diagrams for the: a) PLL; and b) DLL	36
2.12	Clock distribution for the Itanium microprocessor.	38
2.13	Tree structures: a) H-tree; b) X-tree; c) binary tree; and d) clock mesh or grid.	38
2.14	Deskewing schemes with: a) static tuning during factory test and calibra-	
	tion; and b) dynamic tuning during circuit operation.	40
2.15	Multidomain clock distribution: a) generic GALS; b) Intel TeraFlops MPU; c)	
	Intel dual-core Xeon MPU; and d) AMD quad-core Opteron MPU	42
3.1	Static Delay Repeaters: a) inverter gate; b) NAND gate; c) tapered buffer	47
3.2	Invr and Invf SEC inverters, used as drop-in replacements of a symmetric	
	inverters (Invt), and their output rise/fall times	48
3.3	Digital voltage controlled TDRs: a) CSI; b) VRI; c) SCI type 1 and type 2	49
3.4	Inverter: a) test circuit; and b) circuit to extract C_{in}	51
3.5	PSN jitter in the reference FO4 inverter, for different: a) noise levels ($v_n =$	
	σ_{psn}/V_{dd}); and b) cut-off frequencies ($f_n = T_n$)	53
3.6	a) TCN jitter vs. sample size (N) ; b) PSN jitter vs. sample size (N) ; c) IPV	
	jitter and simulation time vs. MC runs	54

3.7	Jitter in the reference inverter, for different fanouts and: a) PSN, TCN and	
•	IPV sources; and b) CMN, DMN and MMN sources.	55
3.8	Jitter in the FO4 reference inverter, for: a) different operating temperatures;	55
20	Berformer of motive for COL VID. COL and COL venestors with respect to	55
3.9	reformance metrics for CSI, VRI, SCI1 and SCI2 repeaters, with respect to input vector (h_1, h_2) ; a) delay: and b) power consumption; for $f_{12} = 500 \text{MHz}$	50
2 10	Input vector $(v_2v_1v_0)$. a) detay, and b) power consumption, for f_{clk} -booten i.e.	61
3.10 3.11	Inverter's voltage and current waveforms for halanced transitions: Inverter's output voltage and current waveforms for halanced transitions:	01
5.11	a) FO1 invertor: and b) FO4 invertor	61
312	Slow-rate and L_{cc} for the reference 90nm inverter for different a) slow-rate	01
5.12	definitions: b) effective current definitions	62
3 1 3	Different test circuit configurations: a) ideal driver and load: b) realistic	02
5.15	driver and load: c) ideal driver and realistic load: d) realistic driver and	
	ideal load	63
3 14	For the circuits shown in Fig. 3.13 plots show: a) slew-rate for different	00
0.11	definitions: and b) L_{cc} obtained with $SR_{20,000}$ for increasing famouts	63
3 15	Performance metrics in 90nm inverters for different sizes and fanouts: a)	00
0.10	TCN jitter: and b) TCN uncertainty	64
316	Results for the reference inverter: a) TCN (RMS) measured at the output	01
0.10	node for constant input voltages: b) voltage transfer characteristic	65
3.17	Performance metrics in 90nm inverters for different sizes and fanouts: a)	00
0.17	PSN jitter: and b) PSN uncertainty	67
3.18	Effective current: a) L_n compared with the FPT model's effective current	01
0.10	and the one obtained from slew-rate measurement: b) impact of both V_{u_k}	
	and V_{dd} .	68
3.19	Crosstalk induced capacitance variability: a) victim wire with two possible	
	aggressors: b) C_v as a Gaussian variable: c) normalized t_d as a function of	
	normalized C_{τ} .	70
3.20	Impact of CRT on: a) r_d and r_{io} ratios; b) PSN, TCN and IPV jitter	72
3.21	CRC: a) extraction from binary clock tree; and b) its circuit model.	73
3.22	CRC π -model and its correspondent C_{ea} model.	74
3.23	Key time instants for the gate's output waveform $(v_2(t))$	75
3.24	Waveform comparison between the Clock Repeater Cell (CRC) π -model	
	and its equivalent model, for balanced repeaters with: a) $C_{int} = 1.4C_{in}$,	
	$R_{int} = R_{on}, C_L = 2C_{in}$ and $C_{eq} = 4.3C_{in}$; and b) $C_{int} = 2.6C_{in}, R_{int} = 2R_{on},$	
	$C_L = 4C_{in}$ and $C_{eq} = 14.4C_{in}$; with $R_{on} = V_{dd}/2I_{D0}$	77
3.25	Static and dynamic jitter error contour plots, using the C_{eq} model	78
3.26	Clock repeater with jitter sources and its C_{eq} model	80
3.27	Scalable jitter model: a) generation flow; and b) normalised scaling func-	
	tion obtained for the reference inverter in a 90nm technology	80
3.28	Simulation framework to characterise C_{eq} variability	82
3.29	Variability in metal four (M4) and top metal layer (M2_2B)	82
3.30	Jitter error in balanced repeaters as a function of: a) r_r ; and b) and r_c	85
3.31	Model error in repeaters with different designs, as a function of r_{io} : a) static	
	jitter; and b) dynamic jitter.	86

4.1	Uniform DCDLs, built with: a) inverter gates; b) NAND gates.	90
4.2	Binary weighted DCDLs with: a) SCIs; b) SDRs in a parallel-path configuration.	91
4.3	H-tree topology with: a) three stages and uniform wire sizing; b) two	
	stages and geometric wire sizing	93
4.4	Jitter and uncertainty in DCDLs, for increasing delays	97
4.5	Jitter and uncertainty along an inverter-based DCDL (with a 3STI multi-	
	plexer) for increasing: a) PSN level (v_n) ; and b) noise step (T_n) .	98
4.6	Jitter simulation results and statistical model predictions for: c) uncorre-	
	lated TCN sources; b) totally correlated PSN sources; and c) totally corre-	
	lated IPV sources.	99
4.7	Error between traditional statistical accumulation jitter predictions and	
	simulation results, for: a) uncorrelated noise sources; and b) correlated	
	noise sources.	100
4.8	Cascaded CRCs and their output jitter.	100
4.9	Waveforms of a reference CRC line (v_{ri}) and a CRC line affected with PSN in	
	the first cell only (v_{ni}) for: a) CMN; and b) DMN	101
4.10	Waveforms of a reference CRC line (v_{ri}) and a CRC line with the same PSN	100
4 1 1	sources applied to all cells (v_{ni}) for: a) CMN; and b) DMN	102
4.11	Jitter gain for A ($v_n = 5\%$, $r_c = 2$, $r_r = 0$), B ($v_n = 5\%$, $r_c = 10$, $r_r = 1$),	
	$C(v_n = 5\%, r_c = 8, r_r = 0)$ and $D(v_n = 10\%, r_c = 8, r_r = 0)$, for: a)	105
1 1 0	Demonstrated noise sources; and b) correlated noise sources	105
4.12	Dynamic litter model predictions compared to simulation results, for: a)	106
1 1 2	Model accuracy in clock trees with: a) $N = 5$ $r = 4$ and uncorrelated	100
4.15	Power Supply Noise (PSN): b) $N = 5$, $r = 4$ and correlated PSN: c) and d)	
	variable N interconnect parameters wire sizing techniques and chin sizes	
	with uncorrelated PSN	106
4.14	Generic DLL based feedback deskewing circuit	109
4.15	Floorplans for DLL based deskewing systems: a) LDS: and b) RDS.	113
4.16	Parallel synchronization, with: a) centralised SDr: and b) distributed SDr.	 116
4.17	Series synchronization with: a) cascaded hierarchy: b) H-Tree hierarchy.	119
4.18	Mesh synchronization: a) global H-tree; b) local SDs; and c) deskewing units.	120
4.19	Skew and jitter as a percentage of T_{clk} : a) reference scenario; b) higher N_c ;	
	c) higher δ_{\Box} and σ_{\Box} ; and d) higher δ_{\Box} , σ_{\Box} and v_n .	123
5.1	Repeater chain PCB with passive probes: a) schematic; b) photograph	128
5.2	Setup used to measure PSN jitter in different circuit boards	130
5.3	Noise generator built with a Xilinx FPGA, custom DAC and daughter boards.	131
5.4	Printed Circuit Boards (PCBs) for a) custom DAC board; and b) custom	
	daughter board.	132
5.5	a) Repeater's supply network with noise coupling; b) signal path and sig-	
	nal's transfer function $(H_s(f))$; c) noise path and noise's transfer function	100
	$(H_n(f)).$	132
5.6 5.7	rest circuits: a) ideal driver and ioad; b) realistic driver and ioad.	134 125
5.7	PCBS for InvA evaluation: a) FO1 type B; b) FO4 type B; and c) FOn type A.	133

5.8	Circuit type A metrics for Little Logic gates: a) jitter; and b) uncertainty.	
	Circuit type B metrics are shown with light grey, unconnected icons	136
5.9	Spectral and statistical properties of noise in the repeater's power supply	100
F 10	nodes: a) external sources OFF; b) external sources ON.	138
5.10	Measurement and model results: a) PSN jitter; b) peak and effective currents	.139
5.11	Impact of unbalanced transitions on delay (t_d) , output switching time (t_{out})	140
= 10	and PSN jitter: a) FOI inverter; b) FO6 inverter.	140
5.12	Circuit board to evaluate the impact of CRT: a) schematic; b) photograph.	141
5.13	Crosstalk jitter measurements with $C_g = 8pF$ and: a) $C_c = 8pF$; b) $C_c = 15pF$.141
5.14	Circuit boards to evaluate the equivalent circuit model: a) inverter fol- lowed by an interconnect π -model and load; and b) inverter followed by	1.40
- 4 -	C_{eq}	143
5.15	Comparison between RC inverter and C_{eq} inverter measurements: a) PSN	1 4 0
- 4 4	jitter; and b) delay (t_d) and switching time (t_{swl}) .	143
5.16	Normalised dynamic jitter, delay and switching time, for different r_c and r_{io}	144
5.17	Board with a FO1 inverter line, used to characterise gain functions	145
5.18	Measured gain functions, with $r_c = 1$, $r_r = 0$ and $v_n = 4\% V_{dd}$, for: a)	
	uncorrelated noise sources; and b) correlated noise sources	145
5.19	Binary tree board with three stages, i.e., with 7 cascaded inverters	146
5.20	Jitter measurements in a binary tree, compared to model predictions, for:	
	a) uncorrelated noise sources; and b) correlated noise sources	147
61	Parameters for FO1 inverters implemented with predictive and commer-	
0.1	cial models, normalised to the PTM 180nm inverter	151
62	TCN precision metrics for inverters implemented with predictive and com-	101
0.2	mercial models: a) absolute iitter (σ_t): and b) uncertainty (U_{terr}).	153
6.3	PSN precision metrics for inverters implemented with predictive and com-	100
0.0	mercial models: a) absolute iitter (σ_t .); and b) uncertainty (U_{nen})	154
6.4	Normalised PSN uncertainty (Y_{ner}) scaling trends with increasing: a) noise	101
0.1	standard deviation (σ_{nsn}); and c) cut-off frequencies (f_n).	156
6.5	Performance metrics in FO4 inverters: a) normalised PSN iitter as a function	
0.0	of r_{i_0} ; and b) $r_d = t_d / t_d$ now and r_{i_0} as a function of C_u / u_c .	157
6.6	Scilab simulation framework to evaluate precision in clock trees.	158
6.7	H-tree performance for increasing synchronous domain area (A_{\Box})	162
6.8	Open- and close-loop uncertainty for: a) TCN sources: and b) PSN sources	163
6.9	Ratio between MMN jitter and the sum of CMN and DMN jitter bounds in	100
0.7	cascaded repeaters, and: a) uncorrelated noise sources; and b) correlated	
	noise sources.	164
6.10	Scaling impact on jitter amplification gain for correlated noise sources,	
	and: a) CMN sources; b) DMN sources; and c) MMN sources	165
6.11	Scaling impact on jitter amplification gain for uncorrelated noise sources,	
	and: a) CMN sources; b) DMN sources; and c) MMN sources	165
6.12	Scaling trends considering constant variability sources (scenario A) for: a)	
	absolute jitter; and b) uncertainty	168

6.13	Scaling trends considering increasing variability sources (scenario B) for:	
	a) absolute jitter; and b) uncertainty.	169
6.14	Scaling scenarios with higher clock frequency or more SDs	170
6.15	Deskewing scaling trends in scenarios DA and DB for: a) skew as a per-	
	centage of clock period; and b) jitter as a percentage of clock period	172
6.16	Deskewing scaling trends in scenarios FA and FB for: a) skew as a percent-	
	age of clock period; and b) jitter as a percentage of clock period	174
6.17	The miniaturisation virtuous circle of the semiconductor industry	175
6.18	Y-chart with digital design domains and levels of abstraction.	178

List of Tables

2.1	Clock distribution characteristics of commercial processors.	39
3.1	Transient noise analysis configuration parameters.	52
3.2	SDRs performance metrics with $\sigma_{vsn} = 6\% V_{dd}$.	56
3.3	Transistor sizes in TDRs, following the structures depicted in Fig. 3.3.	58
3.4	TDRs performance metrics with $\sigma_{psn} = 6\% V_{dd}$	58
3.5	Heuristic TCN jitter model error for 90nm inverters.	66
3.6	Heuristic IPV jitter model error for the reference 90nm inverter	66
3.7	Heuristic PSN jitter model error for 90nm inverters	69
4.1	DCDL performance metrics with $\sigma_{psn} = 6\% V_{dd}$.	95
4.2	DCDL jitter and uncertainty variability within the dynamic range	97
4.3	Model for the worst-case static and dynamic deskewing uncertainty	121
4.4	Design parameters and performance metrics for model evaluation	123
5.1	Performance metrics in circuit type B, with $\sigma_{psn} = 6.66\% V_{dd}$.	134
5.2	Performance metrics in circuit type A, with $\sigma_{psn} = 6.66\% V_{dd}$.	136
5.3	Relevant circuit model parameters for the analog inverter	137
5.4	PSN jitter measurements ($\sigma_{t_d,mea}$) and model predictions ($\sigma_{t_d,mod}$)	139
6.1	Scaling factors for key inverter parameters, in different technologies	151
6.2	Reference inverter's TCN jitter model error.	153
6.3	Reference inverter's PSN jitter model error	155
6.4	Linear fitting results for t_d/μ_{t_d} as a function of C_v/μ_c .	157
6.5	H-tree performance for different design options.	160
6.6	Ratio between measured and expected jitter after three inverters	164
6.7	ITRS intermediate interconnect's parameters and capacitances	167
6.8	Scaling factors for model and circuit parameters in scenarios DA and DB.	171
6.9	Scaling factors for model and circuit parameters in scenarios FA and FB.	173
6.10	System drivers in the high-performance circuit segment.	177

Abbreviations and Acronyms

3STI	Three-State Inverter
MPU	Microprocessor Unit
ACDL	Analog Controlled Delay Line
ADE	Analog Design Environment
AOCV	Advanced On-Chip Variation
ATE	Automatic Test Equipment
AUC	Advanced Ultra-low-voltage
AWE	Asymptotic Waveform Evaluation
BSIM	Berkeley Short-channel Insulated-gate field-effect transistor Model
BGA	Ball Grid Array
C4	Controlled Collapse Chip Connect
CAD	Computed Aided Design
CDN	Clock Distribution Network
CMN	Common Mode Noise
CMOS	Complementary Metal Oxide Semiconductor
СР	Cost-Performance
CPU	Central Processing Unit
CRC	Clock Repeater Cell
CRT	Crosstalk
CSI	Current-Starved Inverter
стѕ	Clock Tree Synthesis
DAC	Digital to Analog Converter
DC	Direct Current
DCO	Digitally Controlled Oscillator

DCDL	Digitally Controlled Delay Line
DL	Delay Line
DLL	Delay Locked Loop
DMN	Differential Mode Noise
DPE	Data Processing Engine
DRAM	Dynamic Random Access Memory
DSK	Deskewing
DSM	Deep Sub-Micron
DUT	Device Under Test
DVFS	Dynamic Voltage and Frequency Scaling
FDSOI	Fully Depleted Silicon-On-Insulator
FET	Field Effect Transistor
FIFO	First In First Out
FPGA	Field-Programmable Gate Array
FPT	First Passage Time
GALS	Globally Asynchronous Locally Synchronous
HP	High-Performance
IC	Integrated Circuit
I/O	Input/Output
IP	Intellectual Property
IPV	Intra-die Process Variability
IR	Current Resistor
ITRS	International Technology Road-Map for Semiconductors
LDS	Local Deskewing System
LPF	Low-Pass Filter
LVC	Low-Voltage Complementary Metal Oxide Semiconductor
МС	Monte Carlo
MG	Multi Gate
MMN	Mixed Mode Noise
MOSFET	Metal Oxide Semiconductor Field Effect Transistor

NMOS	N-Channel Metal Oxide Semiconductor
NoC	Network-on-Chip
ocv	On-Chip Variation
PCC	Power-Connectivity-Cost
РСВ	Printed Circuit Board
PDF	Probability Density Function
PD	Phase Detector
PDN	Power Delivery Network
PLL	Phase Locked Loop
PMOS	P-Channel Metal Oxide Semiconductor
PSD	Power Spectral Density
PSN	Power Supply Noise
PST	Post-Silicon Tunable
PTG	Pass-Transistor Gate
РТМ	Predictive Technology Models
Ρ٧Τ	Process, Voltage and Temperature
RC	Resistor and Capacitor
RDS	Remote Deskewing System
RF	Radio Frequency
RLC	Resistor, Inductor and Capacitor
RMS	Root Mean Square
RMSE	Root Mean Square Error
RO	Ring Oscillator
RV	Random Variable
SC	Skew Controller
SCI	Shunt-Capacitor Inverter
SD	Synchronisation Domain
SDR	Static Delay Repeater
SEC	Single-Edge Clock
SMA	Sub-Miniature version A

SoC	System-on-Chip
SOI	Silicon-On-Insulator
SPICE	Simulated Program with Integrated Circuits Emphasis
SSTA	Statistical Static Timing Analysis
STA	Static Timing Analysis
TCN	Thermal Channel Noise
TDR	Tunable Delay Repeater
UTB	Ultra-Thin Body
VCDL	Voltage-Controlled Delay Line
VCO	Voltage Controlled Oscillator
VLSI	Very-Large-Scale Integration
VRI	Variable Resistor Inverter
VRM	Voltage Regulator Module
νтс	Voltage Transfer Characteristic

List of Symbols

T_{clk}	clock period	12
f_{clk}	clock frequency	12
t_r	rise time	12
t _f	fall time	12
ŚR	slew-rate	12
S	skew	14
J	jitter	14
Т	temperature	18
V_{dd}	supply voltage	20
V_{ss}	ground voltage	20
ΔV_{dd}	instantaneous supply voltage noise	22
ΔV_{ss}	instantaneous ground voltage noise	22
R_{int}	interconnect resistance	24
C_{int}	interconnect capacitance	24
L _{int}	interconnect inductance	24
t _{out}	output transition time	25
t _{d,LH}	charging gate delay (low-to-high transition)	25
t _{d,HL}	discharging gate delay (high-to-low transition)	25
t _{in}	input transition time	25
V_{th}	transistor threshold voltage	25
C_L	load capacitance	25
V_{d0}	threshold drain saturation voltage	25
I_{d0}	threshold drain current (drivability)	25
V_{gs}	gate-to-source voltage	25
V_{ds}	drain-to-source voltage	25
Ieff	effective switching current	26
t_d	gate delay	26
<i>C</i> _{eff}	effective capacitance	27
σ_{t}	iitter	29
U_{a}^{d}	delay uncertainty	30
Cat	total coupling capacitance	31
C_{τ}	total victim's capacitance	31
σ_{μ}	crosstalk induced iitter	31
t u _{d,crt}	mean switching time	46
r _{sw} 7	buffor taporing factor	10 16
5 M	NMOS chapped width	±0 //6
V_n	NMOS channel length	±0 46
L_n	PMOS channel length	10 17
	DMOC channel width	17 17
vvp		±/ 17
t_d	propagation delay	±/
п	electrical effort or fanout	±/

β	ratio between PMOS and NMOS channel width	48
C_{in}	input capacitance	51 52
1 n 1)	PSN level	52
σ_n	PSN standard deviation	52
f_{m}	PSN cut-off frequency	52
ј <i>п</i> Т.:	simulation time	53
σ_{t}	TCN jitter	64
Uten	TCN uncertainty	64
Ide	transistor's drain to source current	64
V _{0.tcn}	TCN noise	64
β_{tcn}	TCN jitter sensitivity factor	65
ϵ_{tcn}	TCN jitter model error	65
$\sigma_{t_{d,inv}}$	IPV jítter	66
$\sigma_{I_{v,ivv}}$	peak current IPV variability	66
β_{ivv}	IPV sensitivity	66
ϵ_{inv}	IPV jitter model error	66
σ_{t}	PSN jitter	67
Unon	PSN uncertainty	67
σ_{m}	RMS noise in power rail	67
$\sigma_{v_{dd}}$	RMS noise in ground rail	67
$\sigma_{v_{ss}}$	PSN at the gate's output node	68
$v_{o,psn}$	normalised threshold voltage	60
7 7	current fitting parameter	69
с R	PSN jitter consitivity factor	60
ρ_{psn}		60
ϵ_{psn}	PSN jitter model error	09 70
C_c	total wire coupling capacitance to each neighbour	70
C_{gt}		70
r_{i0}	balance ratio	71
vv _{int}		70
r	resistance ratio	85
t _D	path delay	91
d	DCDL delav step	95
Ynen	normalised PSN uncertainty	98
- psn Ф	post-silicon clock phase	108
Ψ 1/ 1	mean clock phase	108
μφ δι	absolute clock skew	108
σ_{ϕ}	absolute clock jitter	100
U_{ϕ}	nominal forward nath delay	100
$\frac{\iota_f}{-}$		100
l_r	nominal return path delay	100
τ_c	nominal distribution delay in SDC	108
φ_c	clock phase in SDc	108
φ_r	reference clock phase	108
Δ		109
ϵ_{pd}		109
I _{dsk}	deskewing period	109
S_c	absolute skew in SDc	109
\mathcal{O}_{τ_r}		109
Δ_m	DCDL aynamic range	109

$\delta_{ au_f}$	forward path delay skew	109
$\delta_{ au}$	clock distribution delay skew	109
t_{θ}^{c}	phase error coherence time	110
$\check{ heta}$	instantaneous phase error	110
t_L	loop lock-in time	110
$\bar{\sigma_{\Lambda}}$	jitter inserted by the DCDL	111
σ_{τ_f}	jitter inserted in the forward clock path	111
σ_{τ_c}	jitter inserted in the controlled SD	111
ρ^{-1}	correlation parameter	111
σ_{\Box}	chip-wide distribution jitter	111
δ_{\Box}	chip-wide distribution skew	111
$ au_{\Box}$	chip-wide distribution latency	111
A_{\Box}	chip area	111
A_c	area of the controlled SD	111
α_c	controlled SD area ratio	111
\mathcal{U}	deskewing system's uncertainty	112
γ	quasi-static skew percentage	112
Δ_{RDS}	nominal DCDL delay in a RDS	113
$ au_m$	matched line's delay	113
$\delta_{ au_m}$	interconnect path skew	114
δ_l	average clock line skew per unit length	114
σ_l	average clock line jitter per unit length	114
$\sigma_{ au_m}$	interconnect path jitter	114
\mathcal{U}_{cpc}	uncertainty in centralised parallel deskewing	116
$t_{L_{cp}}$	lock-in time in centralised parallel deskewing	116
\mathcal{U}_{dp}	uncertainty in distributed parallel deskewing	118
$t_{L_{dp}}$	lock-in time in distributed parallel deskewing	118
L_s	number of synchronisation levels	118
\mathcal{U}_{cs}	uncertainty in cascaded series deskewing	118
$t_{L_{cs}}$	lock-in time in cascaded series deskewing	118
N_c	number of controlled SDs	119
\mathcal{U}_{ts}	uncertainty in tree series deskewing	119
$t_{L_{ts}}$	lock-in time in tree series deskewing	120
\mathcal{U}_m	uncertainty in mesh deskewing	120

Chapter 1 Introduction

This chapter introduces the motivation behind this thesis, discusses its main goals, and highlights its main contributions. It also provides a list of original publications resulting from this work and describes the thesis organisation.

1.1 Motivation

UTTING edge Very-Large-Scale Integration (VLSI) designs today are built on Complementary Metal Oxide Semiconductor (CMOS) processes below 32nm [1–3] and have multi gigahertz clock frequencies. The challenge of clock distribution in such systems is extremely complex, as clock networks may include thousands of interlinked clock signals that can be amplified, gated, branched and/or merged several times. Ideally, the clock distribution network is supposed to distribute the clock signal simultaneously (no skew) and periodically (no jitter) to all registers. As this is an impossible target in practise, it is common to consider a clock uncertainty budget in every design, usually set bellow 10% of the clock period. Increasing this budget reduces the portion of the clock period that can be used for computation and offsets some of the performance increase offered by technology scaling. Thus, mitigating clock uncertainty is one of the major goals when designing high-performance synchronous systems.

Fig. 1.1 shows the International Technology Road-Map for Semiconductors (ITRS) speed target for the Metal Oxide Semiconductor Field Effect Transistor (MOSFET) intrinsic performance metric ($1/\tau$) and Intel Central Processing Unit (CPU) trends. The ITRS plot includes MOSFET and Ring Oscillator (RO) speed for planar bulk, Silicon-On-Insulator (SOI) and Multi Gate (MG) high-performance logic devices [4]. RO speed (in delay per stage, for

fan-outs of one and four) is slower than the intrinsic transistor speed $(1/\tau)$, but is considered the fastest circuit speed that can be realised, so it is the most common parameter used to monitor the real speed performance of a CMOS technology. Fig. 1.1a shows that it increases roughly 11% per year, which is slightly less than the current 13% per year ITRS target. Nevertheless, comparing both plots, we can see that none of these trends translate directly to higher clock frequency. Fig. 1.1b shows that it has become harder and harder to exploit higher transistor speeds. This is mainly due to power consumption constraints (dynamic and static) but also due to increasing Process, Voltage and Temperature (PVT) variability affecting the synchronous design paradigm [5,6].



Figure 1.1: Scaling trends in: a) transistor intrinsic speed; and b) transistor density, clock speed, power and instruction-level parallelism in Intel CPUs.

The usual approach to clock circuit design is to use a Clock Distribution Network (CDN), which is traditionally designed as a clock tree or tree-based structure. Clock trees are networks of wires and buffers that lead from the central clock source to the clock loads. They consume the minimum wiring resources and provide the minimum wiring capacitance and thus, represent the best low-power solution [7]. Unfortunately, they suffer from high sensitivity to spatial variation in either load capacitance and/or buffer strength. Thus, most modern CDNs today are hybrid tree structures, with time averaging schemes (clock grids and spines), delay compensation and/or noise reduction circuits to mitigate clock uncertainty [8]. However, the cost of developing and deploying such

techniques is increasing with technology scaling, making uncertainty management increasingly difficult.

Technology scaling is making CDNs increasingly difficult to design because of several factors. First, the decreasing cost of transistors is being used to integrate more functions onto chips. Thus, even if the die size is kept constant, circuit density increases. This, along with increasing clock frequencies, has increased power consumption to levels now limited by package cost, reliability, and cooling cost issues. Thus, techniques to mitigate clock uncertainty must comply with stringent power consumption envelopes. Second, the relative non-scaling of wire delay and the increasing amount of capacitance per unit area exacerbate clock latency and increase the required gain of the clock network (i.e., additional levels of clock buffers are necessary). This further increases clock uncertainty, which is known to be roughly proportional to the latency of the clock distribution. Third, complex systems increasingly rely on previously designed Intellectual Property (IP) cores, which represent an obstruction to clock lines and buffer placement. Finally, because of the complexity of the deep-submicron processes, designers can no longer ignore the PVT (temporal and spatial) variations. The impact of variability is more evident on the clock network, in which clock path delay uncertainty can lead some parts to fail performance targets or even become inoperable.

This scaling challenges have led to a search for alternative clock distribution methods. Standing-wave [9] clock distributions have been proposed to reduce clock uncertainty and power dissipation. However, nonuniform amplitude and phase across the distribution makes integration with existing local clocking methodologies more difficult and expensive. Travelling-wave clock distribution [10], eliminates these limitations and has been proposed as a viable alternative to traditional global electrical clock distribution. Other methods include optical [11], Radio Frequency (RF) [12], current-mode [13] or package level CDNs [14]. However, these techniques have less Computed Aided Design (CAD) tool support and typically require auxiliary circuits to convert the signal back to electrical form and restore its integrity. Also, with the strong emergence of multidomain clock distribution in multicore processors and System-on-Chips (SoCs), constraints due to global interconnect are significantly alleviated. As those methods are efficient only for global clock distribution, on-die electrical clock distribution is expected to continue being the predominant clocking technology in the future.

Asynchronous circuits have also been proposed as a means to eliminate the clock uncertainty problem [15]. Since asynchronous circuits by definition have no globally distributed clock, there is no need to worry about clock skew or jitter. However, asynchronous circuits are more difficult to design and, in general, cannot leverage off existing CAD tools (e.g., placement, routing, partitioning, logic synthesis, etc.). Moreover, even though most of the advantages of asynchronous circuits are towards higher performance, it is not clear that asynchronous circuits are actually any faster in practical applications.

Future high-performance VLSI systems are thus expected to continue relying in the synchronous (or loosely synchronous) design paradigm, where the clock signal is distributed through electrical hybrid tree structures, incorporating static CMOS logic gates ¹. The research in this thesis is driven by the need to understand the sources of clock uncertainty in these CDNs and evaluate its evolution with techonology scaling. It proposes models that highlight the key parameters in jitter insertion and accumulation mechanisms, which are then used to discuss the limits of the synchronous design paradigm within increasingly noisy digital environments.

1.2 Goals

I N current high-performance synchronous and loosely synchronous VLSI designs, multiphase clock solutions can boost the performance of parallel processing architectures. Thus, performance evolution may depend more on clock uncertainty decrease rather than on clock frequency increase. The primary goal of this thesis is to better understand clock uncertainty in order to identify the key circuit and environmental parameters on which it depends and predict its evolution with the challenges introduced by technology scaling. This goal was accomplished through the development of jitter insertion and accumulation models for clock repeaters and repeater structures. These models considered intrinsic and environmental variability sources, thus providing the capability for

¹Dynamic logic, while attractive for performance in low-frequency or clock-gated regimes, is increasingly less popular because it consumes more power and is more sensitive to delay variability [16].
predicting jitter evolution in different scaling scenarios and discussing the future of the synchronous design paradigm.

Specific goals are the following:

- Compare the performance of different static and tunable repeater designs in order to identify opportunities for clock precision improvement at the buffer level.
- Develop a general model to predict jitter insertion in static and tunable clock repeaters. It should consider the complete repeater cell, including gate and interconnect, as well as intrinsic and environmental variability sources. Also, it should only include simple circuit parameters that can be easily obtained from technology providers, so it can be used to discuss clock precision trends and limits under different technology scaling scenarios.
- Compare the performance of different static and tunable delay lines, evaluating their linearity and clock precision. Results can be used to identify opportunities for performance improvement in direct and feedback CDNs.
- Develop a general model to predict jitter accumulation in static and tunable repeater lines, considering intrinsic and environmental variability sources. The model should be sufficiently simple so it can be used to evaluate the impact of design choices on clock precision, with low computational effort.
- Develop an analytical model to evaluate static and dynamic uncertainty in multidomain clock synchronisation schemes.
- Estimate the effects of technology scaling on clock precision, using the aforementioned models, and determine the limits beyond which the synchronous design paradigm fails and ceases to be the most effective digital design methodology.
- Provide guidelines for jitter-aware clock distribution design in synchronous and loosely synchronous design styles.

1.3 Original Contributions

HIS work includes several original contributions in the area of clock distribution. Specifically, it provides comparisons and proposes jitter models for clock repeaters

and clocking structures, which can be used to assess the limits and trends of the synchronous design paradigm. These contributions are described bellow.

- A performance comparison between static and tunable clock repeaters, including delay, signal integrity, power, area, and time precision. Time precision is evaluated considering intrinsic and environmental jitter sources, for different fanouts. Results show that uncertainty, measured as jitter per unit delay, is almost constant in these structures. This means that for a given insertion delay, time precision is determined essentially by the implementation technology. This comparison was published and presented in ISCAS 2009 - IEEE International Symposium on Circuits and Systems, held in Taipei, Taiwan [17].
- 2. A model to estimate jitter in CMOS inverters, using circuit parameters that can be easily obtained for a given technology. This model includes sensitivity metrics to intrinsic and environmental variability sources, providing a valuable insight regarding the key circuit parameters responsible for jitter generation. It can also be used to assess the expected behaviour of existing and future technologies in terms of clock precision. A first approach to identifying the key circuit parameters in jitter insertion was published and presented in PRIME 2007 IEEE PhD. Research in Microelectronics and Electronics, held in Bordeaux, France [18]. The complete model, along with a discussion on clock precision degradation with technology scaling, was published in Integration, the VLSI Journal [19]
- 3. A scalable model to estimate jitter in general clock repeaters with RC interconnects. It includes expressions to estimate both static and dynamic clock jitter insertion in repeaters with different sizes, interconnects and slew-rates, with low computational effort. It requires only the pre-characterisation of a reference repeater, which can be accomplished with a small number of simulations or measurements. The model was shown to be accurate to within 10% of simulation results, for repeaters with variable fanouts and input transition times. A partial model, considering only power supply noise induced jitter, was published and presented in PATMOS 2009 - International Workshop on Power and Timing Modeling, Optimization and Simulation, held in Delft, Netherlands [20]. The complete model was submitted to

the IEICE Transactions on Fundamentals, in January 2012 [21], along with a jitter accumulation model described as the sixth original contribution.

- 4. A performance comparison between uniform and capacitively loaded digitally controlled delay lines, including intrinsic and environmental jitter. Simulation results and a simple accumulation model show that thermal noise induced jitter in uniform delay lines is always higher than in capacitively loaded lines, with similar latency, while power supply noise induced jitter is comparable in both structures. Results were published and presented in two conferences, both national and international, namely: ICECS 2006 IEEE International Conference on Electronics, Circuits and Systems, held in Nice, France [22]; and ConfTele 2007 Conference in Telecommunications, held in Peniche, Portugal [23].
- 5. A study on clock uncertainty with technology scaling, in digitally controlled delay lines. It evaluates clock uncertainty trends considering different noise sources and loading conditions. It shows that the device size-scaling trend is increasing the uncertainty associated with these circuits, decreasing their precision. The correlation between circuit's parameters and selected performance metrics was also highlighted. Results were published and presented in PATMOS 2008 - International Workshop on Power and Timing Modeling, Optimization and Simulation, held in Lisbon, Portugal [24].
- 6. A model for jitter accumulation in general clock repeaters, considering both intrinsic and environmental jitter sources. It proposes expressions for dynamic jitter accumulation, considering the the dual nature of power and ground noise impact on delay. Along with the scalable jitter model described as the third contribution, it provides power supply noise jitter predictions for clock trees with an error within 10% of simulation results, for typical designs. This is a much better accuracy than the conventional statistical accumulation model can provide. Also, it can be used to replace time-consuming transient noise simulations when evaluating jitter in clock distribution systems, and provide valuable insights regarding the impact of design parameters on jitter accumulation. This model was published and presented in ISCAS 2011 IEEE International Symposium on Circuits and Systems,

held in Rio de Janeiro, Brazil [25]. It has also been included in [21].

7. An uncertainty model for Delay Locked Loop (DLL)-based deskewing systems, including floorplanning and scalability issues. It shows that, in spite of the multiple schemes proposed in the last two decades, DLL-based deskewing systems are either implemented as Local Deskewing Systems (LDSs) or Remote Deskewing Systems (RDSs). LDSs are used to eliminate skew between two adjacent synchronous domains, while RDSs eliminate only clock distribution skew. This fundamental difference impacts both their skew and jitter performance, which can be evaluated using the proposed analytical model. As it depends only on parameters that can be easily obtained from design or early simulation data, it can be incorporated in an automatic tool to determine the best topology for a given application or to evaluate the system's tolerance to power-supply noise. Also it can be used to evaluate the performance of alternative deskewing schemes, under different scaling scenarios. Results show that regardless the system architecture, deskewing schemes trade static for dynamic uncertainty, with the additional disadvantage of area and power overheads. This model will be submitted for publication in PATMOS 2012 - International Workshop on Power and Timing Modeling, Optimization and Simulation, held in Newcastle, United Kingdom [26].

1.4 Software Support

THE work reported in this thesis was accomplished using commercial and open source tools. Text editing was done using Kile, a TeX/LaTeX editor for the KDE desktop environment running in a Ubuntu 9.10 based platform. Graphics and figures were designed with GIMP, OpenOffice drawing tool and OpenOffice spreadsheets. Numerical simulation of proposed models was performed using Scilab 5.1, which is an open-source alternative to MATLAB. For electronic circuit design, layout and simulation, we used Cadence Design Framework II, available at the University of Aveiro. Some published simulation results were obtained using SMASH, a logic and mixed-signal simulator from Dolphin Integration.

1.5 Thesis Organisation

HIS thesis is divided into seven chapters. This chapter presents the motivation, the objectives and overview aspects. The following chapters are described next.

Chapter 2 introduces the synchronous design paradigm and fundamental concepts for understanding clock uncertainty and its sources. It also provides a general introduction to timing analysis techniques and traditional models for clock delay and delay uncertainty. Finally, it discusses the most common techniques and structures used for clock distribution in today's high-performance VLSI systems.

Chapter 3 introduces different clock repeater architectures, compares their performance and proposes two different jitter models to predict their performance under intrinsic and environmental jitter sources. The first is referred as the reference jitter model, and can be used to predict jitter and jitter sensitivity based on simple circuit parameters. The second is called the scalable jitter model, and can be used to estimate both static and dynamic jitter in repeaters with different sizes, interconnects and slew-rates.

Chapter 4 investigates how uncertainty propagates and accumulates in clocking structures. These can be delay lines, used to introduce controllable amounts of delay in the clock path, or clock trees, used to distribute a clock signal from one source to multiple sinks. It begins introducing different delay line architectures and comparing their performance, considering timing, power and area metrics. Then, it proposes a model for dynamic jitter accumulation that can be used to predict jitter in these structures, with much higher accuracy than the traditional statistical accumulation models. Finally, it proposes an uncertainty model for deskewing systems, with different architectures.

Chapter 5 describes the experimental framework used to evaluate the proposed models. Results show the accuracy and applicability of those models and support the conclusions taken in previous chapters, which were based on simulation results.

Chapter 6 discusses the limits and trends in synchronous clocking, using the proposed jitter insertion and accumulation models coupled with models for variability sources and their evolution with technology scaling. Different scaling scenarios are considered to evaluate the limits imposed by clock uncertainty in repeaters and clocking structures, as well and the expected trends with technology scaling. The latest ITRS reports are also

used to discuss the future directions of clock distribution in high-performance systems, and the limits of the synchronous design paradigm.

Chapter 7 concludes by summarising the developments and contributions this thesis, and identifying possible areas for future work.

Chapter 2 Timing in Synchronous Systems

Synchronous clock delivery in VLSI circuits has always been a major design challenge. Without an adequate clock signal, synchronous circuits will experience setup time and/or hold time violations, and they will consequently fail to operate properly. This chapter begins with a review of synchronous digital systems and the role of the clock in these systems. Clock parameters used throughout this thesis are here introduced, and the main sources of timing uncertainty discussed. Next, it provides a brief overview of some of the models and methods commonly used for timing analysis. Finally, this chapter provides a brief review of current high-performance clock distribution network topologies and optimisation techniques.

2.1 The Synchronous Paradigm

M OST high-performance digital systems today are synchronous systems that use a clock signal to control the flow of data throughout the chip. This greatly eases the design of the system because it provides a global framework that allows many different components to operate at a given reference time while sharing data. However, the clock signal typically has the largest fanout, travels the longest distances, and operates at the highest speeds, when compared to others. Thus, it usually requires a network with several levels of amplification (clock repeaters), which may introduce delay uncertainty as a consequence of noise, interference or process variations. Any uncertainty in clock arrival times between two registers, especially if these registers share the same data path, can limit overall circuit performance or even cause functional errors. This section defines the most significant clock timing parameters that will be used throughout this thesis.

2.1.1 Synchronous Operation

The synchronous system assumes the presence of storage elements and combinational logic which together make up a finite-state machine, as shown in Fig. 2.1a. The clock signal is used to define a time reference for the movement of data within that system. The machine's present state (S_n) is fed back as part of the logic inputs, keeping its value between two clock's rising edges. The signals at the outputs of the logic have to be stable before the next clock, which means that the clock period (T_{clk}) has to be longer that the longest path through the logic. On the other hand, the clock frequency (f_{clk}) has to be high enough to catch all the input changes.



Figure 2.1: The synchronous paradigm: a) concept of a finite-state machine; and b) clock signal's timing parameters.

The clock signal is also characterised by the ratio between the pulse width (W_{clk}) and T_{clk} , which is defined as the clock duty-cycle. If the clock has a symmetric shape, it has a 50% duty-cycle. Other important timing parameters are the rising (t_r) and falling (t_f) times. A common definition for these times is the time the signal takes to go from 10% to 90% of the full swing, as shown in Fig. 2.1b. However, when the signal settles very slowly, is very noisy, or the swing is small, 20% and 80% levels are usually better suited to measure rise/fall times ($t_{r_{20/80}}$ and $t_{f_{20/80}}$). Another possibility is to consider the rise/fall times that would be measured if transitions were linear, with slope equal to the maximum slew-rate ($max{SR} = SR_{max}$). To simplify the notation throughout this thesis, symbols t_r and t_f correspond to $t_{r_{10/90}}$ and $t_{f_{10/90}}$, unless otherwise noted.

To assure the correct behaviour of a synchronous system it is necessary to guarantee that setup and hold times are respected. Setup specifies the amount of time during which a digital signal from one stage of the sequential structure has to be stable, before being captured by the next stage of the sequential structure. Hold specifies the amount of time during which that signal has to remain stable, after the capturing clock edge. Fig. 2.2a shows a typical synchronous sequential structure bounded by two flip-flops with a logic circuit that exhibits a nominal propagation delay (t_p). The sequential elements are clocked by a source clock Clk1 and a destination clock Clk2.



Figure 2.2: Synchronous operation constraints: a) sequential structure; b) setup time constraint; and c) hold time constraint.

For the system to operate correctly, the clocks must be delivered at a precise relative time. If the clock sinks were located close together there would be no problem. However, since the clock signals are routed via a distribution network that includes clock distribution logic and interconnects, they may arrive at the inputs of the processing elements at different times. The absolute difference of the clock arrival times ($|t_1 - t_2|$) is known as clock uncertainty. This uncertainty plays a fundamental role in determining whether the setup and the hold constraints can be robustly met.

The setup constraint (Fig. 2.2b) specifies how data from the source sequential stage at cycle N can be captured reliably at the destination sequential stage at cycle N+1. This situation is defined in inequality (2.1), where $t_{p,slow}$ is the slowest (maximum) data path delay, t_{su} is the setup time for the receiver flip-flop, and t_1 and t_2 are the arrival times for clocks Clk1 and Clk1 (at cycle N), respectively. In this situation, the available time for data propagation is reduced by clock uncertainty. In order to accommodate clock uncertainty and meet the inequality in (2.1), either the clock period must be extended or the path delay reduced. In either case, power and operating frequency are affected.

$$T_{clk} \ge t_{p,slow} + t_{su} + |t_1 - t_2| \tag{2.1}$$

The hold constraint (Fig. 2.2c) refers to the situation where the data propagation delay is fast ($t_{p,fast}$). Clock uncertainty makes the problem even worse and the data intended to be captured at cycle N+1 may be erroneously captured at cycle N, corrupting the receiver state. In order to ensure that the hold constraint is not violated, the design has to guarantee that the minimum data propagation delay is sufficiently long to satisfy (2.2), where t_{hold} is the hold time requirement for the receiving flip-flop. Meeting this constraint with large clock uncertainty could result in setup violation since the slowest manifestation of the same path could violate the delay requirement in (2.1). Such two-sided constraints are not uncommon in current high-performance synchronous systems if the clock uncertainty is high [27].

$$t_{p,fast} \ge t_{hold} + |t_1 - t_2| \quad \text{, with } t_{p,fast} < t_p < t_{p,slow}$$

$$(2.2)$$

2.1.2 Clock Uncertainty

Clock uncertainty can be used when referring to clock jitter and/or clock skew. In the past, jitter was considered as mainly introduced by the clock generator while skew was considered to be caused by static path-length mismatches in the CDN. Thus, skew and jitter were used to distinguish between static and dynamic clock uncertainty, respectively. However, as clock distribution delay became dominant, this distinction became less appropriate. In this thesis, clock jitter is used to represent the difference between the actual and the nominal clock arrival times, whether it is a static or dynamic uncertainty. On the other hand, skew is used to describe the unintentional time difference between two spatially distinct clock edges¹. This is illustrated in Fig. 2.3. Skew (*S*) is shown as the difference between the mean of two edges ($S_{12,a}$ and $S_{12,b}$), while jitter (*J*) is the standard deviation (or peak-to-peak range) of a single edge ($J_{k,a}$ and $J_{k,b}$, with k = 0, 1, 2).

¹In some situations, the clock arrival times are intentionally skewed to facilitate time borrowing across sequential boundaries [28]. This is not considered here to be part of skew.



Figure 2.3: Skew and jitter definitions, as components of clock uncertainty: a) clock distribution network with two clock paths; and b) absolute jitter in clock edges vs. skew between different clock signals.

There are three different definitions for clock jitter, depending on the timing reference. These are cycle-to-cycle jitter, period jitter and absolute jitter. Cycle-to-cycle jitter is defined as the clock signal variation between two consecutive clock edges. Because it is a very high-frequency phenomenon, it is also sometimes referred as short-term jitter. Period jitter is defined as the difference between a given period of the clock signal and its average period. This is specially relevant in systems where the minimum (or maximum) time period is of importance. Finally, absolute (or long-term) is defined as the difference between the edges of a clock signal being measured at the ideal locations (where the edges would occur in the absence of variations). Because it represents accumulated effects, absolute jitter is the most representative metric when discussing clock precision and is the one considered in this thesis when clock jitter is discussed.

To benefit from the faster clock frequencies allowed by technology scaling, clock uncertainty must remain a constant portion of T_{clk} . However, variability sources do not scale equally with the transistor speed. Fig. 2.4a shows clock skew as a percentage of cycle time vs. operating frequency for a number of recent Microprocessor Units (MPUs) [29]. The trend is for skew to be kept around 5% of T_{clk} , due to the adoption of skew tolerant clock distribution topologies, more robust design flow, and the incorporation of post-silicon compensation techniques. On the contrary, Fig. 2.4b shows that clock jitter has been continuously increasing when compared to T_{clk} , meaning that it is increasingly more important when designing high-performance synchronous systems [30–37].



Figure 2.4: Clock uncertainty as a percentage of cycle time vs. processor clock frequency: a) clock skew; and b) clock jitter.

Besides the increase in variability sources, other aspects have been contributing to the increasing importance of clock uncertainty in the design of synchronous systems. First, chip area has increased when compared to the transistor dimensions. As the chip area increases, more resources are needed to compensate for interconnect effects, like loss and dispersion, which worsens with technology scaling [38]. With more buffer stages, the CDN becomes more sensitive to variations and harder to tune. Second, the increasing design complexity introduces new challenges in CDNs. Modern digital systems use more than one clock frequency, so the system is no longer fully synchronous; rely on previously designed IP cores, which represent an obstruction to clock lines; and/or employ different techniques to save power, which induce clock loading varitions. Mitigating uncertainty in such complex networks is thus often impossible, or the power cost of doing so is unacceptable. Finally, scaled devices are increasingly sensitive to sources of clock uncertainty, as will be shown latter in this thesis.

2.2 Sources of Clock Uncertainty

CLOCK uncertainty results from variations of intrinsic and/or environmental parameters. This section describes the most relevant variability sources, responsible for both static and dynamic clock uncertainty in digital CMOS circuits. Typical measures to mitigate their impact are also briefly discussed.

2.2.1 Intrinsic Variations

Process Parameters

Process variations are a major challenge in the semiconductor industry today, because of processing and masking limitations [4]. The increasing difficulty in controlling the uniformity of critical process parameters in increasingly smaller devices, makes their electrical properties much less predictable than in the past. The combined effect is a statistical performance distribution of final products. Several taxonomies can be used to describe the different variability mechanisms, according to their causes, spatial scales and the particular Integrated Circuit (IC) layer they impact. However, for the circuit designer, the primary distinction is between inter-die (or die-to-die) and intra-die (or within-die) variations [39]. This distinction is conceptually represented in Fig. 2.5 [40].



Figure 2.5: Partition of process variation in inter-die and intra-die variations.

Inter-die variations (lot-to-lot, wafer-to-wafer or within-wafer) affect all devices on a die in the same way and are usually modelled as a shift in the circuit device parameters. Although they may have systematic trends according to the die orientation and location on the wafer, this information is usually not available at design time. Thus, the impact of inter-die variability must be captured using a random variable, which is usually assumed to have a simple distribution (e.g., Gaussian), with a given variance.

Intra-die variation causes device parameters to vary across different locations within a single chip. Depending on the variability source, they may be classified as spatially correlated or uncorrelated. Uncorrelated variations affect transistors and interconnects in a different way even if they are relatively close. These variations are also usually referred to as random. Random variations include those whose origins can be truly said to be random (e.g., random dopant fluctuations) as well as those that are not truly random, but that are difficult to model. On the contrary, correlated variations are usually referred to as systematic, because they affect close devices in the same way according to layoutpattern-dependent factors. While systematic variations can be modelled and accounted for in the design flow, random variations can only be handled through worst-case design margins or sophisticated optimisation methods [41]. As CMOS devices are scaled down, the increased contribution of the random dopant fluctuations and limitations in optical lithography are expected to continue contributing to increase intra-die variability. Thus, clock performance is expected to decrease with further dimensional scaling [42].

Intrinsic Noise

In the frequencies of interest for current digital systems, thermal and flicker noise are the dominant sources of intrinsic variability [43]. Thermal noise derives from carrier agitation and requires only a population of carriers within a conductive region. Therefore, it appears in both passive and active devices and is generally modelled as a white and statistically stationary random process, characterised by a Gaussian distribution. On the contrary, flicker noise appears only in active devices. It is roughly inversely proportional to frequency and is usually considered a statistically non-stationary random process. This implies that its mean changes with time and its uncertainty on any period cannot be reduced by averaging over longer periods. For the analysis and design of CMOS analog and RF integrated circuits, flicker noise is one of the factors limiting the achievable performance [44]. However, it is usually considered a second order effect for jitter analysis in digital circuits.

In digital circuits, thermal noise is dominated by the transistor's Thermal Channel Noise (TCN), which increases with technology scaling [45]. The most common expression for its single-sided Power Spectral Density (PSD) in the saturation region (S_{i_d}) is shown in (2.3), where *T* is the carrier temperature, k_B is the Boltzmann constant, g_{d0} is the output conductance at zero drain bias, γ is the channel noise coefficient in saturation and g_m is

the maximum transconductance in saturation [46]. For long channel devices, $g_{d0} \approx g_m$ and γ is considered to be approximately 2/3.

$$S_{i_d} = 4k_B T \gamma g_{d0} \approx (8/3) \cdot k_B T \gamma g_m \tag{2.3}$$

For short channel devices, both g_{d0} and γ are complex functions of the device parameters. Two alternative TCN models are available in the Berkeley Short-channel Insulatedgate field-effect transistor Model (BSIM) [47] for Simulated Program with Integrated Circuits Emphasis (SPICE) thermal noise analysis. However, they depend on several biasdependent and fitting parameters and thus, are suitable for circuit simulators only.

2.2.2 Environmental Variations

The two main environmental variability sources are temperature fluctuations and the current changes induced by the circuit's activity. When current changes are reflected in power, ground or substrate voltage changes, they are referred to as Power Supply Noise (PSN). When changes in one circuit node have a direct impact on other circuit's node, they are called interference (or crosstalk).

Temperature

With increasing circuit densities and system complexity, temperature variations due to the heat generated on-chip have become increasingly more significant. Thermal differences between circuit regions can be as high as 40 °C or 50 °C in high-performance designs, creating non-uniform thermal maps [48]. Because the global clock signal is distributed throughout the chip, temperature induced global skew can be significant. Fortunately, heat conductivity of silicon substrate is usually good and thermal maps are known to change smoothly in space and time [49]. This means that temperature variations can be considered quasi-static and thus, their effects can be easily mitigated with pre and/or post layout optimization/compensation schemes [50].

Power Supply Noise

The goal of a Power Delivery Network (PDN) is to provide a clean supply voltage to active devices. However, the flow of time-varying currents through the PDN impedance (resistance, capacitance, and inductance) generates undesired voltage fluctuations, commonly referred to as PSN. This noise can impact the circuit delay in two ways. On the one hand, a reduced supply voltage lessens the gate drive strength, thereby increasing the gate delay. On the other hand, a difference in the supply voltage between a driver and receiver pair creates an offset in the voltage with which the driver/receiver gates reference the signal transition. This has the effect of creating either a positive or negative time shift in the perceived signal transition at the receiver gate [51], which makes PSN induced delay variation much more complex to analyse.

PSN level on power (V_{dd}) and ground (V_{ss}) rails depends on the PDN impedance and on the transient current associated with each rail. Thus, to minimise PSN, the designer should guarantee a low PDN impedance at all frequencies that can be excited by the current waveforms. This is usually done with power and ground grid structures, commonly referred to as power grids or power planes. Other popular techniques to minimise the PDN impedance include:

- Board level locate the Voltage Regulator Module (VRM) as close to the IC as possible, to reduce the inductive and resistive components of the power supply leads linking them. Decoupling capacitors should also be used to flatten the on-board PDN impedance, helping the VRM to respond to instantaneous current requirements.
- *Package level* replace traditional wire-bond packages with low inductance packaging styles, as flip-chip or bump-bond packages. On-package decoupling capacitors can also help in reducing the impact of package inductance.
- *Chip level* add on-chip decoupling capacitance to help reducing the impact of onchip inductance [52]. Although it has been shown in [53] that chip performance is less sensitive to amount on-die decoupling capacitance than it was conventionally expected, this is still a popular technique. Another option it to use of more power supply and ground pins, which helps reducing the total pin inductance.

A typical PDN using these techniques is illustrated in Fig. 2.6. The supply current comes from the on-board VRM, and is fed into the package through a Ball Grid Array (BGA). The current then flows through power planes and vias on the package, enters the chip through Controlled Collapse Chip Connect (C4) bumps, and finally is distributed to on-chip circuitry by on-chip power grids. Decoupling capacitors in each stage serve as local storage to supply charge to the next stage when quickly needed. The coverage frequency increases from the regulator to the die, using progressively higher quality (e.g., smaller parasitic inductance and resistance) and lower valued decoupling capacitors [54].



Figure 2.6: PDN with on-chip, package and board components.

In high-performance designs, the on-chip PDN is also typically designed as a hierarchical structure [51]. The top-level network connects to the macro-blocks while a local network inside the macro-block connects to the logic gates. A simplified circuit model is shown in Fig. 2.7. In this model, power supply pins and wires are modelled as a series of RLC elements, usually referred to as supply parasitics [55]. With this arrangement, localised supply variations induced by signal transitions inside each block do not contribute to clocking errors because the effect is the same on every clock cycle, and hence affects each rising clock edge the same way. However, they may have a significant impact in circuits that drive circuits in another blocks, as in the case of clock repeaters in global and/or regional CDNs.

Voltage fluctuations are primarily generated by the PDN's resistive and inductive components. Static voltage drops (IR drops) are developed on the power grids due to the circuit's average current consumption. On the contrary, dynamic voltage drops/surges occur due to transient currents caused by inductive parasitics and are known as switching



Figure 2.7: Simplified circuit model for a typical PDN with bump-bond packaging.

(or di/dt) noise. From a statistical point of view, static noise corresponds to the difference between the mean supply/ground voltages and their to nominal values, while dynamic noise corresponds to supply/ground voltage standard deviation.

If the package-level inductance dominates the PDN's total inductance, supply noise is usually symmetric to ground noise ($\Delta V_{dd} = -\Delta V_{ss}$), and Differential Mode Noise (DMN) is assumed to be dominant [56]. If not (e.g., if low inductance packaging is used), no single parasitic can be considered to dominate and the PDN may exhibit Common Mode Noise (CMN), which corresponds to in-phase power/ground fluctuations ($\Delta V_{dd} = \Delta V_{ss}$) [51]. Noise modes are also known to depend on the switching activities of active circuits and the correlation degree (spatial and/or temporal) among the switching nodes [57],[58]. Thus, it is not reasonable to consider specific assumptions regarding the PSN profile seen by a given clock repeater or a clocking structure in a modern high-performance digital circuit. Thus, only three general PSN assumptions are considered in this thesis:

- PSN is low-frequency local decoupling capacitance (parasitic and/or added) is considered to limit the magnitude of the highest speed noise excursions, so they occur at a slower time scale than the clock switching transitions.
- 2. PSN has a zero-mean Gaussian distribution noise is considered to include only dynamic variations (*di/dt* noise), which results from the ensemble effect of on-chip devices switching at a variety of different frequencies, slew-rates and/or time instants. Static or quasi-static IR drops are not considered a relevant component

of PSN, for the same reason that temperature was not considered to be a relevant environmental source of clock uncertainty (the effects of static or quasi-static variability sources can be mitigated with compensation/calibration schemes).

3. PSN is a Mixed Mode Noise (MMN) source - power and ground rails are affected by independent noise sources, which can be evenly decomposed into common and differential mode noise (CMN and DMN) components.

Part of the reason that PSN has become one of the most significant uncertainty sources in synchronous systems is the steadily-shrinking design rule used in semiconductors, which results in reduced supply voltage margins, shorter transistor switching time and increased on-chip currents. A change of one generation in design rule means about $2\times$ the quantity of transistors per unit area and $0.7\times$ the gate width, so total current consumption per chip unit area increases to about $1.4\times$. On the other hand, spatial imbalances between the currents in various parts of a chip are accentuated, particularly with the advent of multicore systems (where some cores may switch on and off entirely) and three dimensional ICs. To reduce noise and/or its impact on circuit performance, different techniques have been proposed, like slew-rate control [59] or on-die power supply filtering [36]. However, these techniques also introduce performance and complexity penalties which cannot be disregarded.

Substrate Noise

All current injected into the substrate causes fluctuations of the substrate voltage, i.e., substrate noise. This noise produces the effect of modulation of the current response (body effect), which degrades or alters the transitory behaviour of devices. However, in digital CMOS circuits, the substrate is commonly biased by a large number of contacts connected to ground. Thus, this thesis considers substrate noise to be part of PSN.

Crosstalk

When conductors are placed sufficiently close to each other, signals on the lines can interfere with each other via near-field electromagnetic coupling. In circuit theory, the electric field coupling is described as capacitive crosstalk while the magnetic field coupling is described as inductive crosstalk. The influence of capacitive and inductive crosstalk between the aggressor line (line A) and the victim's line (line B), is represented in Fig. 2.8 [60]. Each line is modelled by a lumped resistance (R_{int}), capacitance (C_{int}), inductance (L_{int}), mutual capacitance (C_m) and mutual inductance (L_m). These mutual parameters are the ones responsible for crosstalk.



Figure 2.8: Electromagnetic coupling in neighbouring interconnects.

Crosstalk between the aggressor and victim lines is a major source of performance degradation for two reasons. First, it introduces variability in the victim's effective line capacitance and inductance, increasing or decreasing the signal delay (crosstalk delay). If this delay exceeds the allowed time margins, time violations or system malfunction may occur. Second, crosstalk introduces unexpected glitches that, if captured by end latches, can also produce erroneous logic values. Crosstalk delay is usually more serious than glitches, because these do not always result in easily perceptible logic changes [61].

The best way to protect the clock signal from aggressors is by shielding it. A common method of shielding is placing ground or power lines at the sides of the clock line. [62]. Differential signalling can also be used to mitigate crosstalk - by encoding the information in the voltage difference of a pair of wires, any noise source affecting both wires of the differential pair is filtered [63]. However, the differential signal needs to be converted back to single ended before reaching the flip-flops. Because these techniques introduce significant resource utilisation penalties, they are usually found only on the higher branches of the CDN, if used.

2.3 Timing Analysis

T HIS section describes the most popular models and tools used to evaluate delay and delay variability in a CMOS inverter, which is the most common clock repeater and the basic building block of any digital circuit.

2.3.1 Timing Models

Circuit designers require accurate timing models for estimating the performance of CMOS circuits. Model inaccuracies should be reduced as much as possible because they directly translate into timing overhead, that degrades speed performance. This section describes analytical and empirical delay models suitable for the CMOS inverter. Some of these models will be used throughout this thesis.

Analytical Models

Early MOSFET timing models were based on Shockley's square law current model [64]. To improve their accuracy in Deep Sub-Micron (DSM) and nanometer technologies, Sakurai et al. [65] proposed an alternative semi-analytic current model - the α -power law model - that included the carrier velocity saturation effect of short channel devices. Expressions for the inverter's output transition time (t_{out}), charging and discharging inverter delays ($t_{d,LH}$ and $t_{d,HL}$) were also proposed, as shown in (2.4) and (2.5).

$$t_{out} = \frac{C_L V_{dd}}{I_{d0}} \left(\frac{0.9}{0.8} + \frac{V_{d0}}{0.8 V_{dd}} + ln \frac{10 V_{d0}}{e V_{dd}} \right)$$
(2.4)

$$t_{d,HL}, t_{d,LH} = \left(\frac{1}{2} - \frac{1 - v_T}{1 + \alpha}\right) t_{in} + \frac{C_L V_{dd}}{2I_{d0}} \text{ , with } v_T = \frac{V_{th}}{V_{dd}}$$
(2.5)

In these expressions, t_{in} is the input transition time, V_{th} is threshold voltage of the transistor, α is the velocity saturation index for sub-micron devices, C_L is the gate's load capacitance, V_{dd} is the supply voltage, V_{d0} is the drain saturation voltage and I_{d0} is the MOSFET drivability at $V_{gs}=V_{ds}=V_{dd}$. Although this model is quite accurate when the t_{in} is small compared to t_{out} , it becomes less accurate for higher t_{in} . To improve its accu-

racy, several modified Sakurai-Newton models were later proposed [66], [67], [68], [69] and [70]. However, they are all based on linear fitting of the current-voltage transistor's characteristic in saturation region and thus, some model parameters have to be obtained either by simulation or measurement.

A different approach was proposed in [71], where the conventional saturation drive current I_{d0} is replaced with an effective switching current (I_{eff}). It is defined as the timeaveraged drain current from 50% input to 50% output in the rise-to-fall and fall-to-rise transitions, and can be computed as shown in (2.6). The main advantage of this approach is that I_{eff} does not dependent on the load capacitance and can be easily adapted for different technology nodes. Similar approaches can also be found in [72], [73], [74] or [75], for simple inverters, or in [76] for more complex gates.

$$I_{eff} = mean\{I_{eff,n} = I_{ds}(V_{gs}=0.7V_{dd} \wedge V_{ds}=0.9V_{dd}); I_{eff,p} = I_{ds}(V_{gs}=-0.75V_{dd} \wedge V_{ds}=-0.95V_{dd})\}$$
(2.6)

Analytical models have also been proposed to compute delay in gates with RC and RLC interconnects [77], [78]. However, as technology scales and new effects come into play, empirical models become more adequate.

Empirical Models

There are two approaches to empirical gate delay modelling which have gained considerable acceptance: 1) computation of delay through delay tables or k-factor equations; and 2) computation of delay by modelling the gate as a voltage source and a resistance in series with the gate load, i.e., using the gate's Thevenin equivalent¹.

In the first approach, delay and rise-times are obtained by loading each gate/cell in a given library with a discrete load capacitor (C_L) and then changing both C_L and t_{in} . Simulation results for gate delay (t_d) and output transition time (t_{out}) are stored in a twodimensional look-up table and/or fitted into analytical functions (k-factor equations). Synopsys' scalable polynomial delay model is shown in (2.7), which resorts to a product of polynomials to fit timing data.

¹This approach is here considered to be empirical because it requires fitting to approximate the resistance value, usually as a function of input slew-rate and output load.

$$(a_0 + a_1 C_L + \dots + a_m C_L^m) \cdot (b_0 + b_1 t_{in} + \dots + b_n t_{in}^n)$$

$$(2.7)$$

In the second approach, the gate is modelled as a simple resistance (R_d) in series with a voltage step. Using empirical observations, it has been shown in [79] that R_d can be computed using the 50% and 90% time points, denoted as t_{50} and t_{90} in (2.8). This allows the use of simple RC delay estimators, as the Elmore delay [80].

$$R_d = (t_{90} (C_L, t_{in}) - t_{50} (C_L, t_{in})) / (C_L \ln 5)$$
(2.8)

Both methods work well when the interconnects behave like an equipotential surface, i.e., when the driver resistance overwhelms the wire resistance. However, if the two are comparable, these models can introduce significant errors due to the phenomenon known as resistive shielding [81]. Because this is an increasingly frequent effect in modern ICs, the wire resistance cannot be ignored in repeater delay models. One possible approach is to use a second-order driving point admittance model that approximates the total gate load as a π -circuit [82]. Higher-order analyses, as the Asymptotic Waveform Evaluation (AWE) technique [83], can also be used at the cost of higher complexity. However, even the simple π -model circuit is often too computationally expensive to be used in design optimisation loops.

For simplification and compatibility with previous gate delay models, Ratzlaff *et al.* [81] proposed the use of a single effective capacitance (C_{eff}) that captures the resistive shielding effect. This procedure involves a set of iterations, whereby the average load current driven by the driver's Thevenin model through the interconnect π -model is equated to that through a capacitor C_{eff} [84]. Equivalently, this may be thought of as equating and matching the total charge delivered to each circuit over a given time period t_m [85]. This is shown in (2.9), where I_{π} and I_C are defined in Fig. 2.9.

$$\frac{1}{t_m} \int_0^{t_m} I_C(t) \, dt = \frac{1}{t_m} \int_0^{t_m} I_\pi(t) \, dt \tag{2.9}$$

Because this model has too many unknowns, a specific waveform must be assumed for $v_2(t)$. One possible approach is to assume a combination of quadratic and linear func-



Figure 2.9: Loaded gate π -model and its equivalent effective capacitance model.

tions, as shown in (2.10). Starting at an initial voltage V_i , the wave-shape is considered to be quadratic until $t_x = t_m - t_{out}/2$. From there to $t_m = t_d + t_{in}/2$, the driving transistor is in saturation and the voltage is assumed to be linear. The constants *a*, *b* and *c* must be set according to the expected waveform, knowing that the delay approximation accuracy depends on how realistic the waveform assumption is.

$$v_{2}(t) = \begin{cases} V_{i} - ct^{2} & , \ 0 < t < t_{x} \\ a + b(t - t_{x}) & , \ t_{x} \le t \le t_{m} \end{cases}$$
(2.10)

Using (2.10) and expressions for the mean current through near-end (C_2) and far-end (C_1) capacitances, the expression shown in (2.11) was derived in [85]. As expected, C_{eff} lies between C_2 and the total circuit capacitance ($C_1 + C_2$), where parameter λ_c represents the percentage of C_1 contributing to C_{eff} .

$$C_{eff} = C_2 + C_1 \left[1 - \frac{RC_1}{t_m - t_x/2} + \frac{(RC_1)^2}{t_x (t_m - t_x/2)} e^{\frac{t_x - t_m}{RC_1}} \left(1 - e^{\frac{-t_x}{RC_1}} \right) \right] = C_2 + \lambda_c C_1 \quad (2.11)$$

Using the previous expressions, C_{eff} can be computed iteratively as follows:

- 1. Set the load capacitance value equal to the total capacitance ($C_L = C_1 + C_2$);
- 2. Use the load capacitance value to obtain a delay and an output-signal transition time (e.g., using the k-factor equations);
- 3. Using t_d and t_{out} obtained in the previous step, compute C_{eff} using (2.11).
- 4. If C_{eff} is still changing, set $C_L = C_{eff}$ and go to step two.

Due to the iterative nature of this algorithm, it is usually considered to be too compu-

tationally intensive to be used in the context of physical design optimisation. Fortunately, accurate and non-iterative approaches to compute the effective capacitance also exist [86].

2.3.2 Jitter Models

The construction of a well-balanced clock tree is a key step in the design of digital synchronous ICs. However, a well-balanced tree in the nominal corner is not necessarily robust to variations [87]. Thus, it is of vital importance for the designer to know how random variations are expected to affect performance and reliability. This section presents the fundamental background on existing jitter insertion and accumulation models, which will be used latter in this thesis.

Jitter Insertion

A lot of research has been done on jitter and phase noise in electric oscillators [88], [89], [90], [91]. In these circuits, jitter models are based on two main assumptions. First, noise is considered to cause only small voltage perturbations and thus, the circuit can be linearised for the purpose of noise analysis. Second, the effects of noise around the nominal crossing time have a higher impact on jitter than effects of noise long before (or after) the nominal crossing time. The system can therefore be described with a transfer function in the frequency domain.

When analysing digital circuits in an open-loop configuration, like clock repeaters or clock distribution networks, the second assumption holds [44]. However, the linear assumption fails, because these circuits operate in large-signal mode. Thus, jitter must be treated in the time domain. The most popular time domain jitter model for digital gates is the First Passage Time (FPT) model, derived for TCN induced jitter in ideal inverter cells [92]. It states that a given amount of voltage noise ($\sigma_{v_n}^2$) produces a time delay variance ($\sigma_{t_d}^2$) that is inversely proportional to the signal's slew-rate (*SR*) squared (2.12).

$$\sigma_{t_d}^2 = \sigma_{v_n}^2 \cdot (1/SR)^2$$
(2.12)

In this expression, σ_{t_d} corresponds to jitter and *SR* represents the inverter's sensitivity

to noise. This sensitivity is usually given by the ratio between the inverter's drivability and its load capacitance. Using the expressions previously shown for I_{eff} (2.6) and C_{eff} (2.11), the clock signal's slew-rate can be expressed as shown in (2.13).

$$SR = I_{eff} / C_{eff} = I_{eff} / (C_2 + \lambda_c C_1)$$
(2.13)

The FPT model has also been used to show that the time uncertainty associated with gate delay, given as jitter normalised to the nominal delay, can only be reduced if the ratio between the voltage swing and noise can be improved [93]. This result can be obtained using (2.12) and (2.5), under the assumption of fast input transition times ($t_{in} \approx 0$), as shown in (2.14). Delay uncertainty (U_d) is particularly useful to compare the clock precision of circuits with different delays, because it is a relative jitter metric.

$$U_d = \sigma_{t_d} / t_d \approx 2\sigma_{v_n} / V_{dd} \tag{2.14}$$

For PSN induced jitter, there are two main models in literature. Both are based on analytical expressions using the α -power law MOSFET model and consider static voltage noise samples. In [51], PSN induced delay variation ($\Delta t_{d,psn}$) is shown to be linear with respect to the power and ground variations (ΔV_{dd} and ΔV_{ss}) but dependent on the package and threshold voltage considered. For bump-bond packaging, the change in delay is given by (2.15), with delay measured at $V_{dd}/2$. For other packages and/or measuring points, the change in delay follows similar expressions but with cross dependencies on slew-rate and t_{in} . Note that both terms in (2.15) are in agreement with the FPT model, where jitter depends on noise divided by slew-rate.

$$\Delta t_{d,psn} = \frac{C_L}{I_{D0}} \cdot \Delta V_{dd} + \frac{t_{in} + \Delta t_{in}}{V_{dd} (1 + \alpha)} \cdot \Delta V_{ss}$$
(2.15)

A more general (and complex) jitter model was proposed in [94], here shown in (2.16). Parameter λ is the channel modulation factor and ΔV_h and ΔV_l correspond to the variations in supply and ground levels in the input waveform (i.e., power and ground noise injected through the previous stage). Note that once again, delay variation depends on voltage variation divided by the circuit's slew-rate, in agreement with the FPT model.

$$\Delta t_{d,psn} = \frac{C_L}{I_{D0}} \cdot \frac{\frac{1}{2} V_{dd} + \Delta V_{dd}}{1 + \lambda \left(V_{dd} + \Delta V_{dd} \right)} \cdot \frac{V_{dd} - V_{th}}{V_{dd} - V_{th} + \Delta V_h - \Delta V_s} + \frac{V_{th} + \Delta V_h + \Delta V_h + \Delta V_{ss}}{V_{dd} + \Delta V_h - \Delta V_l} \cdot \frac{t_{in}}{2} \quad (2.16)$$

Traditional crosstalk induced jitter models consider only capacitive effects. There are two main reasons for this. On one hand, most on-chip interconnects are designed in such way that their mutual inductance is not sufficiently large to influence each other's electrical characteristics [95]. On the other hand, including inductive effects would prohibitively increase the complexity of jitter models [78]. Thus, crosstalk jitter will hereafter be considered to result from capacitive coupling only.

Crosstalk (CRT) delay is traditionally analysed with techniques to decouple multicoupled lines into an equivalent RC line [96], using an effective capacitance model [97], [98]. This capacitance reflects the signal transient characteristics due to the different switching patterns in potential aggressors. Although very popular, this approach requires detailed layout information on the victim circuit and its neighbors. A more interesting approach is described in [99], where the impact of CRT delay on timing is statistically investigated. They introduce a probabilistic coupling rate (ζ_c), as the ratio between the total coupling capacitance causing CRT delay (C_{ct}) and the total victim's capacitance when no crosstalk is present (C_v). It obeys to a normal distribution with zero mean and standard deviation given in (2.17). Here, t_{sw}/T_{clk} represents the victim's crosstalk window, and M is the number of aggressor segments along the victim's line.

$$\sigma_{\zeta_c} = (C_{ct}/C_v) \cdot \sqrt{(t_{sw}/T_{clk})/M}$$
(2.17)

Crosstalk induced jitter ($\sigma_{t_{d,crt}}$) is then shown to follow the same Gaussian distribution as ζ_c , with a standard deviation (CRT jitter) given by (2.18). Here the symbol ' \approx ' is used because the expression is accurate only on average, as it depends on the aggressor's timing. Nevertheless, it shows that jitter depends on the crosstalk window, on the number of aggressor segments and on the ratio C_{ct}/C_v .

$$\sigma_{t_{d,crt}} \approx t_d \cdot \sigma_{\zeta_c} = t_d \cdot (C_{ct}/C_v) \cdot \sqrt{(t_{sw}/T_{clk})/M}$$
(2.18)

Jitter Accumulation

When several gates are cascaded, the statistics of timing jitter depend on the correlation among the noise sources involved. If each transition is affected by independent noise sources, jitter inserted by a stage can be considered to be totally independent of the jitter introduced by other stages. Thus, the total variance of jitter is given by the sum of the variances introduced at each stage. On the contrary, if noise sources are totally correlated, the standard deviations rather than variances should be added [100]. As as example, lets consider the CDN shown in Fig. 2.10. It represents a source clock path with *M* buffer stages and a receiver clock path with *N* buffer stages, with path delays t_{D1} and t_{D2} , respectively. If τ_i is the actual delay of stage *i* and μ_{τ} is the average delay per stage, t_{D1} and t_{D2} can be computed as shown in (2.19).



Figure 2.10: Sample clock distribution for uncertainty accumulation model.

$$t_{D1} = \sum_{i=1}^{M} \tau_i \approx M \cdot \mu_{\tau} \; ; \; t_{D2} = \sum_{i=1}^{N} \tau_i \approx N \cdot \mu_{\tau}$$
 (2.19)

Assuming that $\mu_{\tau} \approx C_L V_{dd} / I_{d0}$ (using (2.5)), the change in delay per stage ($\Delta \tau$) can be formulated as the sum of partial derivatives (2.20). $\Delta \tau$ is shown to be roughly proportional to the stage delay, which corroborates experimental observations where higher jitter is observed in electrically longer paths [29].

$$\Delta \tau \approx \frac{C_L}{I_{d0}} \Delta_{v_{dd}} + \frac{V_{dd}}{I_{d0}} \Delta_{C_L} + \frac{C_L V_{dd}}{I_{d0}^2} \Delta_{I_{d0}} = \left(\frac{\Delta_{v_{dd}}}{V_{dd}} + \frac{\Delta_{C_L}}{C_L} - \frac{\Delta_{I_{d0}}}{I_{d0}}\right) \cdot \mu_{\tau} = \lambda \mu_{\tau}$$
(2.20)

Additionally, if the delay per stage (τ) is a normally distributed Random Variable (RV)

with standard deviation $\sigma_{\tau} \approx \lambda \mu_{\tau}$, the standard deviations of skew (measured between t_{D1} and t_{D2}) and jitter (associated with clock signals on those paths) can be computed as shown in (2.21), where λ_{skew} and λ_{jitter} are variation coefficients for skew and jitter.

$$S_{12} = \sqrt{M+N} \cdot \lambda_{skew} \cdot \mu_{\tau} \wedge J_1 = \sqrt{M} \cdot \lambda_{jitter} \cdot \mu_{\tau} \wedge J_2 = \sqrt{N} \cdot \lambda_{jitter} \cdot \mu_{\tau}$$
(2.21)

These expressions show that the delay uncertainty grows with the square-root of the number of distribution stages and linearly with the nominal delay per stage. However, they are derived under the assumption that λ_{skew} and λ_{jitter} are known, which is not usually the case. Moreover, they rely on a statistical accumulation model that is too optimistic about statistical independence of variations. This is especially true for sources that are partially correlated in time and space, such as PSN sources.

A different approach is to consider low-frequency sinusoidal PSN variations to analytically evaluate PSN jitter accumulation. It has been used to estimate jitter in oscillators [101], DLLs [102], clock and data recovery circuits [103] and clock trees [104],[105], because it provides a means to analyse jitter accumulation with a significant speedup compared to circuit simulations. However, it considers PSN to have a single dominant low-frequency spectral component, shared by the circuit elements. Although this can be a reasonable assumption for some circuits (and some packaging technologies) it may not be so in others, because it disregards the impact of high and mid frequency PSN components, as well as their temporal and spatial correlations.

2.3.3 Simulation Tools

Most circuit simulation tools today are based on the industry standard SPICE, a freeware simulator developed at the University of Berkeley [106]. Commercial versions of SPICE include HSPICE from Synopsis or SPECTRE from Cadence Design Systems. These simulators resort to delay models of individual components to obtain the circuit's overall timing behaviour. They perform different types of analysis, but the most relevant for this thesis is transient analysis. It computes output variables as a function of time, over

a specified time interval, with initial conditions determined by a DC analysis. Although transient analysis can be used to perform very accurate timing analysis, it can also become prohibitively computationally expensive for large circuits. The alternative is to perform Static Timing Analysis (STA), which is much faster than gate-level simulation (for the basic algorithm, run time is linear with circuit size).

In the traditional STA flow, variations are captured in the form of PVT corners. For example, the fast corner is computed by considering that all the gates (or transistors) are faster than expected and performing a regular deterministic timing analysis. While very successful, STA has three significant limitations: 1) it requires too many corners to handle all possible cases; 2) it is too pessimistic when there are significant random variations [107]; and 3) it cannot easily handle intra-die correlations. To address the increasing number of scenarios (corners), On-Chip Variation (OCV) analysis was introduced around the 130nm node. It allows designers to add margin to the timing paths, accounting for the aggregate number of total variations from a wide variety of sources. However, implementing designs with OCV can also be very computational expensive at advanced technology nodes.

To deal with the pessimism associated with STA, Statistical Static Timing Analysis (SSTA) was proposed. SSTA takes into consideration the statistical distribution of variability sources, the arrival times and gate delays [87]. SSTA algorithms fall into numerical or analytical approaches. Numerical techniques, like Monte Carlo (MC) simulation, generate values for input parameters assuming that they satisfy some distribution (e.g., uniform or Gaussian)[108]. The circuit delay is computed using these values and the procedure is repeated hundreds or thousands of times until enough delay values are obtained for a delay distribution curve [109]. On the contrary, analytical approaches take as input the statistical models for gate delays and variability sources, and construct a Probability Density Function (PDF) of path delays [58], [110]. However, neither of these approaches are currently affordable (in computational cost) in practical designs. To reduce STA pessimism and SSTA cost, Advanced On-Chip Variation (AOCV) analysis has been proposed [111] and incorporated in commercial tools [112]. Yet, it is not immune to the problem of growing PVT corners.

2.4 Clocking Systems

CLOCKING systems can be divided into clock generation and distribution. Their physical implementation depends on the required clock precision, power consumption and implementation area. This section describes the most common clock generation and clock distribution structures, pointing out their strengths and limitations in regard to clock precision. It discusses on-die electrical clock distribution methods only, as they are expected to continue being dominant [29].

2.4.1 Clock Generation

Clock generation begins on a system board, where an accurate and stable system clock reference is generated, usually from a quartz-crystal oscillator. Given the size and limitations of quartz-crystals, the frequency of such clock signal is usually much lower than the desired on-chip clock rate. Even if the system clock reference could be generated at the desired frequency, it would be very hard to bring it on-chip due to the large parasitics associated with packages. Thus, a low-frequency system clock is first brought on-chip and then frequency multiplication is performed to achieve the desired on-chip clock rate. Clock multiplication and alignment can be performed by a Phase Locked Loop (PLL) or a DLL, which produce a clock signal phase-locked to the system clock (reference clock).

The PLL includes a Voltage Controlled Oscillator (VCO) that generates the internal clock, which is then aligned to the reference clock by virtue of negative feedback loop, as shown in Fig. 2.11a. The phase difference between the reference clock and the internal distributed clock is measured by the Phase Detector (PD) and filtered by the Low-Pass Filter (LPF), generating the control voltage for the VCO. When the PLL locks, the VCO generates an output frequency and phase such that the phase detector detects no phase error between the reference and feedback inputs. In addition, the PLL is able to perform clock multiplication if a frequency divider is inserted between the output and the feedback PD input. Typically, a copy of the distribution delay (insertion delay) is included into the feedback loop, ensuring that the internal clock is in phase with the reference clock.

A DLL has a similar structure, as shown in Fig. 2.11b. It also includes a PD and a LPF,



Figure 2.11: Generic block diagrams for the: a) PLL; and b) DLL.

but the VCO is replaced by a Voltage-Controlled Delay Line (VCDL). The filter's output controls the VCDL delay until the external and internal clocks are aligned. Unlike the PLL, any noise present in the input clock reference is passed through the VCDL to the output of the DLL, without any filtering. Thus, DLLs perform better when the reference clock is not the main noise source and most uncertainty is introduced by the VCDL. On the contrary, PLLs are better in cases where the input reference noise is dominant and typically worse in cases where the major noise source is introduced in the VCO (where noise accumulates over time), given that VCOs and VCDLs are implemented using the same type of delay element [113], [114].

Jitter in PLLs and DLLs has scaled well with process technology while clock distribution jitter has not. This jitter is known to be proportional to clock distribution latency, which has been scaling slower than clock frequencies. As technology shrinks, wire delay and chip size are constant at best, while clock speeds increase with gate delay (t_d). Yet, the total number of buffering stages (and thus, total latency) increases with $\sqrt{t_d}$. This results from the fact that along an optimally buffered clock distribution line, the distance between buffers decreases with the root of t_d (i.e., the ratio of gate delay (t_d) to wire delay (τ_w) is constant and τ_w is proportional to the square of the wire length). As a result, the current dominant source of clock jitter arises in clock distribution [36].

2.4.2 Clock Distribution

On-chip CDNs rely on device parameter matching. This section discusses CDNs under the optimistic simplification that all systematic variations are compensated by design. Thus, it considers jitter to be induced only by random variations which, if spatially uncorrelated, also contribute to skew. Based on this assumption, it compares several clock architectures and discusses their ability to mitigate the impact of jitter and skew.

Distribution Topology

An important requirement for a low-jitter clock network is to have sharp clock edges. Designers achieve this by inserting buffers and repeaters in the clock network, creating multistage clock trees. This isolates downstream capacitance and reduces transition times. Thus, nearly all on-chip clock distribution networks consist of a series of buffers and interconnects that distribute the clock signal to storage elements.

Regarding its structure, the CDN traditionally consists of two parts: a global clock network and a local network. The global clock network distributes the clock signal from the clock source to local regions and usually follows a symmetric structure. Because only the relative phase between two clocking points is important, symmetry allows the system to exploit the irrelevance of the absolute delay from a central clock source to clocking elements. On the contrary, the local distribution network typically delivers clock signals to registers using an unconstrained tree style structure, because it has a limited span and the clock load is not evenly distributed [7].

Between global and local clock distribution, it is not uncommon to find more hierarchical levels, as shown in Fig. 2.12 [33]. These regional levels do not span as much area as the global level and do not drive as much load as the local level. Typically, regional buffers can be found in a symmetric structure (much like the global network) or may drive clock grids. A clock grid is composed by wires to which the local networks within a region can be connected. Grids are inherently much more immune to variations than trees, due to the redundancy in source-to-sink paths [115]. Also, they make clock design almost independent of floorplanning, which is a very attractive feature. The drawback, of course, is the power dissipation due to extra wiring capacitance and short-circuit currents between drivers [116].

At the global level, most high-performance VLSI circuits use some form of lengthmatched tree to distribute the clock. It is usually electrically balanced and completely symmetric, to simplify the design and provide nominally low skew, as shown in Fig.



Figure 2.12: Clock distribution for the Itanium microprocessor.

2.13. These structures maintain the distributed interconnect and buffers identical from the clock signal source to the clocked register of each clock path. Thus, each clock path has practically the same delay and exhibits good tracking across PVT variations.



Figure 2.13: Tree structures: a) H-tree; b) X-tree; c) binary tree; and d) clock mesh or grid.

H-trees (or X-trees) can efficiently and symmetrically cover large areas due to its simple regular pattern. One important characteristic of these trees is that by continuing to expand the buffer hierarchy, they are capable of delivering the clock to all part of the silicon die in both the horizontal and vertical dimensions. Unfortunately, floorplan constraints often lead to non-ideal driver placements and loss of performance. Because binary trees provide higher flexibility in buffer placement and routing, they are usually preferred over H-trees. Spine clock distribution is a specific implementation of a binary tree, also sometimes called a one dimensional mesh or grid. With a clock spine, the clock signal can be transported in a balanced fashion across one dimension of the die with low structural skew, although with a significant power consumption. Moreover, its path redundancy makes it less susceptible to the effects of variability [117].

Name	Ref	Frequency	skew	Technology	skew/ T_{clk}	Distribution Style	Deskew
			[ba]	լլույ	[/0]	Style	
Itanium	[33]	800	28	180	2.24%	H-Tree/Grid	Yes
Pentium4	[36]	>2000	16	180	3.20%	Spine/Grid	Yes
Itanium2	[118]	1000	52	180	5.20%	Asymmetric Tree	No
Power4	[31]	>1000	25	180	2.50%	Tree/Grid	No
Itanium2	[119]	1500	24	130	3.60%	Asymmetric Tree	Yes
Power5	[120]	>1500	27	130	4.05%	H-Tree/Grid	No
Banias Mobile	[121]	>1500	32	130	4.80%	Spine/Grid	Yes
Pentium4	[115]	3600	7	90	2.52%	Recombinant tile	Yes
Itanium2	[122]	>2000	10	90	2.00%	Asymmetric Tree	Yes
Xeon	[123]	3400	11	65	3.74%	Tree/Grid	Yes
Opterom	[124]	2800	12	65	3.36%	Tree/Grid	-
Power6	[125]	5000	8	65	4.00%	H-Tree/Grid	Yes
Merom	[126]	3000	18	65	5.40%	Tree/Grid	Yes
Tukwila	[127]	2400	n.a.	65	_	Asymmetric Tree	Yes
Nehalem	[128]	3200	n.a.	45	_	Multidomain	Yes
Xeon	[129]	3200	21	45	6.72%	Multidomain	Yes
Westmere	[130]	4000	12	32	4.80%	Multidomain	Yes
Poulson	[1]	2000	n.a.	32	-	Multidomain	Yes

Table 2.1: Clock distribution characteristics of commercial processors.

Clock Deskewing

Deskewing schemes are based on the idea that clock skew can be minimised if the size of the distribution network is reduced, because the main variability sources are spatially correlated. This translates into partitioning the chip into individual Synchronisation Domains (SDs), which should be small enough so that conventional clock distribution schemes yield acceptable local skew. The delay error between adjacent SDs is then compensated with Post-Silicon Tunable (PST) clock buffers [29]. Table 2.1 shows the prevalence of Deskewing (DSK) techniques in commercial MPUs, regardless the multitude of clock distribution styles.

According to their operating rate, deskewing systems can be separated into static or dynamic. The former operate only once, during boot time or factory test, while the last operate continuously or periodically during system operation. Fig. 2.14a illustrates a static deskewing system, where one-time-programmable PST buffers are used to adjust clock delays based on data obtained from the Automatic Test Equipment (ATE). To achieve the maximum tuning capability with minimal hardware cost, different techniques have been proposed [131], [132], [133]. However, they all rely on complete controllability and observability with ATE, which is often difficult and costly [134].

Because PVT variations may change over time, an initial single clock adjustment may not suffice over the device's lifetime usage. A better approach is to resort to tuning loops that self-monitor clock delay mismatches and appropriately adjust their tunable buffers during normal system operation [135], [136]. This type of dynamic (or active) deskewing is represented in Fig. 2.14b. Next to each PST buffer there is a controller to measure skew and generate the appropriate tuning information for delay adjustment, allowing to compensate for dynamic skews that fall within the circuit bandwidth.



Figure 2.14: Deskewing schemes with: a) static tuning during factory test and calibration; and b) dynamic tuning during circuit operation.

The usage of PLL and DLL circuits to implement deskewing circuits was first proposed in [137] and [138], respectively. Since then, multiple schemes have been proposed with either structures. However, DLL based schemes are more common because PLLs have a longer lock-in time and higher power and area overheads. Skews induced by static or quasi-static variability sources (e.g., process variability, circuit defects or temperature gradients) can be mitigated with simple circuits, because the delay adjustment is performed only once or with a coarse periodicity [50], [134]. On the contrary, to cope with dynamic variability, these circuits have to operate continuously and fast [139]. This introduces an additional risk of creating new timing critical paths and render the circuit unstable and thus, dynamic DSK is used only when there are stringent floorplan and power limitations that preclude the usage of clock grids [130].
Multidomain Clock Distribution

To reduce the design time in modern VLSI systems, it is essential to reuse verified and tested IP blocks. However, the integration of various IP cores usually requires a multiclock domain design. It typically embodies multiple islands operating synchronously, served by independent clocks and dedicated interfaces to manage inter-domain communications. This provides functional flexibility, as each of the domains can operate at the optimal frequency, and minimises the complexity and power associated with distributing a low-skew clock to the entire die [140].

Multidomain CDNs belong to a class of designs called Globally Asynchronous Locally Synchronous (GALS) systems, and are typically found in multicore processors and SoCs [29]. A generic illustration of the GALS design style is shown in Fig. 2.15a, where multiple clock domains are embedded in a single silicon die. The chip may receive multiple copies of the system clock and use multiple PLLs to generate the clocks for each synchronous unit. According to the relationship between these clocks, the system can be categorised as: a) mesochronous, when there is a single synchronous unit but its clock distribution network has non-constant delay offset among branches; b) plesiochronous, when there are multiple SDs with a nominally identical frequency; or c) heterochronous, when SDs have different operating frequencies.

Fig. 2.15b shows the global mesochronous clocking technique used in an integrated Network-on-Chip (NoC) architecture containing 80 tiles arranged as an 8 × 10 array of floating-point cores and packet-switched routers [141]. Communication across tiles is made asynchronously, while each tile operates synchronously. The on-chip PLL output is routed using horizontal and vertical spines. Within each tile, the clock is distributed using a balanced H-tree. An example of a plesiochronous clock distribution system is shown in Fig. 2.15c, where independent clock frequencies and distribution styles are used in each domain [123]. Finally, Fig. 2.15d shows a heterochronous distribution [124]. It includes independent PLLs for the cores, un-core, and the Input/Output (I/O) interface blocks, that are capable of operating at different frequencies. Clock domain crossing is accomplished with low-latency First In First Out (FIFO) buffers. Similar schemes can also be found in the most recent high-performance MPUs [127, 128, 130].



Figure 2.15: Multidomain clock distribution: a) generic GALS; b) Intel TeraFlops MPU; c) Intel dual-core Xeon MPU; and d) AMD quad-core Opteron MPU.

2.5 Final Remarks

THIS chapter provided a compact overview on key subjects related to timing in synchronous systems. The emphasis was on concepts, models and techniques that will be referred later on this thesis, but it also covered a broader spectrum of related subjects. It started identifying key timing parameters, performance metrics and variability sources. Then, a brief review of timing models and simulation techniques was provided. Some of these models and techniques are latter used in this thesis. Finally, the fundamental background on clock generation and clock distribution was introduced for the reader's convenience.

From this brief overview, three fundamental ideas should be retained. First, the increasingly complex structure and manufacturing process of digital VLSI systems has been and will continue to be an impairment to clock precision. Each challenge overcome by the IC industry and designers creates new opportunities to shrink device dimensions and increase circuit complexity, which further contribute to increase the number and impact of uncertainty sources. Second, accessing the performance and reliability of synchronous systems is an increasingly complex task as the impact of those sources becomes increasingly difficult to analyse. Finally, it should be noted that although loosely synchronous styles can alleviate the clock uncertainty problem, they are not a definite solution.

In loosely synchronous systems, clock domains of the same frequency can be crossed over synchronously using simple deskewing devices while clock domains of different frequencies can be crossed over asynchronously, using FIFO registers. However, these devices introduce a global latency penalty that gets worse when the clock cycle shrinks. Thus, even with these design styles, minimising clock uncertainty can increase the overall system performance. On the other hand, the design of individual SDs still relies on the synchronous paradigm, using hybrid clock distribution trees with passive or active clock deskewing units. This means that GALS are also affected by the fundamental performance limits imposed by clock precision. This thesis proposes models for jitter insertion and accumulation in clock distribution networks, which can be used to explore those performance limits in synchronous and loosely synchronous designs.

Chapter 3 Uncertainty in Clock Repeaters

Clock repeaters are used in digital synchronous systems with two different purposes - to amplify the clock signal or to introduce intentional delay. The designer can choose from a large variety of physical implementations, depending on the desired performance. Traditional performance metrics include the repeater's delay, power consumption and implementation area. Time uncertainty is known to be roughly proportional to the cell's propagation delay, but there is no practical means to accurately quantify this relationship. This chapter proposes two different models to predict uncertainty in clock repeaters: a circuit model for reference inverters and a scalable model for general repeaters with RC interconnects.

3.1 Clock Repeaters

PROPAGATION delay through conventional clock repeaters depends on their size and spacing and cannot be manipulated once the chip is manufactured. These repeaters are here called Static Delay Repeaters (SDRs). In the last decade, Post-Silicon Tunable (PST) clock repeaters have gain popularity, as their propagation delay can be statically or dynamically manipulated to compensate for PVT variations [142]. As opposed to SDRs, they are hereafter referred as Tunable Delay Repeaters (TDRs). Besides being used as amplification stages in clock distribution networks, both SDRs and TDRs are basic building blocks of other clocking systems, such as Delay Locked Loops (DLLs) [143], Phase Locked Loops (PLLs) [144], Digitally Controlled Oscillators (DCOs) [145, 146], Dynamic Random Access Memory (DRAM) interface units [147], Deskewing (DSK) circuits [148] or spread-spectrum clock generators [149], to name a few. This section describes their typical architecture, discusses implementation trade-offs and evaluates their time precision. Although analog TDRs have been widely used in the past and are still used in some applications for their simplicity and precision [150], only all-digital implementations will be discussed here because they can provide more robust operations over PVT and loading effects, with the benefit of portability across multiple processes.

3.1.1 Static and Tunable Delay Repeaters

Clock repeaters may be symmetric or asymmetric, balanced or unbalanced, inverting or non-inverting. Symmetric repeaters have equal rising and falling switching times ($t_r = t_f$), while balanced repeaters have similar input and output switching times ($t_{in} = t_{out}$). Balanced symmetric repeaters can thus be characterised by a single switching time parameter, t_{sw} . When the repeater is not balanced nor symmetric, t_{sw} can still be used to represent the mean between input/output and rise/fall transition times (3.1).

$$t_{sw,in} = (t_{r,in} + t_{f,in})/2$$
; $t_{sw,out} = (t_{r,out} + t_{f,out})/2$; $t_{sw} = (t_{sw,in} + t_{sw,out})/2$ (3.1)

Inverting repeaters are usually implemented with basic inverters or NAND gates. Inverters are the most common as they provide the shortest delay of any digital gate. This is useful to implement high frequency oscillators, provide fine grain delay control in DLLs or implement low uncertainty clock repeaters. If non-inverting operation is required, tapered clock buffers are the most usual choice, for their short propagation delay and low power consumption. In these clock buffers, the ratio of the second inverter size to the size of the preceding inverter is called the tapering factor (ζ). Long tapered buffers (a chain of inverters of gradually increasing size) are common when driving large off-chip capacitive loads, but cannot be considered general on-chip clock repeaters. Thus, in this thesis, tapered buffers are always considered to include only two cascaded inverters.

In Fig. 3.1 the circuit and transistor level representations of these SDRs are shown. Next to each transistor, there is an indication of its size in terms of channel width (W) and length (L). The size of the N-Channel Metal Oxide Semiconductor (NMOS) transistor in the inverter gate is considered the reference when comparing with other transistors and thus, 1/1 means that W_n/L_n are reference values. In the inverter gate, the size of the P-Channel Metal Oxide Semiconductor (PMOS) transistor is 2/1, so its channel length is the same as in the NMOS ($L_p = L_n$) but its width is two times the width on the NMOS ($W_p=2W_n$). The NAND gate is usually designed to deliver the same output current as the inverter. Hence, the represented gate has similar PMOS transistors and NMOS transistors, with W = 2L. Finally, the buffer has the same input capacitance as the inverter and is represented with a generic tapering factor ζ .



Figure 3.1: Static Delay Repeaters: a) inverter gate; b) NAND gate; c) tapered buffer.

Propagation delay in these gates depends not only on their load but also their logic function. Using the method of logical effort [151], any gate delay can be modelled in terms of a basic delay unit (τ), particular to that process. Being τ the delay of an inverter driving and identical inverter with no parasitics, the absolute propagation delay in a logic gate (t_d) can be expressed as the product of a dimensionless gate delay (d) and τ . This delay is comprised of two components: the parasitic delay (p), which is an intrinsic component and can be found by considering the gate driving no load; and stage effort (f), which depends on the load. The stage effort can be further divided into two components: a logical effort (g), which is the ratio of the input capacitance of a given gate to that of an inverter capable of delivering the same output current; and an electrical effort (h), which is the ratio of the input capacitance of the load to that of the gate. The electrical effort is also commonly called the gate's fanout. These relationships are equated in (3.2).

$$t_d = \tau \cdot d = \tau \cdot (p+f) = \tau \cdot (p+g \cdot h) \tag{3.2}$$

Considering the reference inverter in Fig. 3.1, the NAND gate has a logical effort g = 4/3 in each input and a parasitic delay twice as large as the inverter's. This means that for the same fanout, the NAND gate has a larger propagation delay. However, it has a significant advantage over inverters: it provides two point-of-entry control signals. This is an interesting feature in many applications, like clock gating, to multiplex clock signals at different rates or to implement Digitally Controlled Delay Lines (DCDLs).

SDRs are usually designed with symmetric transitions. However, in circuits with single-edge triggered flip-flops (where a 50% duty-cycle clock is not mandatory), it is possible to design asymmetric gates that focus the majority of their drive current on the critical clock edge. Single-Edge Clock (SEC) inverters, as proposed in [152], have been shown to reduce latency and uncertainty in clock distribution networks. They are designed to have the same size ($W_p + W_n$) as typical symmetric inverters (Invt), but variable PMOS to NMOS width ratios ($\beta = W_p/W_n$). Thus, they can be used as drop-in replacements of symmetric inverters. Fig. 3.2 shows two SEC inverters that can be used to replace a symmetric inverter with $\beta = 3$, along with their output rise/fall times obtained for a 90nm technology. It can be observed that both clock edges, travelling through a cascade of Invf/Invr gates, will experience balanced and symmetric transitions: the critical has a $\approx 10ps$ transition time; while the neglected edge has a $\approx 30ps$ transition time.



Figure 3.2: Invr and Invf SEC inverters, used as drop-in replacements of a symmetric inverters (Invt), and their output rise/fall times.

In contrast to SDRs, TDRs can be configured to exhibit a controllable amount of propagation delay. TDRs can be divided in three categories, according to their operating principle: Variable Resistor Inverters (VRIs) [153], Current-Starved Inverters (CSIs) [154], and Shunt-Capacitor Inverters (SCIs) [155]. Figure 3.3 illustrates their symmetric architectures with 3 binary weighted controlling transistors, starting with a minimum-sized unit switcher ($\times 2^0$). The number of controlling elements depends on the desired number of different separate delays and the required delay resolution. Each cell is represented with an additional output inverter, commonly used to restore the output signal's integrity.





Figure 3.3: Digital voltage controlled TDRs: a) CSI; b) VRI; c) SCI type 1 and type 2.

Symmetric VRIs are built with a static inverter, a series-connected NMOS pull-down stack and a PMOS pull-up stack. Control stacks use transistor arrays in which multiple rows are allowed. Nevertheless, single-row stacks are more common due to their simplicity (Fig. 3.3b). By applying a specific binary vector to the controlling transistors, different pull-up and pull-down resistances are produced and thus, different delays. However, the delay is not only influenced by the resistance of the controlling transistors. It also depends on the capacitance seen by the supply nodes of the first inverter. Thus, increas-

ing the length of a controlling transistor may not increase the circuit's delay. Its higher capacitance increases the charge sharing effect that causes the output capacitance to be charged/discharged faster. This induces a non-monotonic behaviour of the delay with respect to the input vector, which is one of the main drawbacks of VRIS.

On the contrary, CSIs can be easily designed to exhibit a monotonic behaviour using the method proposed in [156]. As shown in Fig. 3.3a, the delay is controlled by the current passing through transistors M5 and M8 (M8 controls the inverter's fall time while M5 controls its rise time). The current passing through these transistors is determined by I_c , which depends on the size of controlling transistors M0-M2 and on the digital input vector. Note that M3 is always ON and thus, determines the repeater's maximum delay. As for VRIs, if the controlling transistors are binary weighted, the circuit can implement 2^N different delays with *N* controlling transistors. However, VRIs need equal PMOS and NMOS stacks to control both rising and falling edges, while CSIs can vary both edges at the expense of only three more transistors (M4, M5 and M7). The main drawback of this circuit is its power consumption, which has a significantly high static component. Adequately sizing the controlling transistors may reduce static power consumption, but it increases the circuit's susceptibility to interference [156].

With a simpler design, SCIs are built with a bank of capacitive loads connected to the output node of a basic inverter. If the inverter is symmetric, so are the output rise and fall transition times. This means that there is no design overhead to obtain symmetric transitions. The most common designs are depicted in Fig. 3.3c, which will hereafter be called SCI type 1 (SCI1) and SCI type 2 (SCI2) configurations. In SCI1, shunt capacitors are switched on and off with transmission gates [157] while SCI2 employs NMOS capacitors with shunted source and drain terminals [158]. Compared to SCI1, SCI2 design is more adequate for small delay steps as it consumes less area, power, and can be designed to exhibit finner delay resolutions.

3.1.2 Uncertainty in Basic Inverters

In this section, clock uncertainty in CMOS inverters is evaluated using circuit simulation. A 90nm minimum length symmetric inverter is used, with $L_n = L_p = 100nm$, $W_n = 1\mu m$ and $W_p=3\mu m$. This inverter is here called the reference repeater because the performance of other SDRs and TDRs will be latter compared to this inverter's performance, using the same simulation framework.

Transient noise simulation was performed with SPECTRE, using a 50% duty-cycle clock waveform as signal source and a single capacitance as load ($C_L = h \cdot C_u$), as shown in Fig. 3.4a. The slew-rate was configured to guarantee balanced transitions and the unit load (C_u) chosen as the one that produces the same delay as the delay shown by an inverter at the middle of a long fanout-of-one (FO1) inverter chain (Fig. 3.4b). Because the load of an FO1 inverter is equal to its own input capacitance, C_u can be considered to be equal to C_{in} . Thus, C_L is as a multiple of C_{in} and h is the inverter's fanout.



Figure 3.4: Inverter: a) test circuit; and b) circuit to extract C_{in}.

Timing parameters were obtained, following their usual definitions: delay (t_d), was measured as the average of the time difference between input and output reaching 50% of V_{dd} , for rise and fall times; switching time (t_{sw}), was measured as the average between t_r and t_f ; and absolute jitter (σ_{t_d}), was obtained as the average of rising and falling standard deviation of delay, in the presence of TCN, PSN, Intra-die Process Variability (IPV) and temperature variations. Simulations were performed with $T_{clk} = 20t_{sw}$, to guarantee the clock signal's integrity, and $T = 27 \,^{\circ}$ C (room temperature) unless otherwise noted.

To evaluate TCN induced jitter, a transient noise simulation tool available in Analog Design Environment (ADE) from Cadence has been used. It allows white noise samples to be generated at each simulation step, with a variance determined by each transistor's bias conditions and simulation temperature. This results in time-dependent, zero mean, random noise current sources being considered in parallel with each transistor's channel. Several parameters may be configured as described in Table 3.1. The configuration used in these simulations is also shown and justified.

Parameter	Description	Value	Justification
noisefmax	Bandwidth of pseudo-random noise sources. A non-zero value turns ON the noise sources during transient analysis.	0.5/p ⁽¹⁾	This is the knee frequency for typ- ical digital signal shapes, which is not too far beyond the inverter's in- trinsic -3dB bandwidth [159].
noisescale	Noise scale factor applied to all generated noise.	10 (2)	This gain used to artificially inflate the small TCN and make it visible, above transient analysis numerical noise floor.
noiseseed	Seed for the random number gen- erator.	1	The same seed has been used across simulations, to compare the re- peater's performances under the same circumstances.
noisefmin	The power spectral density of the noise sources depend on frequency in the interval from noisefmin to noisefmax.	noisefmax (default)	In this case, only white noise is con- sidered.
noisetmin	Time interval between noise source updates.	1/noisefmax (default)	Smaller values would produce smoother noise signals, but would reduce time integration step.

Table 3.1: Transient noise analysis configuration parameters.

(1) *p* is the repeater's parasitic delay.

(2) This gain can be disregarded as results were back-scaled to correspond to the real performance.

To evaluate PSN induced jitter, simple transient simulations were performed with independent random Gaussian noise sources in power and ground rails (MMN). Several sets of 5000 noise samples were generated in MATLAB and imported into SPECTRE as piece-wise linear voltage sources with configurable noise gain and step (T_n). Fig. 3.5a shows the impact of different PSN levels ($v_n = \sigma_{psn} / V_{dd}$) on jitter insertion, for a FO4 inverter with $T_n = T_{clk}$ and $T_n = 4T_{clk}$. It shows that jitter grows almost linearly with v_n if it is small, and exponentially if it is large. Hereafter, only small PSN levels will be considered (< 10%), as it is the most common scenario in well designed ICs. Thus, jitter can be considered to depend linearly on PSN magnitude, as is usually observed in practise [160]. In Fig. 3.5b, jitter is shown as a function of the noise cut-off frequency ($f_n=1/T_n$). It shows a resonance peak for $f_n = f_{clk}$ and again for $f_n = 2f_{clk}$. In contrast, for $f_n < 0.25f_{clk}$, jitter is almost constant. Because PSN is usually considered to have a low-frequency spectrum compared to the clock frequency, the noise step will be hereafter set to $T_n = 4T_{clk}$.

The system being here simulated can be classified as terminating¹ if an event can be

¹In general, the run-length of a transient simulation depends on the system nature. In a terminating system, the run-length is fixed by specification or by an event definition that marks the end of the simulation. The simulation goal is to understand system behaviour for this fixed duration. On the other hand, a non-



Figure 3.5: PSN jitter in the reference FO4 inverter, for different: a) noise levels ($v_n = \sigma_{psn}/V_{dd}$); and b) cut-off frequencies ($f_n = T_n$).

specified to mark the end of simulation. If jitter could be calculated during the simulation run, the event could be the time instant for which a given confidence level was reached for the chosen performance metric. Unfortunately, absolute jitter can only be calculated after the simulation run and thus, a fixed simulation time (T_{sim}) had to be imposed. Because the standard deviation's accuracy is directly proportional to sample size, it is only possible to reach a reasonable value for T_{sim} by inspection of simulation results.

Figures 3.6a and 3.6b show the inverter's TCN and PSN jitter evolution for growing sample sizes ($N = T_{sim}/T_{clk}$). Both have shown to follow inverse exponential functions towards a reasonably constant final value, although PSN jitter took a longer time to do that. To have accurate results within a reasonable simulation time, the simulation run length was set to one thousand clock cycles for TCN jitter (N = 1000) and three thousand for PSN jitter (N = 3000). IPV jitter was evaluated with Monte Carlo simulation, which also usually requires thousands of simulation steps until enough delay values are obtained. Fortunately, screening experiments have shown that a reasonable number of runs could be used in such simple structures as clock repeaters. Fig. 3.6c shows IPV jitter in the reference repeater for an increasing number of runs and the correspondent simulation time (compared to the time needed for 50 runs). A good compromise between accuracy and simulation time was found to be around 200 runs.

terminating system is in perpetual operation and the goal is to understand its steady-state behaviour.



Figure 3.6: a) TCN jitter vs. sample size (N); b) PSN jitter vs. sample size (N); c) IPV jitter and simulation time vs. MC runs.

Jitter simulation results are shown in Fig. 3.7a for a balanced reference repeater, with $\sigma_{psn} = 10\% V_{dd}$ and increasing fanouts (FO*h*, *h* = 1..6). Jitter is shown to increase linearly with fanout for all sources, but with different rates. For this fanout range, TCN jitter increases $1.1\times$, while PSN and IPV jitter increase $3.2\times$ and $2.6\times$, respectively. TCN jitter grows slower with fanout because the high-frequency noise components are affected by the low-pass filtering imposed by C_L . It is also shown to be much smaller (two orders of magnitude) than PSN or IPV jitter. Yet, TCN jitter will not be neglected in this section as it represents a fundamental limit on dynamic timing precision. On the contrary, PSN and IPV jitter have the same order of magnitude for this PSN level.

The impact of different noise modes has also been evaluated. Simulations were repeated using random noise sources in power and ground rails, with different modes but the same magnitude ($\sigma_{dmn} = \sigma_{cmn} = \sigma_{mmn}$). Fig. 3.7b shows that jitter induced by CMN sources is higher than for DMN sources, while jitter induced by MMN sources (independent noise sources in power and ground rails) falls between CMN and DMN bounds.

The impact of temperature on PSN, TCN and IPV jitter can be observed in Fig. 3.8a, for an FO4 inverter with $0 \degree C \le T \le 100 \degree C$. Values are normalised to jitter measured at room temperature ($T = 27 \degree C$). As expected, temperature has a significant impact on TCN jitter. A variation of 21% was measured in TCN jitter values while PSN and IPV jitter varied less than 5%, in the specified temperature range.



Figure 3.7: Jitter in the reference inverter, for different fanouts and: a) PSN, TCN and IPV sources; and b) CMN, DMN and MMN sources.



Figure 3.8: Jitter in the FO4 reference inverter, for: a) different operating temperatures; and b) unbalanced transition times.

Finally, the FO4 inverter's performance for unbalanced transitions was evaluated. Simulation results are shown in Fig. 3.8b. When $t_{in} \leq t_{out}$, jitter is not very much affected by the input transition time. On the contrary, it increases faster when $t_{in} \gg t_{out}$, following the typical output transition time behaviour under fast and slow input transitions [161]. Thus, a good design practise to achieve low clock uncertainty is to keep balanced transitions in clock repeaters. However, if clock repeaters have internal unbalanced cells (like tapered buffers or TDRs), their performance is inevitably affected by this effect. For example, in a tapered buffer with high ζ (please refer to Fig. 3.1), the smaller jitter gen-

		Tim	e [ps]]	Jitter [ps]			Un	certainty	Power	Area	
SDR		t_d	t_{sw}	PSN	TCN	IPV		PSN	TCN	IPV	[µW]	$[L_{sq}]$
T.	FO1	14	16	1.30	0.023	0.75		9.02%	0.159%	5.21%	5.40	22.4
Inv	FO4	39	49	3.54	0.039	1.92		9.20%	0.102%	4.98%	14.5	33.4
SEC Inv	FO1	11	14	1.04	0.014	0.71		8.44%	0.126%	6.27%	33.0	22.4
Invr	FO4	31	42	2.80	0.026	1.94		9.15%	0.087%	6.35%	29.4	55.4
SEC Inv	FO1	7	7	0.57	0.007	0.28		8.23%	0.116%	4.26%	33.0	22.4
Invf	FO4	17	20	1.52	0.016	0.70		8.84%	0.093%	4.07%	29.4	33.4
	FO1	20	24	1.94	0.027	1.14		9.61%	0.138%	5.80%	7.70	02.2
INAIND	FO4	47	62	4.60	0.045	2.75		9.75%	0.095%	5.83%	18.7	03.3
Buffer	FO1	29	18	2.00	0.038	1.62		6.89%	0.132%	5.63%	10.9	667
$\zeta = 1$	FO4	52	47	3.79	0.056	2.77		7.33%	0.108%	5.35%	19.9	00.7
Buffer	FO1	42	21	2.84	0.046	2.75		6.78%	0.109%	6.55%	26.0	
$\zeta=4$	FO4	50	27	3.28	0.048	3.08		6.58%	0.096%	6.18%	35.5	166.7
	FO16	75	55	4.88	0.057	4.10		6.51%	0.076%	5.47%	72.4	

Table 3.2: SDRs performance metrics with $\sigma_{vsn} = 6\% V_{dd}$.

erated in the first inverter (with $t_{in} < t_{out}$) will not fully compensate for the higher jitter generated in the second cell (with $t_{in} > t_{out}$).

3.1.3 Performance Comparison

This section evaluates the performance of SDRs and TDRs, using the same simulation framework described for the reference inverter. Table 3.2 compares timing, precision, area and power metrics, using $\sigma_{psn} = 6\% V_{dd}$, $T_n = 4T_{clk}$, T = 27 °C and $t_{in} = t_{out}$. Because SDRs have different nominal delays, both jitter and delay uncertainty (jitter as a percentage of propagation delay) are shown. These results were obtained from circuit simulation only, as different layout styles could influence the comparison. Thus, implementation area values are given in terms of logical squares (L_{sq}), which correspond to units of a minimum-size NMOS transistor in this technology (L_n =100nm and W_n =120nm). Power consumption is evaluated as the average power consumed per clock cycle, with f_{clk} =500MHz.

For each SDR, the table presents simulation results for FO1 and FO4 repeaters. It also includes results for an FO16 buffer with $\zeta = 4$, in which inverters have balanced transitions (both are FO4 inverters). A higher fanout corresponds to higher jitter, delay,

transition times and power in all repeaters. However, when these cells are used to insert a given amount of delay, precision is best evaluated with the uncertainty metric. A higher fanout is shown only to slightly increase PSN uncertainty and to have a beneficial impact on both TCN and IPV uncertainty. This results from the fact that PSN was defined as having a low-frequency spectrum and thus, it is not affected by a lower repeater's bandwidth.

Regarding different topologies, SEC inverters are shown to have a better timing performance than symmetric inverters (especially the Invf), at the cost of higher power consumption. Comparing buffers with inverters, it can be observed that although inverters have lower absolute jitter, dynamic uncertainty is smaller in buffers. Also, uncertainty is smaller in buffers with higher ζ values. This means that for the same total delay, buffers with high ζ have higher precision than simple inverters or low ζ buffers. Also, better results are achieved if balanced transitions are guaranteed, i.e., for fanouts equal to ζ^2 .

The NAND gate repeater seems to have no significant advantage compared to the inverter, because its most important feature is not evident in Table 3.2. The NAND gate has the benefit to provide two point-of-entry control signals which can be used to build compact DCDLs. For that particular application, inverter/buffer based SDRs have to be associated to a multiplexer, which increases the fanout of each cell and introduces more uncertainty. Therefore, the FO1 NAND gate should be compared to an FO3 or FO4 inverter, depending on the multiplexer design. In this case, NAND repeaters may be a good alternative to inverter/buffer based SDRs with the advantage of allowing very regular DCDL designs.

The performance of different TDRs was also compared, using the same simulation framework. The repeater's design followed the architecture depicted in Fig. 3.3, using reference inverters. To allow a fair comparison, TDRs were designed to have similar maximum and minimum delays, using transistor sizing. However, due to the charge-sharing effect, this technique could not be used in VRI and SCI1 repeaters. In these repeaters, an extra static NMOS capacitance was added to the output of the first inverter to increase their minimum delay. Table 3.3 shows the size of transistors used in the controlling structures of each TDR (refer to Fig. 3.3). In SCI1, transmission gates (TGi) are used to control the bank of NMOS capacitance (Mi). Transistors M3-M4, M5-M6 and M7-M8 correspond

				,		0			1		0
Repeater	Size [nm]	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9 ⁽¹⁾
CSI	L	700	700	700	100	100	100	100	100	330	
	W	1320	660	330	330	1000	1000	330	330	330	
VRI	L	1000	500	500	120	300	700	700	1400		1000
	W	750	750	1500	750	250	500	250	250		1200
	L	2000	2000	2000	100	100	100	100	100	100	2000
5011	W	700	1400	2800	700	700	1400	1400	2800	2800	2000
SCI2	L	2000	2000	2000							
	W	700	1400	2800							

Table 3.3: Transistor sizes in TDRs, following the structures depicted in Fig. 3.3.

(1): NMOS capacitance added to the first inverter's output node to increase the cell' minimum delay.

				1					P			
		Timir	ıg [ps]		Jitter [ps]		_	Une	certainty [Power	Area	
TDR	$b_2 b_1 b_0$	t _d	t_{sw}	PSN	TCN	IPV		PSN	TCN	IPV	[µW]	$[L_{sq}]$
CCI	000	113	50	16.9	0.343	8.8		15.0%	0.303%	7.8%	216	220
CSI	111	183	78	31.0	0.704	17.9		16.9%	0.385%	9.8%	114	229
VDI	000	105	48	8.7	0.127	6.3		8.2%	0.121%	5.9%	20	410
VKI	111	195	77	17.5	0.257	11.9		9.0%	0.131%	6.1%	21	410
SCI1	000	111	47	7.9	0.104	8.9		7.1%	0.093%	8.0%	46	1208
5011	111	196	79	10.8	0.181	10.8		5.5%	0.092%	5.5%	129	1290
SC17	000	111	41	6.1	0.104	7.0		5.5%	0.094%	6.3%	48	883
5012	111	196	76	13.6	0.155	11.6		6.9%	0.079%	5.9%	90	000

Table 3.4: TDRs performance metrics with $\sigma_{psn} = 6\% V_{dd}$.

to TG0, TG1 and TG2, respectively, while M9 corresponds to the NMOS capacitance used in VRI and SCI1 repeaters to increase their minimum delay (not shown in Fig. 3.3).

Table 3.4 presents the same performance metrics shown for SDRs, for maximum and minimum input vectors (maximum and minimum delays). SCI cells show the best jitter and uncertainty performance, at the cost of higher implementation area. Compared to each other, SCI1 performs better for large delays while SCI2 is better for small delays. In fact, the SCI1 is the only TDR for which uncertainty decreases with increasing delay. Thus, the SCI1 is more adequate to insert coarse delays while SCI2 is best fitted to implement fine tuning TDRs. On the other hand, VRIs and CSIs have shown to be more sensitive to dynamic variations. To increase the CSI robustness, authors in [162] proposed an alternative design where the controlling transistors are replaced by current sources. Yet, it further increases its power consumption, which is already high.

In TDRs, linearity is also an important figure of merit. To evaluate this feature, Fig. 3.9 shows the repeater's delay and power consumption for all possible input vectors. The SCI2 repeater shows a good delay linearity and a reasonable power consumption compared to others. The VRI is the one with lower power consumption, but is not very linear with the input vector. Moreover, it has to be carefully designed due to the charge-sharing effect. The impact of charge-sharing is also clearly observable in the SCI1 repeater delay, when the largest controlling transistor is turned on ($b_2 = 1$). As expected, the worst power consumption of all is shown by the CSI, specially for small delays.



Figure 3.9: Performance metrics for CSI, VRI, SCI1 and SCI2 repeaters, with respect to input vector ($b_2b_1b_0$): a) delay; and b) power consumption; for f_{clk} =500MHz.

At this point, two comments are due regarding uncertainty in clock repeaters. First, simulation results show that absolute jitter increases for higher fanouts in SDRs, and higher input vectors in TDRs. However, gate delay seams to increase almost by the same amount, which reduces the uncertainty variability in each structure (at least for the most significant jitter sources - PSN and IPV). This means that uncertainty cannot be significantly reduced by manipulating the repeater's fanout. Second, except for the CSI repeater, results have shown that uncertainty variability is also small among SDRs and TDRs. The mean values for PSN, TCN and IPV are 7.8%, 0.11% and 5.77% with standard deviations equal to 1.35%, 0.02% and 0.86%, respectively. Thus, when clock repeaters are used to insert delay in the clock path, time precision is not significantly dependent on their particular design.

3.2 Reference Inverter Jitter Model

D IFFERENT variability sources affect the repeater's precision in different ways. This section presents heuristic expressions to determine the inverter's sensitivity to intrinsic and environmental jitter sources, and identify the key parameters on which they depend. A symmetric inverter repeater is here considered, for two main reasons. First, its low gate complexity allows enables the identification of key parameters involved in jitter insertion and development of tractable models; second, knowledge of inverter properties leads to knowledge of larger gates and more complicated clock repeaters. Simulation results presented here were obtained with the same simulation framework and reference repeater design described in section 3.1.2.

3.2.1 Circuit Parameters

Section 2.3.2 presented the most popular jitter model for digital gates, the First Passage Time (FPT) model. For the reader's convenience it is reproduced in (3.3), where σ_{v_n} is the gate's output voltage noise and *SR* is its slew-rate. Although slew-rate can be represented by the gate's effective switching current (I_{eff}) divided by its effective load capacitance (C_{eff}), it is not known if these parameters can be used to obtain accurate dynamic jitter predictions, nor which of the existing I_{eff} models provide the better results.

$$\sigma_{t_d} = \sigma_{v_n} \cdot (1/SR) \tag{3.3}$$

Fig. 3.10 shows the reference inverter's output waveforms for fast and slow rising input signals. The output discharging current is also shown. These results were obtained for the reference inverter, with h = 4 and a ramp input clock source. Similar plots could be obtained for a falling input and/or other fanouts. The black time intervals correspond to idle periods of time, here called the inverter's *rest* state. In this state, one transistor is conducting in the ohmic region and the other is at cut-off. When $V_{in} \ge V_{th,n}$ the NMOS starts conducting and the inverter starts switching (for these transistors $V_{th,n} \approx V_{dd}/2$). The grey time interval corresponds to the *switch* state, where transistors go through different regions depending on t_{in} , C_L , $V_{th,n}$ and $V_{th,p}$. When $V_{out} \le 0.1V_{dd}$ the inverter is



considered to be back to the *rest* state. At the threshold crossing ($V_{out} = V_{dd}/2$), one transistor is usually off while the other can be in ohmic or saturation region.

Figure 3.10: Inverter's voltage and current waveforms for: a) $t_{in} < t_{out}$; and b) $t_{in} > t_{out}$.

Considering balanced inverters only, the analysis can be restricted to a single path through operating regions. Fig. 3.11 shows the output voltage and current waveforms for balanced FO1 and FO4 inverters. For both, the input voltage reaches its final value before the output voltage crosses the logic threshold. At the threshold, the NMOS is in the ohmic region and the PMOS is at cut-off. On the other hand, the peak output current (I_p) occurs before the threshold crossing, almost simultaneously with the time when V_{in} reaches V_{dd} . For the FO1 inverter this corresponds to $V_{out} \approx 0.8V_{dd}$ while it occurs for $V_{out} \approx 0.7V_{dd}$ for higher fanouts. For simplicity, and because repeaters are usually designed with balanced transitions, only balanced inverters will hereafter be considered.



Figure 3.11: Inverter's output voltage and current waveforms, for balanced transitions: a) FO1 inverter; and b) FO4 inverter.

Even with this simplifying assumption, the repeater's circuit parameters depend on circuit bias, which changes continuously during the *switch* state. Fig. 3.12 shows different *SR* and I_{eff} simulation results, according to different definitions. Slew-rate was obtained for different intervals of an output transition, while I_{eff} was computed according to different heuristic models. Values obtained from [73], [71] and [75] are represented by $I_{eff,Na}$, $I_{eff,Yo}$ and $I_{eff,Hu}$, respectively. Model results are also compared with the maximum output current (I_p) and the actual I_{eff} , computed as $I_{eff} = C_L \cdot SR_{20/80}$. One can see that no current model, nor I_p , follow the actual I_{eff} . This observation also sustains if a different *SR* definition had been used (e.g., $SR_{10/90}$).



Figure 3.12: Slew-rate and I_{eff} for the reference 90nm inverter, for different a) slew-rate definitions; b) effective current definitions.

The same experiments were repeated for different driving and loading conditions, using the circuits shown in Fig. 3.13. These circuits will be referred as A, B, C and D. Circuit A is the one used so far (with ramp input and constant load capacitance), while circuit B corresponds to the most realistic situation, where the repeater drives and is driven by similar gates. Remember that in circuit A, C_L was chosen as the capacitance that induces the same propagation delay shown by the inverter in circuit B, while t_{in} was manipulated to guarantee balanced transitions throughout simulations. Circuits C and D, correspond to balanced mixed configurations, using the same C_L .

For each circuit, slew-rate was measured using different thresholds. Fig. 3.14a shows that circuit D is the one that best represents realistic conditions for $SR_{10/90}$. However, for



Figure 3.13: Different test circuit configurations: a) ideal driver and load; b) realistic driver and load; c) ideal driver and realistic load; d) realistic driver and ideal load.

other *SR* definitions, circuit A provides a better approximation to the behaviour of circuit B. The effective current, obtained with $SR_{20/80}$, was also evaluated and is shown in Fig. 3.14b. Results show that circuit A is the one that best mimics the most realistic situation (circuit B), at least for fanouts higher than one (typical situation in most repeaters).



Figure 3.14: For the circuits shown in Fig. 3.13, plots show: a) slew-rate for different definitions; and b) I_{eff} obtained with $SR_{20/80}$, for increasing fanouts.

Because the FPT model depends essentially on slew-rate, it is acceptable to assume that circuit B can be replaced by circuit A when evaluating the inverter's jitter performance. Its simplicity reduces the complexity associated with parameter extraction - both C_L and t_{in} become constant parameters during the *switch* state - allowing simple heuristic expressions to be obtained for the inverter's jitter sensitivity.

3.2.2 Intrinsic Variability Sources

Intrinsic variability sources include TCN and IPV, which determine the fundamental dynamic and static circuit precision, respectively. This section starts analysing the circuit's response to TCN and then discusses the impact of IPV. Simulation results for TCN jitter $(\sigma_{t_{d,ten}})$ and uncertainty (U_{ten}) are shown in Fig. 3.15, for the 90nm inverter with different sizes and fanouts. Transistor's width was increased from $W_p = 3W_n = 750$ nm up to $W_p = 3W_n = 3\mu$ m, resulting in sizes from $1 \times$ up to $8 \times$. Large inverters were built with a single finger (continuous lines) or with multiple fingers (dashed lines), to evaluate the impact of different sizing techniques.



Figure 3.15: Performance metrics in 90nm inverters for different sizes and fanouts: a) TCN jitter; and b) TCN uncertainty.

Three relevant observations can be made from these plots. First, smaller inverters have higher TCN jitter. This happens because the slew-rate is proportional to the transistor's drain current (I_{ds}) while TCN (Root Mean Square (RMS)) grows with the root of I_{ds} . Second, while jitter increases with C_L , uncertainty decreases because the inverter's noise bandwidth decreases. This means that highly loaded inverters can be used to generate delays with lower TCN uncertainty than lightly loaded ones. Finally, inverters built with multiple fingers have lower jitter and higher uncertainty, because they have lower output parasitic capacitance. Nevertheless, differences are not very significant.

According to the FPT model, TCN jitter depends on slew-rate and output noise. Using transient noise simulations, TCN was measured at the inverter's output node ($v_{o,tcn}$) for different input DC voltages and different fanouts. Although these simulations can not fully represent the noise behaviour of the switching inverter, it is the only way to measure the inverter's output TCN at the threshold crossing. Fig. 3.16a shows simulation results using the simulation setup described in Table 3.1. They show that $v_{o,tcn}$ is highly dependent on circuit bias, especially for those values of V_{in} that correspond to the *switch* state. On the other hand, the peak noise is shown to occur when $V_{in} = 0.6V$, which corresponds to $0.5V_{dd}$ in this 90nm technology.



Figure 3.16: Results for the reference inverter: a) TCN (RMS) measured at the output node for constant input voltages; b) voltage transfer characteristic.

Fig. 3.16b shows the inverter's Voltage Transfer Characteristic (VTC). When $V_{in} = 0.6V$ (i.e., at $k_i = V_{in}/V_{dd} \approx 0.5$) the output voltage is around $70\% V_{dd}$ ($k_o = V_{out}/V_{dd} \approx 0.7$), which is also the voltage for which the current is maximum (I_p) in a balanced inverter with typical fanouts (refer to Fig. 3.11). Based on this insight and using the FPT model, an expression to estimate TCN jitter is proposed in (3.4), where $\beta_{tcn} = C_L/I_p$ is defined as the inverter's TCN jitter sensitivity factor.

$$\sigma_{t_{d,ten}} = \sigma_{v_{o,ten,max}} \cdot \left(C_L / I_p\right) = \sigma_{v_{o,ten,max}} \cdot \beta_{ten} \tag{3.4}$$

Table 3.5 shows the error between the model predictions and simulation results, as a percentage of simulation results (ϵ_{tcn}), for inverters with different sizes and fanouts. Also shown are the exact input and output voltage parameters (k_i and k_o), used to measure peak noise, and the mean error ($\mu \epsilon_{tcn}$) and standard deviation ($\sigma \epsilon_{tcn}$) for typical fanouts

				ϵ_{tcn}	[%]	Typical	Fanouts				
Size Fingers	Fingers	FO1	FO2	FO3	FO4	FO5	FO6	$\mu \epsilon_{tcn}$	$\sigma \epsilon_{tcn}$	k _i	ko
1×	1	-12.6	-7.47	-1.60	-1.93	2.92	5.55	-0.20%	2.71%	0.458	0.888
$4 \times$	1	0.91	1.51	0.53	0.14	-2.08	-0.24	-0.47%	1.41%	0.475	0.830
$4 \times$	4	-5.77	0.95	-0.65	0.66	2.42	-2.24	0.81%	1.54%	0.442	0.835
$8 \times$	1	0.32	0.60	0.96	-0.78	-0.25	-1.52	-0.03%	0.89%	0.500	0.704
$8 \times$	4	-6.12	0.13	-0.99	1.33	1.94	0.23	0.76%	1.55%	0.446	0.811

Table 3.5: Heuristic TCN jitter model error for 90nm inverters.

(between FO3 and FO5). Although some significant error values were obtained for small inverters and fanouts, the model has shown to provide accurate jitter predictions for typical inverter sizes and fanouts (FO1 and/or minimum-sized inverters are seldom used in practise). This means that β_{tcn} can be considered a good TCN sensitivity metric and $\sigma_{v_{o,tcn,max}}$ a good estimator for the inverter's output noise during the *switch* state.

Sensitivity to static jitter (induced by IPV) can also be shown to depend on the inverter's peak current. Several MC simulations were performed, using the reference 90nm inverter with balanced transitions and increasing fanouts. Static jitter ($\sigma_{t_{d,ipv}}$) was found to depend on I_p and on its variability ($\sigma_{I_{p,ipv}}$) with process parameters. Based on these results, an heuristic model is proposed in (3.5), where β_{ipv} is the inverter's IPV sensitivity.

$$\sigma_{t_{d,ipv}} = \sigma_{I_{p,ipv}} \cdot \left(t_d / I_p \right) = \sigma_{I_{p,ipv}} \cdot \beta_{ipv}$$
(3.5)

The model's error as a percentage of simulation results (ϵ_{ipv}) is shown in Table 3.6, for increasing fanouts. Again, there is a close agreement between model predictions and simulation results. This means that the key parameter determining the inverter's sensitivity to intrinsic variability sources is the peak current (I_p). This means that all design options that contribute to reduce this current will also contribute to increase the repeater's sensitivity to TCN and IPV.

	FO1	FO2	FO3	FO4	FO5	FO6
σ _{I_{p,ipv} [μA]}	16.5	18.5	19.2	19.5	19.9	20.1
$\sigma_{t_{d,ipv}}$ [ps]	0.75	1.14	1.53	1.92	2.31	2.70
ϵ_{ipv} [%]	-1.55	0.46	1.67	1.94	1.37	0.65

Table 3.6: Heuristic IPV jitter model error for the reference 90nm inverter.

3.2.3 Environmental Variability Sources

Power Supply Noise (PSN) and Crosstalk (CRT) are the most relevant environmental variability sources affecting the inverter's timing performance. This section proposes metrics for the inverter's sensitivity to these sources.

Power Supply Noise

Fig. 3.17 shows simulation results for PSN jitter ($\sigma_{t_{d,psn}}$) and uncertainty (U_{psn}) in a 90nm inverter, with the same sizes and fanouts used for TCN jitter evaluation. Results were obtained with $T_n = 4T_{clk}$ and MMN sources with $\sigma_{v_{dd}} = \sigma_{v_{ss}}$.



Figure 3.17: Performance metrics in 90nm inverters for different sizes and fanouts: a) PSN jitter; and b) PSN uncertainty.

Both PSN jitter and uncertainty are shown to increase for higher fanouts, because the significant noise spectral components are well bellow the inverter's bandwidth. Regarding inverter's size, one can see that the minimum-size inverter has worse performance compared to others, as long as they are designed with multiple fingers. Also, inverters with multiple fingers have similar performances while single-finger transistors have increasingly worse performances with sizing (due to their higher parasitic capacitance and resistance). Thus, as long as good design practises are used and minimum-transistor width is avoided, the repeater's size is not a relevant PSN jitter parameter.

To derive an expression for PSN jitter sensitivity, it is first necessary to express the in-

verter's output noise as a function of PSN at the supply rails. Although noise sources in power and ground rails are independent, their contribution to output jitter is not - the impact of power noise in a given signal edge actually depends on the ground noise affecting that edge on the same period of time. Thus, the output noise responsible for jitter $(\sigma_{v_{o,psn}})$ should also be considered as the result of partially correlated power and ground noise sources $(\sigma_{v_{dd}} \text{ and } \sigma_{v_{ss}})$. Because uncorrelated random variables are usually added as variances and correlated variables sum up as standard deviations, the expression in (3.6) is here proposed to define $\sigma_{v_{o,psn}}$.

$$\sigma_{v_{o,psn}} = 0.5 \cdot \left(\sqrt{\sigma_{v_{dd}}^2 + \sigma_{v_{ss}}^2}\right) + 0.5 \cdot \left(\sigma_{v_{dd}} + \sigma_{v_{ss}}\right) \tag{3.6}$$

Using $\sigma_{v_n} = \sigma_{v_{o,psn}}$, the FPT model can now be used to derive a PSN jitter sensitivity metric, as long as an appropriate effective current is considered when computing *SR*. Fig. 3.18a compares I_p with the effective current for which the FPT model predictions match simulation results ($I_{eff,fpt}$), for different FO4 inverter sizes. It also shows the effective current obtained with $SR_{10/90}$, referred here as $I_{eff,sr}$. One can see that $I_{eff,fpt}$ follows $I_{eff,sr}$ better than I_p , meaning that the inverter's PSN sensitivity depends on the mean slew-rate during the entire signal transition and not only on the peak slew-rate (as TCN sensitivity does).



Figure 3.18: Effective current: a) I_p compared with the FPT model's effective current and the one obtained from slew-rate measurement; b) impact of both V_{th} and V_{dd} .

Fig. 3.18b shows that $I_{eff,fpt}$ depends on both V_{th} and V_{dd} . Faster transistors, with

				,					
				Typical	Typical Fanouts				
Size I	Fingers	FO1	FO2	FO3	FO4	FO5	FO6	$\mu\epsilon_{psn}$	$\sigma\epsilon_{psn}$
$1 \times$	1	-11.9	-5.43	-2.10	-0.34	1.01	1.9	-0.47%	1.56%
4 imes	1	-12.5	-6.08	-3.15	-1.50	-0.06	0.67	-1.57%	1.54%
4 imes	4	-13.1	-2.66	0.59	2.52	3.87	4.79	2.33%	1.65%
$8 \times$	1	-14.4	-8.22	-5.62	-3.97	-2.91	-1.97	-4.16%	1.37%
$8 \times$	4	-9.01	-2.19	1.19	3.27	4.30	5.09	2.92%	1.58%

Table 3.7: Heuristic PSN jitter model error for 90nm inverters.

lower V_{th} or higher V_{dd} result in higher effective currents. Based on this observation, an heuristic model for the effective current is proposed in (3.7). It depends on both I_p and on the normalised threshold voltage ($v_T = V_{th} / V_{dd}$). Parameter ξ is a fitting parameter close to one. Using the FPT model and this expression for I_{eff} , the final expression for PSN jitter can be written as shown in (3.8), where β_{psn} is defined as the PSN jitter sensitivity factor.

$$I_{eff} = I_p \cdot (\xi - (v_{th} / V_{dd})) = I_p \cdot (\xi - v_T)$$
(3.7)

$$\sigma_{t_{d,psn}} = \sigma_{v_{o,psn}} \left[C_L / \left(I_p \left(\xi - v_T \right) \right) \right] = \sigma_{v_{o,psn}} \cdot \beta_{psn}$$
(3.8)

Table 3.7, presents the error between model predictions and simulation results, compared to simulation results (ϵ_{psn}). Predictions were obtained using the ξ value that minimises the error, which is 1.2 in these repeaters. The mean error ($\mu\epsilon_{psn}$) and its standard deviation ($\sigma\epsilon_{psn}$) for fanouts between three and five (typical fanouts) are also given. The proposed model has consistently shown to under-estimate jitter for small fanouts and over-estimate jitter for high fanouts. This results from the fact that (3.7) does not depend on fanout, which is not fully realistic. However, the goal was to identify the key circuit parameters in PSN jitter insertion and not to derive an accurate effective current model. Moreover, the I_{eff} model has shown to be much more accurate than previously published models for this purpose. Thus, β_{psn} (3.8) can be considered to be an accurate heuristic PSN sensitivity metric.

Crosstalk

Crosstalk (CRT) is also a significant jitter source in current high-performance digital circuits, changing the inverter's delay according to the switching behaviour of its neighbours. Fig. 3.19a presents a typical victim interconnect with two potential aggressors, where C_c is the coupling capacitance to each neighbour. Here, the total coupling capacitance is $C_{ct} = 2C_c$ and the total ground capacitance (C_{gt}) includes the load capacitance plus the parallel-plate and fringing capacitance between the conductor and the upper and lower planes.



Figure 3.19: Crosstalk induced capacitance variability: a) victim wire with two possible aggressors; b) C_v as a Gaussian variable; c) normalized t_d as a function of normalized C_v .

Following the approach described in section 2.3.2, the victim's capacitance (C_v) is here considered to be a random variable with a Gaussian probability density function. This is shown in Fig. 3.19b, where μ_c and σ_c correspond to the mean and standard deviation of C_v , respectively. When the potential aggressor lines are quiet, the capacitance of the victim's wire (C_v) has a mean value equal to $\mu_c = C_{gt} + 2C_c$. However, C_v may exhibit any value between $C_{v,min} = C_{gt}$ and $C_{v,max} = C_{gt} + 4C_c$ when aggressors switch, according to Miller bounds [163]. The minimum occurs when the aggressors switch in the same direction and simultaneously with the victim's wire, while the maximum occurs when aggressors switch simultaneously and in opposite directions.

The impact of C_v variability on jitter was evaluated with a reference inverter (size $8\times$), using the arrangement previously described as circuit A (represented in Fig. 3.13a). The input transition time (t_{in}) was kept constant and equal to the output transition time when $C_v = \mu_c$ (balanced transitions when there is no crosstalk). Then, the inverter's output capacitance (C_v) was varied in the range $[0.5\mu_c, 1.5\mu_c]$ and its propagation delay

measured. It was found that CRT induced delay increases linearly with C_v and thus, results could be fitted into a linear function ($y = m \cdot x + b$), like the one shown in Fig. 3.19c. This function represents the inverter's delay sensitivity to the variability in C_v . Thus, CRT jitter ($\sigma_{t_{d,crt}}$) can be expressed as shown in (3.9). Comparing this expression with the one presented in (2.18), and noting that μ_c is the victim's capacitance with no crosstalk, σ_c can be written as shown in (3.10). A fitting parameter (k_c) was included to reflect variable switching profiles and switching probabilities of the aggressor lines.

$$\sigma_{t_{d,crt}} = t_d \cdot (\sigma_c / \mu_c) \cdot m \tag{3.9}$$

$$\sigma_c = (k_c/m) \cdot C_{ct} \cdot \sqrt{(t_{sw}/T_{clk})/M}$$
(3.10)

Using (3.9) and (3.10), the crosstalk sensitivity factor (β_{crt}) can be written as shown in (3.11). Here, both k_c and M are considered to be associated with crosstalk sources (aggressor lines) and thus, do not represent the victim's sensitivity. Sensitivity depends essentially on the circuit's delay, switching window and interconnect layout choices (which determine the coupling capacitance).

$$\beta_{crt} = t_d \cdot \left(C_{ct} / \left(C_{gt} + C_{ct} \right) \right) \cdot \sqrt{t_{sw} / T_{clk}}$$
(3.11)

Besides the direct impact on delay, C_v variability has also an impact on the repeater's balance and thus, on jitter induced by other sources. Fig. 3.20a shows the inverter's delay and balance ratios ($r_d = t_d/t_{d,nom}$ and $r_{io} = t_{in}/t_{out}$) as a function of normalized C_v . As discussed in section 3.1.2, jitter increases linearly with fanout (and thus, with delay) and exponentially with r_{io} . Fortunatelly, r_d and r_{io} ratios change in opposite directions with crosstalk, which has a beneficial impact on jitter inserted by the line driver. Fig. 3.20b presents the simulated results of PSN, TCN and IPV jitter, as a function of normalized C_v . It can be observed that the variation in r_{io} has a noticeable impact on the jitter's linearity with the load capacitance, previously shown in Fig. 3.7. However, it also has an impact on jitter inserted by the next cell, which will experience unbalanced transitions.



Figure 3.20: Impact of CRT on: a) r_d and r_{io} ratios; b) PSN, TCN and IPV jitter.

3.3 Scalable Jitter Model

C LOCK repeaters are usually sized and spaced to guarantee sharp clock edges and maintain acceptable uncertainty levels in CDNs. However, big repeaters consume more power and generate higher PSN. Accurately predicting uncertainty in clock repeaters can thus help in preventing circuit over-design and increase its global performance. This section proposes a novel scalable jitter model for clock repeaters that can be used to estimate both static and dynamic jitter in repeaters with different sizes, interconnects and slew-rates, with low computational effort. It requires only the characterisation of a reference repeater, which can be done with a small number of simulations or measurements. This model can be used to replace time-consuming transient noise simulation when evaluating jitter in clock distribution systems, and provide a valuable insight regarding the impact of design parameters on jitter. It includes IPV and PSN, as these are the dominant static and dynamic variability sources. Crosstalk is not discussed, as it depends more on choices regarding the the routing strategy of neighbouring wires and their switching activity, than on the repeater's design.

3.3.1 Equivalent Circuit Model

Previous sections have considered only the amplification stage of clock repeaters, loaded with capacitive loads. In applications where repeaters have short interconnects between them, this is a good approximation to reality. However, when repeaters are required to drive long interconnects, the wire capacitance and resistance become relevant. Inductance can also have a significant impact in the wire impedance when fast switching signals travel in low resistance, long interconnects. However, it is disregarded in most interconnect analysis due to the high computational cost of inductance extraction and inductance-aware timing analysis [164]. For the same reason, the proposed jitter model considers only the impact of RC interconnects.

According to the FPT jitter model, if the interconnect parasitics do not significantly affect the repeater's output noise, an RC loaded repeater inserts the same amount of jitter as a capacitively loaded repeater as long as slew-rates match. This section proposes a method to obtain this equivalent load capacitance and thus, an equivalent circuit model for jitter analysis in general clock repeaters. For simplicity, a symmetric inverting Clock Repeater Cell (CRC) in a binary clock tree will be considered, as the one shown in Fig. 3.21a. R_{int} and C_{int} are the total resistance and capacitance of each wire connecting the driver to load repeaters, expressed as the product between the resistance/capacitance per unit length multiplied by the length of the wire.



Figure 3.21: CRC: a) extraction from binary clock tree; and b) its circuit model.

The circuit model for this CRC is represented in Fig. 3.21b, featuring a similar gate with half the original size driving a capacitive load (C_L) through a distributed interconnect π -model. In this model, C_{int} is equally divided in two sections², connected on either side of R_{int} . The following analysis shows how to convert the interconnect π -model into a

²If the driver and load are connected through an RC network, with multiple branches, obtaining the interconnect π -model is not so straightforward and some approximations must be considered. One possible approach is to replace the branches outside the clock path with an effective capacitance $C_{eff,bi}$. The total interconnect capacitance is then given by $C_{int} = C_{wire} + \sum C_{eff,bi}$.

single capacitive load in this symmetric CRC, but it can be easily extended to asymmetric and/or non-inverting CRCs.

The *effective capacitance* model (described in section 2.3.2) allows the designer to estimate the gate's delay ($t_{d,gate}$) in RC (and RLC) loaded repeaters. However, this capacitance is not able to capture the signal's slew-rate at the repeater's output node and thus, cannot be used to evaluate the repeater's output jitter. To do that, an *equivalent capacitance model* is here proposed. The equivalent capacitance (C_{eq}) is the one that captures the repeater's slew-rate at the output node, as shown in Fig. 3.22. It can be obtained using the same methods used to find C_{eff} and thus, it is also accurate only up to the point when the gate begins to behave like a resistor. However, for balanced repeaters (with similar input and output transition times) it can reasonably capture the cell's slew-rate during the first half of the output voltage waveform transition (coarsely between 30% and 50% of V_{dd}).



Figure 3.22: CRC π -model and its correspondent C_{eq} model.

The methodology presented in [85] was modified to enable the computation of C_{eq} . First, an analytical waveform for $v_2(t)$ was considered to compute of the mean currents through the near-end ($C_2 = C_{i2}$) and far-end ($C_1 = C_{i1} + C_L$) capacitance. This waveform should be as realistic as possible to minimise the approximation error. Thus, the shape of $v_2(t)$ was defined as a combination of quadratic and linear functions according to the typical waveform at the output of a balanced repeater. The quadratic region was defined from $t_1 (\approx t_d)$ to $t_2 (\approx t_{ig})$, while the linear region goes from t_2 to $t_3 (\approx t_d + 0.5t_{ig})$, as shown in Fig. 3.23. Here, t_{ig} corresponds to the input transition time (0% to 100% V_{dd}) necessary for the gate's output transition to be balanced. Thus, $t_{ig} \approx t_{o,gate}/0.8$.

If t_1 is considered to be the initial time instant, $v_2(t)$ can be defined as shown in (3.12). Here, V_i is the initial voltage, $t_m = 0.5t_{ig}$, $t_x = t_{ig} - t_d$ and α and β are fitting constants. For an input rising transition, the initial output voltage is V_{dd} . When $t = t_x$, the output



Figure 3.23: Key time instants for the gate's output waveform ($v_2(t)$).

voltage is $\approx 70\% V_{dd}$, falling to $\approx 50\% V_{dd}$ at $t = t_m$. Thus, fitting parameters are $\alpha \approx 0.3V_{dd}/t_x^2$ and $\beta \approx 0.2V_{dd}/(t_x(t_m - t_x))$.

$$v_{2}(t) = \begin{cases} V_{i} - \alpha t^{2}, & 0 \le t < t_{x} \\ V_{i} - \alpha t_{x}^{2} - \beta t_{x}(t - t_{x}), & t_{x} \le t \le t_{m} \end{cases}$$
(3.12)

Using (3.12), the current through the near-end capacitance $(i_2(t))$ can be easily computed, as shown in (3.13).

$$i_{2}(t) = C_{2} \cdot \frac{v_{2}(t)}{dt} = C_{2} \cdot \begin{cases} -2\alpha t, & 0 \le t < t_{x} \\ -\beta t_{x}, & t_{x} \le t \le t_{m} \end{cases}$$
(3.13)

The current through the far-end capacitance $(i_1(t))$ is not so straightforward to obtain. While $v_2(t)$ is quadratic (for $0 \le t < t_x$), the current through C_1 may be computed in the Laplace domain with $\tau = R_{int}C_1$, as shown in (3.14). However, to compute $i_1(t)$ when $v_2(t)$ is linear, it is necessary to have the voltage in C_1 when $t = t_x$. This initial voltage $(V_{c1x} = v_1(t_x))$ is computed as shown in (3.15).

$$V_{2q}(s) = R_{int}I_{1q}(s) + \frac{I_{1q}(s)}{sC_1} \implies I_{1q}(s) = \frac{V_2(s)}{R_{int} + 1/sC_1} = -2\alpha C_1 \frac{1/\tau}{s^2 (s+1/\tau)}$$
(3.14)

$$V_{c1x} = v_1(t_x) = \frac{t_x}{C_1} \int_0^{t_x} i_{1q}(t)dt + v_1(0) = V_i - \alpha \left(t_x^2 - 2\tau t_x + 2\tau^2(1 - e^{-t_x/\tau})\right)$$
(3.15)

Using this as the initial voltage in C_1 , the current $i_{1l}(t)$ for the linear portion of $v_2(t)$ (for $t_x < t \le t_m$) can be obtained. Its expression in the Laplace domain is shown in (3.16).

$$I_{1l}(s) = C_1 \frac{1/\tau}{s+1/\tau} \left[\left(V_i - \alpha t_x^2 - V_{c1i} \right) - \frac{\beta t_d}{s} \right]$$
(3.16)

Using the inverse Laplace inverse transform, the final expression for $i_1(t)$ is shown in (3.17), where $\mu_x = (e^{-t_x/\tau} - 1)$.

$$i_{1}(t) = \begin{cases} -2\alpha C_{1} \left(t + \tau \left(e^{-t/\tau} - 1 \right) \right), & 0 \le t < t_{x} \\ -\beta t_{x} C_{1} \left(1 - e^{-t/\tau} \right) - & \\ -2\alpha C_{1} \left(\tau \mu_{x} + t_{x} \right) e^{-t/\tau}, & t_{x} \le t \le t_{m} \end{cases}$$
(3.17)

The equivalent capacitance is the one that produces the same slew-rate shown by $v_1(t)$. Thus, to find C_{eq} , one must compute the time period between $v_1(t_x)$ and $v_1(t_m)$, represented by t_{swl} in (3.18). The expression for the mean current through C_1 during that period ($\overline{I_{1l}}$) is shown in (3.19), with $t_f = t_m - t_x$ and $\mu_f = (e^{-t_f/\tau} - 1)$.

$$t_{swl} = C_1 \cdot \frac{v_1(t_m) - v_1(t_x)}{\overline{I_{1l}}} = \frac{v_2(t_m) - v_2(t_x) + R_{int} \left(i_1(t_x) - i_1(t_m) \right)}{-\beta t_x \left(1 + \frac{\tau \mu_f}{t_f} \right) + 2\alpha \tau \mu_f \left(\tau \mu_x + t_x \right)}$$
(3.18)

$$\overline{I_{1l}} = C_1 \left[-\beta t_x \left(1 + \frac{\tau \mu_f}{t_f} \right) + 2\alpha \tau \mu_f \left(\tau \mu_x + t_x \right) \right]$$
(3.19)

Because these expressions depend on t_x and t_m , which are also unknowns, t_{swl} must be calculated iteratively using the same procedure used to compute C_{eff} in [85]. Once t_{swl} is obtained, C_{eq} can be obtained using empirically derived k-factor equations. They provide the gate's output transition time for a given C_L and t_{in} . Because only balanced
repeaters ($t_{in} = t_{out}$) are considered, the output switching time becomes a function of load capacitance only. Thus, C_{eq} can be easily computed from t_{swl} . Note that k-factor equations are usually available in technology library files, but can also be obtained through simple transient simulations.

Fig. 3.24 shows a comparison between the π -model waveforms and the one obtained with the C_{eq} model, for different interconnect sizes and load capacitances. Note how the model fails to accurately predict delay or capture the full CRC response, specially the waveform tail, but reasonably captures its partial output slew-rate (t_{swl}).



Figure 3.24: Waveform comparison between the CRC π -model and its equivalent model, for balanced repeaters with: a) $C_{int} = 1.4C_{in}$, $R_{int} = R_{on}$, $C_L = 2C_{in}$ and $C_{eq} = 4.3C_{in}$; and b) $C_{int} = 2.6C_{in}$, $R_{int} = 2R_{on}$, $C_L = 4C_{in}$ and $C_{eq} = 14.4C_{in}$; with $R_{on} = V_{dd}/2I_{D0}$.

To verify the accuracy of the equivalent capacitance model in jitter evaluation, jitter was measured (through simulation) in the output node of a balanced reference inverter with metal four (M4) interconnects. Results were obtained with FO2 and FO4 loads, and interconnects with different widths (W_{int}) and lengths (L_{int}). Then, the interconnect and load were replaced by C_{eq} and simulations repeated. PSN jitter was obtained with transient simulation, using independent Gaussian MMN sources, while IPV jitter was evaluated with Monte Carlo (MC) simulation. Fig. 3.25 presents the C_{eq} model jitter error



contour plots, for PSN and IPV, as a function of normalised interconnect width and length. Minimum width and length for M4 are $L_{min} = 10 \mu m$ and $W_{min} = 140 nm$.

Figure 3.25: Static and dynamic jitter error contour plots, using the C_{eq} model.

The error ($\epsilon_{eq,psn}$ and $\epsilon_{eq,ipv}$) was computed as the difference between the C_{eq} model results and the ones obtained with interconnect and load, as a percentage of the last. Results show that static jitter is over-estimated for long and thin interconnects while it is underestimated for shorter and wider lines. On the contrary, dynamic jitter is under-estimated in the direct proportion to interconnect resistance. This means that the interconnect resistance has a beneficial impact in $\sigma_{I_{p,ipv}}$ and a detrimental impact in $\sigma_{v_{o,psn}}$. Nevertheless, the error is shown to be sufficiently small for a wide range of interconnect lengths and widths, which correspond to a line resistance $R_{int} \in [0 ... 2.4R_{on}]$ and line capacitance $C_{int} \in [0 ... 3.6C_{in}]$, where C_{in} is the repeater's input capacitance.

At this point, three important observations are due. First, the only assumption behind the C_{eq} model is that the repeater's gate can be seen as a constant current source up to the threshold crossing, which is true in balanced clock repeaters. Hence, for jitter estimation purposes, an RC loaded clock repeater can be conceptually seen as gate loaded with a single equivalent capacitance. Second, if higher accuracy is desired, the proposed analytical method can be replaced by an heuristic approach. C_{eq} can be extracted through transient simulation, using the following procedure: 1) clock paths and CRCs in those paths are extracted from the clock tree; 2) simple transient simulations are performed for each CRC, using a t_{in} that guarantees balanced transitions; 3) t_{out} measured between 30% and 50% of V_{dd} (t_{swl}) is used to infer C_{eq} from empirically derived k-factor equations. Finally, it should be noted that although C_{eq} is computed under the assumption of balanced transitions, it can be used to predict jitter in unbalanced CRCs as long as they have the same design (same repeater size, interconnect and load). This will be shown in section 3.3.4.

3.3.2 Jitter Model for Symmetric Repeaters

Sections 3.1 and 3.2, have shown jitter and uncertainty results for repeaters with different designs, sizes and fanouts. Except for TCN, where noise generation strongly depends on the repeater's size and load, results have shown that uncertainty is almost constant for general repeaters with fanouts higher than two. Because small fanouts are seldom used in practical designs and TCN jitter can be neglected in most applications (where PSN is relevant), a scalable jitter model is here proposed. It is based on the assumption that design parameters affecting the CRC's timing parameters will equally affect jitter generation. Thus, the delay and output transition time characterisation of a reference repeater can be used to scale jitter insertion in CRCs with different designs.

For this purpose, two key design parameters are here defined: the cell's capacitance ratio, defined as the ratio between the equivalent capacitance and the cell's input capacitance ($r_c = C_{eq}/C_{in}$); and the cell's balance ratio, defined as the ratio between the input to output transition times ($r_{io} = t_{in}/t_{out}$). Note that C_{eq} was derived assuming that the CRC is balanced and thus, r_c and r_{io} are independent parameters. Also, the gate's size and interconnect parasitics are embedded in r_c , through C_{in} and C_{eq} , respectively.

Fig. 3.26 shows how IPV and PSN sources can be mapped to the repeater's equivalent circuit model. PSN is represented by voltage variations in power (Δv_{dd}) and ground (Δv_{ss}) rails, which include the effect of power, ground and substrate noise sources. IPV associated with the repeater is represented by Δv_{th} and ΔL , as it has the overall effect of varying the transistor's threshold voltage and channel length. IPV also affects the input capacitance of the next stage, the width, thickness and the inter-level dielectric thickness of interconnects. These effects are represented by variations in the load capacitance (ΔC_L) , interconnect resistance (ΔR_{int}) and capacitance (ΔC_{int}) , which are mapped to the equivalent model as ΔC_{eq} .



Figure 3.26: Clock repeater with jitter sources and its C_{eq} model.

The proposed model flow is schematically represented in Fig. 3.27a. The first step is to select a reference repeater and measure its delay for $r_c = r_{io} = 1$, here represented as $t_{d,ref}$. The reference repeater should be the smaller available in the library. The second step is to characterise its nominal delay (t_d) and output transition time (t_{out}), as a function of r_c and r_{io} . Although this characterisation has to be done for each technology, the model itself is technology independent. Moreover, the required characterisation data is usually already available in technology library files, which can virtually eliminate the model's computational cost. A scaling function (Γ_d) is then obtained as shown in (3.20). Fig.3.27b shows that Γ_d is a smooth function of design parameters, reflecting their impact on the CRC's timing parameters and thus, on jitter.



Figure 3.27: Scalable jitter model: a) generation flow; and b) normalised scaling function obtained for the reference inverter in a 90nm technology.

$$\Gamma_d(r_c, r_{io}) = \frac{t_d(r_c, r_{io})}{t_{d,ref}} \cdot \frac{t_{out}(r_c, r_{io})}{t_{out}(r_c, r_{io} = 1)}$$
(3.20)

This function can be used to estimate dynamic jitter ($\hat{\sigma}_{t_{d,D}}$) in any repeater, as long as a reference jitter value is available. This is shown in (3.21), where $\sigma_{t_{d,D,ref}}$ corresponds to the reference repeater's dynamic jitter. It may be obtained with transient noise simulations or the heuristic model presented in section 3.2.3. If $\sigma_{t_{d,D,ref}}$ is not available, Γ_d can still provide useful information as it is a measure of performance deterioration induced by design choices - it quantifies jitter magnification in a given repeater cell, compared to the balanced reference repeater with a fanout of one.

$$\hat{\sigma}_{t_{d,D}} = \sigma_{t_{d,D,ref}} \cdot \Gamma_d \left(r_c, r_{io} \right) \tag{3.21}$$

The scaling function defined in (3.20) can also be used to estimate the repeater's static jitter ($\hat{\sigma}_{t_{d,Sr}}$), as shown in (3.22). Again, a reference jitter value ($\sigma_{t_{d,S,ref}}$) is required, which can be obtained with MC simulations.

$$\hat{\sigma}_{t_{d,Sr}} = \sigma_{t_{d,S,ref}} \cdot \Gamma_d \left(r_c, r_{io} \right) \tag{3.22}$$

However, to predict the overall CRC static jitter, it is necessary to further consider the impact of interconnect and load variability. To do that, it was necessary to obtain ΔC_{eq} using MC simulations. The variability associated with each metal layer of interest was characterised using a simple circuit that comprises a ramp voltage source with constant slope (dv/dt = m), a metal interconnect and a reference repeater as load. This circuit is shown in Fig. 3.28. To avoid leaving the repeater's output node open, a single capacitance to ground (C_{out}) was used at the inverter's output node, which was set to C_{eq} in each experiment. Simulations were repeated for several interconnect lengths and widths, using the following procedure: 1) start with a minimum size interconnect and obtain the average current (I_{avg}) through this line; 2) use this to estimate C_{eq} as I_{avg}/m ; 3) obtain the current standard deviation $\sigma_{I_{avg}}$ with MC simulation and compute $\sigma_{Ceq} = \sigma_{I_{avg}}/m$; 4) finally, select a different load and interconnect dimensions (different ratio C_{int}/C_{eq}) and repeat the procedure. The stop condition depends on the desired accuracy, but the func-



tion is smooth enough to require only a few points.

Figure 3.28: Simulation framework to characterise C_{eq} variability.

Using this procedure, $\sigma_{C_{eq}}/C_{eq}$ was obtained as a function of W_{int}/W_{min} and C_{int}/C_{eq} . The interconnect length is not a direct parameter (is embedded in C_{int}/C_{eq}) because it has an equal impact on both R_{int} and C_{int} . Fig. 3.29 shows the results for two different interconnect layers - metal four (M4) and top metal layer (M2.2B), in a IBM's 90nm technology. The width is normalised to the minimum value allowed in each layer (W_{min}), and neighbouring wires were considered to be quiet (no crosstalk). In these plots, variability is shown to increase with higher (C_{int}/C_{eq}). Also, it increases with a faster rate in thin interconnects, which means that IPV variability is directly proportional to line resistance.



Figure 3.29: Variability in metal four (M4) and top metal layer (M2_2B).

The function $\sigma_{C_{eq}}/C_{eq}$ was then used to estimate static jitter induced by the interconnect and load variability ($\hat{\sigma}_{t_{d,Si}}$), using the Elmore gate delay as shown in (3.23).

$$\hat{\sigma}_{t_{d,Si}} \approx 0.67R_{on} \cdot \sigma_{C_{eq}} \text{ with } \sigma_{C_{eq}} = C_{eq} \cdot f\left(C_{int}/C_{eq}; W_{int}/W_{min}\right)$$
(3.23)

As partial static jitter components in (3.22) and (3.23) are not correlated, their variances can be added to estimate the total CRC static jitter (3.24). However, because the proposed C_{eq} over-estimates static jitter when the interconnect resistance is high (which is exactly when it is more affected by IPV variability), IPV jitter was found to be well estimated with the repeater's contribution only ($\hat{\sigma}_{t_{d,Sr}}$). Nevertheless, considering both contributions can be useful, as it provides a worst case jitter prediction.

$$\hat{\sigma}_{t_{d,S}} = \sqrt{(\hat{\sigma}_{t_{d,Sr}})^2 + (\hat{\sigma}_{t_{d,Si}})^2}$$
(3.24)

3.3.3 Jitter Model for Asymmetric Repeaters

Clock repeaters are usually designed to have symmetric transitions, with the exception of SEC inverters and asymmetric TDRs. As explained in section 3.1.1, SEC inverters are designed to favour the propagation of the critical clock edge. When they are cascaded, the critical and the neglected clock edges see virtually different balanced repeaters. The critical clock edge sees a fast repeater because the load is smaller than what would be expected in a symmetrical repeater with that transistor's size. Likewise, the neglected clock edge sees a slow repeater because its load is bigger than expected. In (3.25) the relation between the size of PMOS and NMOS transistors in an asymmetric inverter, compared to the correspondent symmetric inverter, is presented. Here, $\beta = W_p/W_n$ is the ratio between PMOS and NMOS transistor's width.

$$W_{n,sec} = \frac{1+\beta}{1+\beta_{sec}} \cdot W_n \; ; \; W_{p,sec} = \frac{(1+\beta)\beta_{sec}}{(1+\beta_{sec})\beta} \cdot W_p \tag{3.25}$$

The SEC inverter can thus be decomposed in two virtual inverters: one seen by the critical clock edge (fast inverter) and another seen by the neglected edge (slow inverter). Because these are balanced virtual inverters, the characterisation data obtained for the reference symmetric repeater can also be used to estimate their jitter, as long as equivalent fanouts are defined for them. To do that, it is necessary to separately consider the rising

and falling transition times in a SEC inverter and in its correspondent symmetric inverter. If $t_{d,LH}$ and $t_{d,HL}$ are the symmetric inverter rising and falling transition times for h = 1and R_p and R_n are the transistor's channel resistances, the SEC inverter's transition times can be expressed as shown in (3.26) and (3.27), using the Elmore delay approximation. Note that $C_{L,sec} = C_L$, by definition, and $R_{p,sec}$ and $R_{n,sec}$ correspond to the Thevenin equivalent resistances of the SEC inverter's transistors.

$$t_{d,LH,sec} = 0.69 \cdot C_{L,sec} \cdot R_{p,sec} = 0.69 \cdot C_L \cdot R_p \cdot \frac{W_p}{W_{p,sec}} \cdot h = t_{d,LH} \cdot \frac{\beta \left(1 + \beta_{sec}\right)}{\beta_{sec} \left(1 + \beta\right)} \cdot h \quad (3.26)$$

$$t_{d,HL,sec} = 0.69 \cdot C_{L,sec} \cdot R_{n,sec} = 0.69 \cdot C_L \cdot R_n \cdot \frac{W_n}{W_{n,sec}} \cdot h = t_{d,HL} \cdot \frac{1 + \beta_{sec}}{1 + \beta} \cdot h$$
(3.27)

Factors affecting the symmetric inverter's transition times can also be seen as factors affecting the virtual inverter's load. For example, the output rising edge in a FO1 strong pull-up inverter (Invr) sees a smaller load capacitance than it would expect, considering the size of its PMOS transistor. Thus, it corresponds to a virtual fanout smaller than one $(h_{fast} < h)$. On the contrary, the falling edge sees a bigger capacitance than it would expect, which corresponds to $h_{slow} > 1$. Thus, the equivalent fanout for the fast and slow inverters in this SEC inverter (Invr) can be expressed as shown in (3.28). Similar expressions can also be easily derived for the strong pull-down inverter (Invf).

$$h_{slow} = \frac{1 + \beta_{sec}}{1 + \beta} \cdot h \wedge h_{fast} = \frac{\beta \left(1 + \beta_{sec}\right)}{\beta_{sec} \left(1 + \beta\right)} \cdot h$$
(3.28)

The proposed jitter model can also be easily extended to asymmetric TDRs. They can be associated to as many virtual inverters as the possible delay increments, each of which with its own jitter model. Alternatively, the model can be applied only to the virtual inverter with the higher propagation delay (worst case jitter), the lower introduced delay (worst case current consumption), or a combination of both.

3.3.4 Model Evaluation

The proposed scalable jitter model assumes that there is a design space where the CRC's timing parameters and jitter are equally affected by design parameters. This section investigates the boundaries within which this assumption is reasonable, for an inverterbased CRC implemented in a 90nm technology. Monte Carlo (MC) and transient noise simulation were used to obtain static and dynamic jitter, using the simulation framework described in Sec. 3.1.2. Transient simulations were also performed to obtain Γ_d (r_c , r_{io}), according to the methodology presented in section 3.3.2.

Jitter simulations were performed using a balanced inverter repeater with 10 times the size of the reference repeater, and interconnects routed in M4 metal layer. The interconnect length and width was varied, to evaluate the impact of different R_{int} and C_{int} parameters. For load, both 2 and 4 similar repeaters were used in parallel. Model predictions were then compared to simulation results and the percent error computed. Static and dynamic jitter error rates are shown in Fig. 3.30, for capacitance ratios $r_c \in [1, 12]$ and resistance ratios $r_r = R_{int}/R_{on} \in [0, 2.5]$. The error is shown to be quite scattered inside this significantly broad design space, but always falling within 6% of simulation results. These results show the model accuracy and applicability for most clock repeaters, which are usually designed to be balanced.



Figure 3.30: Jitter error in balanced repeaters as a function of: a) r_r ; and b) and r_c .

Moreover, the model can also be applied to unbalanced repeaters, as shown in Fig.

3.31 for $r_{io} \in [0.4, 1.8]$. Simulations were performed for the same load and interconnect sizing, described for balanced repeaters. According to these plots, jitter predictions tend to be more accurate when $r_{io} > 1$. Nevertheless, most jitter predictions are within 10% of simulation results, for a significantly wide design space. Thus, results show that the proposed jitter model is applicable and accurate for balanced CRCs, but can also be used to predict jitter in unbalanced CRCs with reasonable accuracy.



Figure 3.31: Model error in repeaters with different designs, as a function of r_{i0} : a) static jitter; and b) dynamic jitter.

3.4 Conclusions

T HIS chapter discussed clock precision in clock repeaters. Section 3.1 described different architectures of Static Delay Repeaters (SDRs) and Tunable Delay Repeaters (TDRs), compared their performance and evaluated the jitter behaviour of their basic building block - the CMOS inverter. Section 3.2 proposed a jitter model for this inverter, based on sensitivity metrics, considering the most relevant intrinsic and environmental variability sources. Finally, section 3.3 proposed a scalable model to estimate jitter in general clock repeaters with RC interconnects. The main conclusions drawn in each of these sections are summarised next.

Results in section 3.1.2 lead to the following observations. First, jitter grows linearly with the load capacitance, temperature, input transition time and noise levels (at least for

 $\sigma_{psn} < 0.1 V_{dd}$), while it grows exponentially with $t_{in} < t_{out}$ and with $v_n > 10\%$. Second, IPV and PSN jitter have the same order of magnitude for common PSN levels, while TCN jitter is two orders of magnitude smaller. This means that dynamic jitter results essentially from noise in the power supply rails. Finally, the spectral content of PSN sources has no influence on jitter as long as these sources can be considered to be low-frequency $(f_n < 0.25 f_{clk})$. This is the most common situation in digital ICs.

Section 3.1.3 compared the precision of common SDRs and TDRs. Jitter and uncertainty were shown to increase with fanout, except for TCN and IPV uncertainty, which decreased with fanout. This means that when repeaters are used to introduce delay in systems with low PSN, it is best to use a small number of heavily loaded inverters than a large number of lightly loaded ones. Comparing different structures, it has been shown that SEC inverters and tapered buffers are the SDRs with higher precision, while SCIs are the best among TDRs. Furthermore, results have shown that both static and dynamic uncertainty is almost constant in clock repeaters. This means that for a given delay insertion, precision is determined essentially by the implementation technology.

Section 3.2 proposed a model to estimate jitter in CMOS inverters and identify the key circuit parameters on which it depends. The inverter was considered to be driven by an ideal clock source and loaded with a single capacitance to ground (C_L). As long as the clock source guarantees balanced transitions and C_L induces the same delay as the one shown by the inverter with its original load, these simplifying assumptions have been proved reasonable. Jitter sensitivity has been shown to depend on the peak current for intrinsic variability sources (TCN and IPV), while it depends on the effective current for PSN sources. The main advantage of this approach is that expressions depend only on parameters that can easily be obtained from early circuit simulation or from data usually disclosed by technology providers. Thus, they can provide information on upcoming constraints very early in the design stage.

Finally, section 3.3 presented a scalable model to estimate jitter in general clock repeaters with RC interconnects. It is based on the observation made in section 3.1.3 regarding the low variability of uncertainty. The proposed model allows the designer to optimise the repeater's size and spacing, for a given jitter budget, with low computational effort. This avoids the use of unnecessary big inverters which consume more power and generate higher PSN. On the other hand, it provides a valuable insight regarding the repeater's key design parameters in jitter insertion, including the gate, load and interconnects. Results show that the proposed model predicts jitter with an error within 10% of simulation results, for a significantly wide design space.

Models proposed in section 3.2 and 3.3 will also be useful in chapter 6, to investigate uncertainty trends with technology scaling.

Chapter 4

Uncertainty in Clocking Structures

This chapter investigates how uncertainty propagates and accumulates in clocking structures. These structures can be used to introduce controllable amounts of delay in the clock path or to distribute a clock signal from one source to multiple sinks. It starts evaluating static and dynamic jitter in cascaded clock repeaters. Conclusions taken from simulation results are then used to support an heuristic dynamic jitter accumulation model, which can be used to predict jitter accumulation bounds. This model takes into consideration the impact of power and ground noise correlations, as well as correlations between PSN sources affecting neighbouring repeaters. Finally, a model to analyse uncertainty in structures with feedback is proposed.

4.1 Delay Lines and Clock Trees

CLOCK repeaters are typically used to distribute a clock signal to different locations or to introduce controllable amounts of delay in a given clock path. The former is usually called a Clock Distribution Network (CDN), while the last is generally referred to as a Delay Line (DL). Apart from their purpose, the most significant difference between these structures is the electrical and physical distance between repeater cells. In DLs, repeaters are usually located in close proximity, so interconnect parameters may be disregarded when analysing the circuit's operation. On the contrary, interconnect parasitics cannot be neglected in CDNs. This section describes the most common architectures of DLs and CDNs, and discusses the key parameters that influence their precision.

4.1.1 Digitally Controlled Delay Lines

When a clock signal travels through more than one repeater it is considered to travel through a DL, which in its simplest form is just a cascade of SDRs. If the line is composed of inverters with increasingly larger size, it is called a tapered DL. This is a very common structure when a small gate has to drive a large capacitive load. DLs are also frequently used to intentionally delay the clock signal, in order to meet some timing specifications. If the delay is digitally adjusted the line is called a Digitally Controlled Delay Line (DCDL) [165]. Analog Controlled Delay Lines (ACDLs) provide finner delay steps, but typically have higher noise sensitivity, smaller tuning range and more complex control circuitry. This thesis discusses only DCDL, for their popularity in current high-performance ICs.

If each stage introduces a constant delay, the delay line is said to be uniform and the control unit is typically a ring counter. This is usually the case when DCDLs are implemented with SDRs in a single-path configuration, as shown in Fig. 4.1. The inverter is usually chosen for its reduced insertion delay [147]. However, the minimum delay is also determined by the multiplexer used to control the number of stages through which the signal travels. On the contrary, NAND delay lines do not require a multiplexer as they have the benefit to provide two point-of-entry control signals. As a consequence, they can be used to build DCDLs with very regular layouts [166,167]. Non-inverting SDRs, like buffers or AND gates, are less popular because DCDLs are usually required to have the lowest unit delay possible.



Figure 4.1: Uniform DCDLs, built with: a) inverter gates; b) NAND gates.

When a large phase range and a reduced implementation area are required, delay stages can be implemented with weighted delays, controlled with a binary counter. This is typically accomplished with SCIs [168] or SDRs associated to multiplexers in a parallel-

path configuration [148]. These structures are shown in Fig. 4.2. With SCIs, the unit delay depends on the cell's minimum load capacitance and a single stage is enough to implement the line. However, more than one may be used to increase the total line delay, as shown in Fig. 4.2a. On the contrary, parallel-path DCDLs (Fig. 4.2b), require the use of multiple stages and the unit delay is obtained from the difference between the delay of alternative paths within the minimum delay stage. Due to their poor resolution, these lines are typically used to implement coarse DLs while single stage SCIs are commonly used to implement fine-tuning DLs.



Figure 4.2: Binary weighted DCDLs with: a) SCIs; b) SDRs in a parallel-path configuration.

The number of delay elements in a inverter-based uniform DCDL depends on the required total delay (t_D) and may be calculated using the Sakurai's propagation delay and transition time expressions, referred in section 2.3.1. Considering $t_{d,LH} = t_{d,HL} = t_d$ and $t_{in} = t_{out}$, the total delay t_D after N cells is shown in (4.1). One can see that t_D is proportional to the gate's output capacitance, which results from the next cell's input capacitance and the multiplexer's input capacitance. For large delays, one can either increase the number of cells or the load capacitance in each cell. However, to have a good delay resolution and a large dynamic range, a large number of fast delay cells is required.

$$t_D = N \cdot t_d = N \cdot \frac{C_L V_{dd}}{2I_{d0}} \cdot \left(2 \cdot \left(\frac{0.9}{0.8} + \frac{V_{D0i}}{0.8V_{dd}} + \ln\frac{10V_{D0i}}{eV_{dd}}\right) \cdot \left(\frac{1}{2} - \frac{1 - v_T}{1 + \alpha}\right) + 1\right)$$
(4.1)

Alternatively, a single stage SCI with a shunt capacitance M = N - 1 times bigger than C_L can be used, saving area and power consumption. However, this large capacitance may compromise slew-rate and increase the signal sensitivity to PSN. To take the best out of both worlds, DCDLs often employ multiple stages of coarse and fine-tuning delay cells with different structures. Coarse tuning is usually accomplished with uniform or parallel-path DLs, for their simplicity and predictability, while fine tuning is implemented with TDRs [167–169]. Fine tuning delays can also be implemented with direct [170, 171] and feedback path phase blenders [172] or variable strength drivers [144]. Yet, these particular structures are not discussed here because they are not built with cascaded repeaters.

Regarding precision, uniform DCDLs are usually seen as less accurate than SCIs, because jitter accumulates along the line. However, several different phenomena should be accounted for when analysing jitter in these structures. Although jitter accumulation may be worse in long uniform DLs (due to the large number of cascaded cells), SCIs are intrinsically unbalanced and have long transition times when a large delay is required. Thus, jitter insertion in each stage is expected to be higher than in uniform DCDLs, which have fast balanced cells (as discussed in Chapter 3). These effects will be further discussed in section 4.1.3.

4.1.2 Clock Distribution Trees

Clock trees have the complex task of equalising the delays from the clock source to clock sinks. They are usually two dimensional structures intended to connect a clock source to clocked units, scattered throughout a synchronous system. Because the clock signal must travel long distances in interconnects with significant Resistor and Capacitor (RC) delays, clock repeaters are usually necessary to regenerate the signal. These repeaters should also guarantee matched latencies from source to sinks, in order to minimise skew. This can be done by matching wire lengths (e.g., with symmetric trees); matching electrical path lengths (e.g., with balanced trees); or both. The buffered H-tree is one good example of both techniques.

The design of buffered H-trees is a very simple task. The selected number of stages determines the number for clock repeaters from source to sink, the total load driven by each sink (considering a uniform load distribution) and the wire length in each tree branch. The interconnect width depends on the selected wire sizing technique, which can be uniform (equal widths in all stages) or geometric (the width is geometrically increased from sink to source), although geometric sizing is the most common in practice. Fig. 4.3a represents an H-tree with three stages (i.e., with N=7 repeaters along the clock path) and uniform wire sizing, while Fig. 4.3b represents a two stage H-tree (N=5) with geometric wire sizing. Note that an odd number of repeaters is here assumed because the final repeater is considered part of the clock tree.



Figure 4.3: H-tree topology with: a) three stages and uniform wire sizing; b) two stages and geometric wire sizing.

For a given branch *i*, the interconnect capacitance (C_{int}) and resistance (R_{int}) can be approximately computed as shown in (4.2). Here, *R* is the wire resistance, C_c is the coupling capacitance and C_g is the capacitance to ground in a nominal width interconnect, *w* is the wire width ratio to the nominal width and L_w is the wire length. Once the interconnect parasitics are determined, the optimal repeater size for minimum skew can be easily obtained [173]. However, trade-offs between power dissipation and skew often impose limits on repeater sizing [174].

$$C_{int,i} = \left(2C_c + C_g \cdot w_i\right) \cdot L_{w_i} ; \quad R_{int,i} = R \cdot L_{w_i} / w_i$$

$$(4.2)$$

For generic clock trees and non-uniform load distribution, tree design is not so straightforward and Clock Tree Synthesis (CTS) tools are needed to meet specifications. The first step is topology generation through partitioning the clock sinks, followed by routing and optimisation steps. The primary job of traditional CTS tools is to vary routing paths and the placement of the clocked cells and clock buffers, to meet maximum skew specifications with minimum area and power consumption [7]. The designer should also be able to trade-off power and area for timing, if higher operation speed is required. Traditional clock tree performance metrics are:

- *power consumption* clock trees consume a significant portion of the total chip power since they have the highest activity factor and drive the largest capacitive load in synchronous systems. It is thus important to minimise its size, not only for power saving concerns but also for temperature and PSN;
- *implementation area* implementation area is not always available for free in a given location. Increasing the repeater's size may be costly (if circuit blocks must be replaced and/or rerouted) or even not possible;
- routing resources the clock net is one of the largest nets in a synchronous systems and is usually one of the first nets to be routed. Its routing area should be minimised because it usually constitutes a blockage for other nets;
- *insertion delay* uncertainty is roughly proportional to path delay, so the tree insertion delay should be minimised;
- *clock skew* since skew represents a cycle-time penalty, it is important to minimise it in order to enable maximum operating frequency;

Today, designing CDNs for high-speed systems is more complex than just meeting skew specifications. Supply voltages have dropped while chip power consumption has remained constant or even increased, causing chip currents to increase. As a consequence, the required impedance to maintain a fixed percentage noise budget on the power supply, and contain clock jitter, became extremely challenging to achieve [175]. On the other hand, modern packaging styles and techniques to reduce power consumption have changed the typical PSN profiles [176], which further contributes to reduce the designers confidence regarding the system's expected dynamic jitter performance.

Several works have shown that clock trees with a small number of large repeaters with wide interconnects between them are more robust to variability sources [63, 105]. This also guarantees sharp transitions, which are critical for high-speed operation and robustness to PVT variations [49]. Authors in [177] have also shown that parameter vari-

Table 4.1. DCDL performance metrics with $v_{psn} = 6 \sqrt{s} v_{dd}$.											
Time [ps]					Jitter [ps]			Unc	Uncertainty [%]		
DCDL	d	$t_{D_{max}}$	$t_{D_{min}}$	Delay	PSN	TCN	IPV	PSN	TCN	IPV	
Inverter +	15	289	92	min	5.59	0.09	6.45	6.05	0.10	6.98	
Multiplexer ⁽¹⁾	15			max	26.4	0.34	19.8	9.13	0.12	6.86	
Inverter +	25	288	114	min	9.10	0.05	5.81	8.00	0.04	5.10	
Multiplexer ⁽²⁾	23			max	26.8	0.15	19.2	9.33	0.05	6.67	
NAND	69	276	69	min	4.67	0.08	3.90	6.90	0.12	5.76	
				max	29.6	0.18	18.6	9.75	0.07	6.76	
SCI	Q 1	297	175	min	15.6	0.13	12.2	9.92	0.07	6.99	
(Type 2)	0.1			max	28.1	0.19	14.2	9.50	0.06	4.80	
Parallel Path +	23	585	247	min	22.8	0.13	17.0	8.99	0.05	6.68	
Multiplexer ⁽²⁾				max	52.0	0.23	36.4	8.64	0.04	6.05	

Table 4.1: DCDL performance metrics with $\sigma_{vsn} = 6\% V_{dd}$

⁽¹⁾: with Pass-Transistor Gate multiplexer

⁽²⁾: with Three-State Inverter multiplexer

ation effects on the final levels of an H-tree have a higher impact on performance than those closer to the source [177]. However, none of these works provide insights regarding the impact of design parameters and noise correlation on jitter.

4.1.3 Performance Analysis

This section evaluates the performance of DCDLs and CDNs, using the simulation framework and gate design techniques described in section 3.1. The performance of clock averaging structures, as clock grids, meshes or spines, will not be discussed as it would be out of the scope of this thesis. The interested reader is referred to [117].

Table 4.1 presents time and precision metrics for uniform and binary weighted DCDLs. These circuits were implemented in a 90nm IBM technology, and designed for the same maximum path delay ($t_{D_{max}}$). The only exception is the parallel-path DCDL, designed for twice the delay of other lines, for practical reasons. Timing metrics include the delay resolution (*d*) and the maximum and minimum path delay ($t_{D_{max}}$ and $t_{D_{min}}$). Clock precision is evaluated with TCN, IPV and PSN induced jitter and uncertainty. These metrics were obtained for maximum and minimum delay and with $\sigma_{psn} = 6\% V_{dd}$, $T_n = 4T_{clk}$ and T = 27 °C. In PSN jitter evaluation, the same MMN sources were used for all cells, so noise is totally correlated between stages but independent in power and ground rails.

Two different inverter-based DCDL implementations are compared: one using a Pass-Transistor Gates (PTGs) multiplexer; and another built with a Three-State Inverters (3STIs) multiplexer. Because the multiplexers have different input capacitance, those DCDLs have different resolutions and number of cells for the same $t_{D_{max}}$. With a PTG multiplexer, the DCDL includes fourteen inverters while it requires only eight inverters when using a 3STI multiplexer. With less stages and a heavier multiplexer, the 3STI DCDL has a smaller dynamic range ($t_{D_{max}} - t_{D_{min}}$), larger delay resolution (*d*) and higher PSN uncertainty. On the contrary, it has a smaller intrinsic uncertainty (induced by TCN and IPV) due to higher cell's fanout.

Using NAND gates designed for the same output current as inverters, the DCDL requires only four stages for the same $t_{D_{max}}$. Compared to other uniform lines, it has a larger resolution (*d*) and dynamic range, because there is no multiplexer adding a delay overhead. Measured values for jitter and uncertainty are very similar to the ones obtained for inverter-based lines, although slightly higher for PSN. This was already expected, as NAND gates have a larger parasitic delay and higher logical effort, which conducts to a larger load capacitance.

Results for a single stage SCI and a parallel-path DCDL are also shown. The parallelpath DCDL has four binary weighted stages with 3STI multiplexers, while the SCI is controlled by four binary weighted capacitors in a type 2 configuration (according to section 3.1.1). The SCI has a smaller delay step but very limited dynamic range. On the contrary, the parallel-path line has about twice the dynamic range, with a delay resolution three times larger. Due to its larger $t_{D_{max}}$, absolute PSN and IPV jitter are also much larger than in other DCDLs, although uncertainty remains similar to other DLs.

According to these results, uncertainty seems to be a quite stable parameter, regardless the selected delay or DCDL architecture. This can be better observed in Fig. 4.4, where jitter and uncertainty evolution are graphically represented. Results are presented for increasing delays, from $t_{D_{min}}$ up to $t_{D_{max}}$. Uncertainty is shown to have a much smaller variability among DCDLs, than absolute jitter. This can be further noticed in Table 4.2. It shows the standard deviation of jitter (σ_I) and uncertainty (σ_U), measured within each line's dynamic range, as a percentage of values obtained for $t_{D_{min}}$. Compared to jitter,

			Stati	Statistics				
Metric		Inv+Mux ⁽¹⁾	Inv+Mux ⁽²⁾	NAND	SCI2	Pp+Mux ⁽²⁾	μ	σ
σ_J / J_{min}	PSN	71.4%	67.7%	230.1%	11.1%	29.1%	81.9%	86.7%
	TCN	66.7%	25.0%	53.9%	9.3%	16.1%	34.2%	24.8%
	IPV	24.6%	12.7%	34.1%	2.9%	9.1%	16.7%	12.5%
σ_U/U_{min}	PSN	7.1%	12.8%	19.2%	1.7%	3.2%	8.8%	7.2%
	TCN	9.1%	7.9%	19.8%	1.4%	4.8%	8.6%	6.9%
	IPV	3.4%	3.1%	4.3%	2.1%	2.9%	3.2%	0.8%

Table 4.2: DCDL jitter and uncertainty variability within the dynamic range.

⁽¹⁾: with Pass-Transistor Gate multiplexer

⁽²⁾: with Three-State Inverter multiplexer

uncertainty is here shown to be a quite stable parameter. Note that this observation has also been made in section 3.1, regarding uncertainty in clock repeaters.



➡Inv+3STImux ✦Inv+PTGmux ₹NAND ▲SCI2 やPP+3STImux

Figure 4.4: Jitter and uncertainty in DCDLs, for increasing delays.

The impact of different PSN magnitudes and bandwidths has also been evaluated using simulations. Fig. 4.5, shows PSN jitter evolution in an 8-cell inverter-based DCDL. The maximum line delay was selected and the same noise sources were applied to all cells. Above each plot, the mean uncertainty along the line (μ) is shown, as well as its standard



deviation (σ). One can see that PSN uncertainty is almost constant along the line, depends almost linearly on the PSN level (v_n) and is not very sensitive to T_n as long as $T_n > T_{clk}$.

Figure 4.5: Jitter and uncertainty along an inverter-based DCDL (with a 3STI multiplexer) for increasing: a) PSN level (v_n); and b) noise step (T_n).

Based on previous observations, an expression for the normalised low-frequency PSN uncertainty (Y_{psn}) is proposed in (4.3). This is expected to be a constant parameter for DCDLs, regardless their architecture. It should depend only on the implementation technology, as will be discussed in Chapter 6.

$$Y_{psn} = U_{psn} / v_n = \left(\sigma_{t_{d,psn}} \cdot V_{dd}\right) / \left(t_d \cdot \sigma_{psn}\right)$$
(4.3)

The accuracy of the traditional statistical accumulation model, presented in section 2.3.2, is also here investigated. Due to the cell's proximity in DCDLs, all elements are considered to be affected by totally correlated PSN and IPV sources. Thus, individual jitter contributions should add as standard deviations, according to that model. On the contrary, TCN jitter contributions are intrinsically uncorrelated and thus, individual contributions should add as variances. Fig. 4.6 compares jitter simulation results with the statistical model predictions, for the inverter-based DCDL with 3STI multiplexer. Individual contributions, here used to compute model predictions, were obtained from simulations. Results show that the statistical model provides reasonably accurate predictions for TCN, IPV and MMN jitter. However, it fails significantly for DMN or CMN sources.



Figure 4.6: Jitter simulation results and statistical model predictions for: c) uncorrelated TCN sources; b) totally correlated PSN sources; and c) totally correlated IPV sources.

The accuracy of the traditional statistical model when predicting PSN jitter in an Htree with five cascaded CRCs is shown in Fig. 4.7. Plots show the error computed as the difference between statistical model predictions and simulation results, as a percentage of the last. One can see that it fails to predict dynamic jitter accumulation in most situations. When noise sources are uncorrelated, the model (sum of variances) follows the same trends shown by simulation results, but underestimates jitter with an error around 40% after only five CRCs. When noise sources are totally correlated, it provides even worse predictions. For CMN the error is large and non-monotonic because the positive effect of CMN on delay is disregarded. The next section proposes a modified statistical accumulation model, to predict PSN jitter accumulation in DCDLs, clock trees and other distribution structures, with much higher accuracy.

4.2 Jitter Accumulation Model

THIS section analyses the mechanism behind PSN jitter accumulation in Clock Repeater Cells (CRCs) and discusses the impact of jitter amplification, according to power/ground noise correlations and correlations among noise sources in cascaded repeaters. TCN and IPV jitter accumulation is not discussed because the traditional statistical accumulation model is reasonably accurate for these jitter sources.



Figure 4.7: Error between traditional statistical accumulation jitter predictions and simulation results, for: a) uncorrelated noise sources; and b) correlated noise sources.

4.2.1 Dynamic Jitter in Cascaded Repeaters

In cascaded clock repeaters, jitter associated to a clock path depends on jitter inserted by each CRC on that path. CRCs are usually designed to to have $t_{in} = t_{out}$, so a given clock path can be represented by a cascade of balanced CRCs (with similar r_c and $r_{io} = 1$). Fig. 4.8 represents such a general clock path with N equivalent CRCs affected by PSN sources. Here, $\sigma_{t_{d,i}}$ is the total jitter observed at the output of cell i, which is different from the jitter inserted by cell i ($\sigma_{t_{d,i}}$), due to jitter amplification and accumulation.



Figure 4.8: Cascaded CRCs and their output jitter.

To explain the physical mechanism behind jitter amplification, the results of a simple experiment with three cascaded CRCs are presented next. The supply and ground levels on the first cell were varied and the waveforms along that line (v_{ni}) were compared with the waveforms of a reference line (v_{ri}), with nominal supply and ground levels. Fig.

4.9 shows the waveforms for CMN and DMN, where Δt_{di} is the instantaneous delay error observed at the output of cell *i*. Similar results could be obtained for $\Delta v_{dd} > 0$. Note that absolute jitter is by definition the standard deviation of Δt_{di} .



Figure 4.9: Waveforms of a reference CRC line (v_{ri}) and a CRC line affected with PSN in the first cell only (v_{ni}) for: a) CMN; and b) DMN.

The graphics show that for both CMN and DMN, the instantaneous delay error introduced by the first cell is transfered to the second cell with gain. This gain results from the fact that the second cell's input voltage is different from its supply voltage, which affects its response to the input transition. After the second cell, the delay error does not increase because there is no further influence of PSN sources (applied to the first cell only). Thus, the gain for uncorrelated PSN sources (g^{μ}) depends on the relative position between the noisy CRC and the observed cell. It has also been shown to be higher for DMN than for CMN and depend on the CRC design parameters, as will be discussed latter in this section.

A second experiment was performed to observe jitter gain when CRCs have totally correlated PSN sources. The same repeater line was used, but now all cells shared the same power and ground levels. The resulting waveforms are shown in Fig. 4.10 for $\Delta v_{dd} < 0$, but similar results could be obtained for $\Delta v_{dd} > 0$. It can be observed that the

instantaneous delay error measured at the second cell is not twice the error measured in the first cell, as would be expected in a cascade of identical cells with correlated noise sources. For CMN, the amplification gain is even negative (attenuation), which almost mitigates jitter accumulation. This negative effect of CMN in cascaded inverters has also been observed in [51]. On the contrary, DMN causes a significant jitter amplification as all contributions have a positive effect on jitter accumulation. The gain for correlated sources (g^c) is thus different from g^u , although it also depends on the noise mode and CRC design parameters.



Figure 4.10: Waveforms of a reference CRC line (v_{ri}) and a CRC line with the same PSN sources applied to all cells (v_{ni}) for: a) CMN; and b) DMN.

4.2.2 Bounds for Jitter Accumulation

If PSN sources are uncorrelated in each CRC, the superposition principle can be used to estimate jitter along the repeater line, as shown in (4.4). To account for the amplification effect identified in the previous section, an amplification gain parameter is used. For uncorrelated sources, the gain elements define a lower triangular matrix $[g^u]$ with $g_{ij}^u = 0$, for j > i and $g_{ij}^u = 1$, for j = i. Each element g_{ij}^u is the gain applicable to jitter

inserted by cell *j* in order to obtain its contribution to jitter in cell *i*. Jitter variances are added because jitter contributions from uncorrelated sources are independent random variables (the superscript '*u*' stands for uncorrelated). These individual contributions ($\sigma_{t_{d,ii}}$) can be obtained using the scalable jitter model proposed in Chapter 3 or transient noise simulation results.

$$[\sigma_{t_{d,i}}^{2^{u}}]_{N\times 1} = [g^{2^{u}}]_{N\times N} \cdot [\sigma_{t_{d,ii}}^{2^{u}}]_{N\times 1}$$
(4.4)

If PSN sources are totally correlated, the superposition principle cannot be used. In this case, the dynamic jitter after N cells depends on the sum of individual standard deviations and on the gain for totally correlated sources. To compute dynamic jitter at the output of cell k, in a cascade of N cells, the expression in (4.5) should be used. Here, the superscript 'c' indicates that PSN sources are correlated.

$$[\sigma_{t_{d,i}}{}^c]_{N\times 1} = [g^c]_{N\times 1} \cdot \sum_{i=1}^k \sigma_{t_{d,ii}}{}^c$$
(4.5)

In a general clock path, the amplification gain depends on many different design and noise parameters, and is associated with the repeater's non-linear behaviour during the signal transition. Thus, it is not straightforward to derive an accurate analytical model for $[g^u]$ and $[g^c]$. Instead, an heuristic method based on the characterisation of a reference repeater line with six similar CRCs is proposed. Screening experiments revealed that this is a sufficient number of cells to provide an accurate characterisation.

For uncorrelated PSN sources, the amplification gain can be obtained with transient noise simulations with PSN applied to the first cell only. This can be done because g^u depends only on the relative position between the noisy CRC and the observed cell. Several transient noise simulations were repeated for different noise modes, noise variances and CRC design parameters (r_c and r_r). Results were then used to obtain the gain elements (g_{ij}^u), computed as the ratio between jitter measured at the output of cell *i* and jitter generated in the first cell (j = 1), as shown in (4.6).

$$g_{ij}{}^{u} = \sigma_{t_{d,i}}{}^{u} / \sigma_{t_{d,jj}}{}^{u} , \ i = 1, ...N , \ j = 1$$
(4.6)

For correlated PSN sources, the amplification gain elements (g_i^c) can be obtained with a similar procedure, but with the same PSN sources applied to all CRCs in the line. They are computed as the ratio between the jitter measured at the output of each cell and the total expected jitter at that node. In this case, the expected jitter at the output of cell *N* is just *N* times the jitter observed at the output of the first cell (4.7).

$$g_i^{\ c} = \sigma_{t_{d,i}}^{\ c} / \left(i \cdot \sigma_{t_{d,i}}^{\ c} \right) \ , \ i = 1, ..., N$$

$$(4.7)$$

Results can be arranged in look-up tables or fitted into polynomial expressions. For uncorrelated PSN sources, gain elements depend on the noise mode, the noise level (v_n) , design parameters $(r_c \text{ and } r_r)$ and the relative position between the observed cell and the noisy cell (M = i - j). For correlated PSN sources, gain elements have the similar dependencies but now *M* is replaced by the number of cascaded cells (N).

4.2.3 Simulation Results

This section evaluates the accuracy of the proposed jitter accumulation model. Model predictions are compared with simulation results, using a two stage symmetric H-tree implemented in a 90nm technology. The tree was designed assuming an uniform load distribution and geometric wire sizing. Each clock path has five inverter-based CRCs, with $r_c = 4$. Jitter accumulation was then evaluated with transient simulations, using low-frequency PSN sources with different modes, amplitudes and correlations.

The repeater's gain functions were obtained using (4.6) and (4.7). Results are shown in Fig. 4.11 for $g_{k1}{}^{u}$ and $g_{k}{}^{c}$, for k = 2, ..., 6. For uncorrelated noise sources, jitter amplification is shown to be almost constant after the second cell and higher for DMN than for CMN. The most relevant design parameter is the resistance ratio (r_r), which significantly reduces jitter accumulation. This means that interconnect resistance is beneficial for jitter accumulation when PSN sources are uncorrelated. When noise sources are correlated, r_r also has a beneficial impact on DMN jitter accumulation but the most relevant parameter is the noise level (v_n). Moreover, jitter gain for CMN is shown to be very small. Note also that the capacitance ratio (r_c) does not have a significant influence on gain parameters.



Figure 4.11: Jitter gain for A ($v_n = 5\%$, $r_c = 2$, $r_r = 0$), B ($v_n = 5\%$, $r_c = 10$, $r_r = 1$), C ($v_n = 5\%$, $r_c = 8$, $r_r = 0$) and D ($v_n = 10\%$, $r_c = 8$, $r_r = 0$), for: a) uncorrelated noise sources; and b) correlated noise sources.

Fig. 4.12 compares simulation results with model predictions, for uncorrelated and correlated PSN sources, with $v_n = 3\%$. Plots show that the proposed model can predict jitter accumulation bounds with good accuracy. This accuracy is graphically represent in Fig. 4.13a and Fig. 4.13b, with x-y plots. The x-axis corresponds to model predictions while the y-axis corresponds to over 30 simulation results, obtained with $v_n = 3\%$ and $v_n = 6\%$ for uncorrelated and correlated noise sources. One can see that jitter is well estimated, with most points falling above the 45 degrees line. The model error, calculated as a percentage of the simulation results, is shown to be inferior to 10% in all the experiments. Note that individual jitter estimates were obtained using the scalable jitter model described in Chapter 3 (section 3.3), which also contributes to this final prediction error.

The model was further used to predict jitter in clock trees, designed with variable interconnect parameters (C_{int} and R_{int}), wire sizing techniques (geometric or uniform) and chip sizes. As it is not practical to graphically represent the results for all of these experiments, the model's accuracy is shown with x-y plots in Fig. 4.13c and Fig. 4.13d. These results were obtained with over 30 jitter measurements for uncorrelated MMN sources, in trees with two and three stages. The model provides good jitter estimations, with an error inferior to 10% in all the experiments. Although this may seem a large error, the proposed model is much more accurate than the conventional statistical accumulation



Figure 4.12: Dynamic jitter model predictions compared to simulation results, for: a) uncorrelated noise sources; and b) correlated noise sources.



Figure 4.13: Model accuracy in clock trees with: a) N = 5, $r_c = 4$ and uncorrelated PSN; b) N = 5, $r_c = 4$ and correlated PSN; c) and d) variable N, interconnect parameters, wire sizing techniques and chip sizes, with uncorrelated PSN.

model, shown in Fig. 4.7, as discussed there.

The proposed jitter accumulation model can also be useful to grasp the importance of noise correlations on dynamic jitter accumulation. It shows that DMN is beneficial for jitter accumulation only if noise sources in adjacent repeaters are uncorrelated. Circuits with wire-bonded packages, usually have symmetric power and ground noise variations (dominant DMN). However, most PSN in wire-bonded packages is low-frequency and highly spatially correlated [178]. Thus, jitter accumulation can only be reduced if noise sources can be decorrelated, using dithering or similar techniques. On the contrary, for low inductance packages, DMN is dominant only if cascaded repeaters share the same local power distribution parasitics. In this case, noise sources are probably also highly correlated and jitter accumulates fast. However, if clock repeaters are placed in different power blocks, DMN may no longer be dominant. This is beneficial for jitter accumulation if noise sources are correlated in adjacent repeaters, but detrimental if they are independent. In this scenario (repeaters in different power blocks), noise sources are not expected to be totally correlated and one cannot take fully advantage of the beneficial impact of CMN. Nevertheless, this approach can result in a positive net effect because the beneficial impact of CMN for correlated sources is by far more significant than its detrimental effect for uncorrelated sources. Also, this difference becomes more pronounced with the number of cascaded repeaters.

4.3 Clock Deskewing Systems

THE performance of clock deskewing systems is usually discussed only in terms of stability, power consumption and/or maximum static error. To the authors' knowledge, there is still no published work investigating their precision with dynamic variability sources. Because these sources are expected to increase with technology scaling, this is an increasingly important issue when selecting a deskewing technique. This section presents a model that allows the designer to evaluate precision in DLL-based deskewing systems, according to layout constraints and the expected on-chip variability levels. This enables him/her to select the best solution for each application and to evaluate the potential gains provided by the selected scheme, at an early design stage.

4.3.1 Deskewing Uncertainty Model

In a perfectly balanced CDN, the clock arrives to all registers at the same time, so they have the same nominal phase ($\phi_{n_i} = \phi_{n_j} \forall i, j$). In this section, the best known techniques are considered to be used during the pre-silicon design stage to guarantee this condition, both nominally and statistically. These may include load balancing [179], clock scheduling [180] and/or statistical design optimisation [181]. However, the resulting circuit de-

sign will not be ideal for each individual chip because most variability sources are not controllable or predictable prior to silicon. Thus, the post-silicon clock phase (ϕ) will be a RV characterised by a PDF with mean (μ_{ϕ}) and standard deviation (σ_{ϕ}).

According to definitions in section 2.1.2, static jitter is the difference between the mean clock phase and its nominal value ($\delta_{\phi} = \mu_{\phi} - \phi_n$), dynamic jitter is the time-dependent standard deviation (σ_{ϕ}), and skew is the difference between two different clock phases. Thus, static skew corresponds to $\delta_{\phi_{ij}} = \delta_{\phi_i} - \delta_{\phi_j}$, while dynamic skew depends on σ_{ϕ_i} and σ_{ϕ_j} . If one of these phases is considered to be the reference clock phase (ϕ_r), for which $\mu_{\phi_r} = \phi_{n_r}$ and $\sigma_{\phi_r} = 0$, skew becomes equal to jitter. For this reason, absolute clock skew and jitter will be here considered to represent static¹ and dynamic uncertainty, respectively. Absolute skew (δ_{ϕ}) and jitter (σ_{ϕ}) are defined in (4.8).

$$S = \delta_{\phi} = \mu_{\phi} - \phi_n \quad ; \quad J = \sigma_{\phi} \tag{4.8}$$

A DLL based deskewing circuit generally includes a Phase Detector (PD), a Low-Pass Filter (LPF) and a Digitally Controlled Delay Line (DCDL), as shown in Fig. 4.14. Its purpose is to eliminate skew between two SDs, which will be here referred to as the controlled domain (SDc) and the reference domain (SDr). Because the two clocks being compared are not available at the same chip-location, interconnect lines must be carefully routed (possibly with repeaters) to and from the deskewing circuit. The nominal delay introduced by these lines is represented by τ_f and τ_r , with subscripts standing for forward and return clock paths. Inside SDc, the nominal distribution delay is represented by τ_c . Although Fig. 4.14 shows a buffered clock tree inside SDc, this model imposes no restriction on the CDN structure inside controlled domains.

Assuming that SDc is sufficiently small so that internal skew is negligible, any clock sink can be used to measure ϕ_c . The PD compares a delayed version of this clock phase (ϕ'_c) with the reference clock phase (ϕ_r) , and generates the appropriate tuning information to adjust the DCDL. The loop filter controls the circuit operating rate, to avoid instability. In such configuration, the phase difference between domains is eliminated as long as

¹Quasi-static delay variations, induced by temperature or ageing mechanisms, are also considered part of skew because they occur in a much larger time scale than the deskewing circuit's operation cycle.



Figure 4.14: Generic DLL based feedback deskewing circuit.

the delays in forward and return clock paths are integer multiples of the clock period (T_{clk}) , as shown in (4.9). Here, n_r and n_f are integer multiplicative factors ($\in N_0$) and Δ is the DCDL's nominal insertion delay. To minimise uncertainty, but also power and area overheads, it is desirable to have $n_f = 1$ and $n_r = 0$ (ideal situation).

$$\Delta + \tau_f + \tau_c = n_f \cdot T_{clk} \text{ and } \tau_r = n_r \cdot T_{clk}$$
(4.9)

The PD usually consists of an up/down detector. Any skew greater than the PD's threshold ($\pm \epsilon_{pd}$) will trigger either an up or down signal to indicate that the delay control word should be adjusted. In regular time intervals (T_{dsk}) the DCDL is adjusted, increasing or decreasing the forward clock delay by a fixed delay step. The tuning process is active until skew is within the PD's guard-band ($2\epsilon_{pd}$). This eliminates static skew and all the frequency components of dynamic skew that fall within the loop's bandwidth. Residual absolute skew in SDc (S_c) depends only on ϵ_{pd} and on the delay difference between the nominal return path delay and its post-silicon mean value, i.e, on the return path delay skew (δ_{τ_c}). This is shown in (4.10).

$$S_c = \epsilon_{pd} + |\tau_r - \mu_{\tau_r}| = \epsilon_{pd} + \delta_{\tau_r} \tag{4.10}$$

The DCDL's minimum delay adjustment is called the delay step (*d*). The number of possible increments (*M*) multiplied by the delay step corresponds to its dynamic range (Δ_m) . This should be enough to guarantee the condition in (4.9), and compensate for the maximum unexpected skew introduced by PVT variations in the forward interconnect (δ_{τ_c}) and in the SDc clock distribution network (δ_{τ_c}) . This is shown in (4.11).

$$\Delta_m = M \cdot d \ge \Delta + \delta_{\tau_f} + \delta_{\tau_c} \tag{4.11}$$

The absolute and relative magnitudes of *d* and ϵ_{pd} are chosen based on several tradeoffs. To minimise the adjustment error, they should be as small as possible. However, a small guard-band is more susceptible to the effects of jitter, which could cause false triggering of the delay detector. On the other hand, a small *d* increases the lock-in time when skew is large². Regarding their relationship, *d* should be smaller than $2\epsilon_{pd}$ to ensure stable locking. Because these parameters are prone to variability, it is also common to introduce some design margin and make $\epsilon_{pd} \approx d$.

Dynamic clock phase variations that fall outside the loop's bandwidth are not compensated and thus, there will be absolute jitter in SDc. To derive an expression for jitter (J_c), the loop's response to phase error at the PD's inputs must be evaluated. In the following analysis, the instantaneous phase error is considered to be $\theta = \phi_r - \phi'_c$ and the error coherence time (t_{θ}) is the time during which θ is seen by the PD as a constant value. Also, the lock-in time (t_L) is defined as the time the loop needs to reach a stable state, which is proportional to the loop's operation cycle ($t_L = m \cdot T_{dsk}$). Here, parameter *m* corresponds to the number of DCDL adjustments needed to eliminate θ , so $m = \lfloor \theta/d \rfloor$. Considering t_{θ} and t_L , two different situations may occur:

- 1. $t_{\theta} > t_L$: the error is static or changes slowly, so it can be eliminated by the deskewing circuit. If $t_{\theta} \gg t_L$, the error in the transient periods between stable states can be disregarded. Thus, θ is assumed to be completely eliminated and $J_c \approx 0$. If not, jitter is still partially eliminated and $J_c < \sigma_{\phi_c}$, being σ_{ϕ_c} the expected jitter without deskewing.
- 2. $t_{\theta} < t_L$: the loop either becomes unstable or does not respond. In the first case, the loop tries to adjust the delay but never reaches a stable state because the error changes faster than t_L . In the second case, θ changes faster that the time the loop needs to start adjusting. In either cases, θ cannot be eliminated and $J_c = \sigma_{\phi_c}$. If

²If DCDLs are implemented with binary weighted delay elements the system's accuracy can be increased without a significant lock-in time penalty. However, the difficulty to initialise the states and to match delay elements makes them acceptable for coarse adjustment only. In that case, it is common to find a second delay line with conventional ring counter for fine adjustment [182]. Although this model considers only single delay-lines, the same approach can be used to analyse dual-delay line configurations.

the loop is too slow, even quasi-static variations (usually seen as skew) can not be mitigated and jitter may be higher than expected ($J_c > \sigma_{\phi_c}$).

To account for different situations, J_c is here considered to depend on g_{δ} and σ_{ϕ_c} , as shown in (4.12). Parameter g_{δ} models jitter gain/atenuation in slow/fast deskewing systems ($g_{\delta} < 1$ in fast systems and $g_{\delta} \ge 1$ in slow/static systems). Its value depends on how the expected jitter has been defined, i.e., if it includes quasi-static variations or not. Furthermore, parameter σ_{ϕ_c} is defined as the sum of jitter contributions generated in the path between SDr and the registers in SDc. Assuming that the reference clock phase has no jitter ($\sigma_{\phi_r} = 0$), that path includes jitter generated in the DCDL (σ_{Δ}), in the forward clock path (σ_{τ_f}) and in the clock distribution network inside SDc (σ_{τ_c}). Because jitter contributions are added as standard deviations, a correlation parameter (ρ) is here included to account for non worst-case jitter accumulation. Its value varies between zero and one, depending on the correlation between individual jitter contributions.

$$J_{c} = g_{\delta} \cdot \sigma_{\phi_{c}} = g_{\delta} \cdot \rho \left(\sigma_{\Delta} + \sigma_{\tau_{f}} + \sigma_{\tau_{c}} \right)$$
(4.12)

To obtain an expression for σ_{τ_c} , a three step approach is here taken. First, the chip is considered not to be partitioned in SDs and thus, the clock is distributed using a single clock distribution network. This network should be similar to the one considered before in SDc. Second, the chip-wide distribution jitter (σ_{\Box}) is defined as the maximum jitter between clock source and sink. Likewise, the chip-wide distribution skew (δ_{\Box}) is defined as the maximum skew introduced by the clock distribution network, including static and quasi-static components. These uncertainties are known to be proportional to the distribution latency (τ_{\Box}) and thus, to the chip area (A_{\Box}). So, if the chip is divided in two SDs with similar sizes, uncertainty should decrease by 50% in each SD. Finally, applying this reasoning to SDc (with area A_c), σ_{τ_c} and δ_{τ_c} can be modelled as shown in (4.13), with $\alpha_c = A_c/A_{\Box}$.

$$\sigma_{\tau_c} = \alpha_c \cdot \sigma_{\Box}$$
 and $\delta_{\tau_c} = \alpha_c \cdot \delta_{\Box}$, with $\alpha_c = A_c / A_{\Box}$ (4.13)

Results presented in section 4.1.3 have shown that the worst-case PSN induced jitter

in DCDLs is almost constant across designs and depends only on their insertion delay (Δ_m) , noise ratio (v_n) and technology sensitivity (Y_n) . Using this relationship and the expression in (4.11), one can write σ_{Δ} as shown in (4.14). It shows that jitter depends on the DCDL's nominal delay (Δ) but also on the amount of skew it is supposed to eliminate.

$$\sigma_{\Delta} = \mathbf{Y}_n \cdot \boldsymbol{v}_n \cdot \Delta_m = \mathbf{Y}_n \cdot \boldsymbol{v}_n \left(\Delta + \delta_{\tau_f} + \delta_{\tau_c} \right)$$
(4.14)

Finally, the expression for the system's uncertainty (U) is given in (4.15), reflecting absolute skew and jitter in SDc.

$$\mathcal{U} = S_c + J_c = \epsilon_{pd} + \delta_{\tau_r} + g_\delta \cdot \rho \left(Y_n \cdot v_n \cdot \left(\Delta + \delta_{\tau_f} + \alpha_c \cdot \delta_{\Box} \right) + \sigma_{\tau_f} + \alpha_c \cdot \sigma_{\Box} \right)$$
(4.15)

Because the loop operates continuously in time, it has two different lock-in times. During boot time, the DCDL is adjusted to mitigate static components of skew. When the loop reaches stability, it will adjust the DCDL only to compensate for dynamic (lowfrequency) components of skew, as long as their magnitude is higher than ϵ_{pd} . This afterboot lock-in time determines the circuit's ability to eliminate dynamic uncertainty, which has been previously defined as t_L . If the quasi-static skew is considered to be a percentage γ of total skew at the PD's inputs, t_L can be written as shown in (4.16). Here, total skew includes interconnect skew and clock distribution skew inside SDc.

$$t_{L} = T_{dsk} \cdot m = T_{dsk} \cdot \left\lfloor \gamma \cdot \left(\delta_{\tau_{f}} + \delta_{\tau_{r}} + \delta_{\tau_{c}} \right) / d \right\rfloor$$
(4.16)

4.3.2 Impact of Circuit Floorplanning

DLL-based deskewing systems can be floorplanned and implemented as a Local Deskewing System (LDS) or a Remote Deskewing System (RDS). This is shown in Fig. 4.15, where the PD and LPF are represented by a single circuit block, the Skew Controller (SC).

In a LDS, the circuit employs only one DCDL in the forward clock path, which is physically close to both SDr and SDc³. As a consequence, the interconnect delays (and their

³If the domains are not physically close to each other, a second DLL based mechanism could be used to


Figure 4.15: Floorplans for DLL based deskewing systems: a) LDS; and b) RDS.

uncertainty) are negligible and uncertainty can be expressed as shown in (4.17), with the nominal DCDL delay $\Delta_{LDS} = T_{clk} - \tau_c$. Also, the expression for after-boot lock-in depends only on the clock distribution skew inside SDc (4.18).

$$\mathcal{U}_{LDS} = \epsilon_{pd} + g_{\delta} \cdot \rho \left(\mathbf{Y}_n \cdot \boldsymbol{v}_n \left(T_{clk} - \tau_c + \alpha_c \cdot \delta_{\Box} \right) + \alpha_c \cdot \sigma_{\Box} \right)$$
(4.17)

$$t_{L_{LDS}} = T_{dsk} \cdot \lfloor \gamma \cdot \alpha_c \cdot \delta_\Box / d \rfloor$$
(4.18)

In a RDS, the reference and controlled domains are in remote locations so the interconnect delays cannot be considered negligible. To enable clock deskewing in forward and return paths, the deskewing circuit employs two matched DCDLs and one SC that equally adjusts the both lines [184],[185]. The design's symmetry guarantees a temporal symmetry in clock paths. If interconnects are routed side-by-side and the DCDLs implemented in close proximity, the forward and return path uncertainty will also be highly correlated, which guarantees the loop's symmetry even under severe PVT variations.

When activated, the DCDLs are dynamically adjusted to eliminate skew between ϕ_d and ϕ_r . When the loop reaches a stable state, the loop delay is a multiple of the clock period, as shown in (4.19). Here Δ_{RDS} is the nominal DCDL delay and τ_m is the matched line's delay. For the minimum uncertainty, the DCDL delay should be as small as possible. This occurs for n = 1, although it results in a clock phase inversion ($\phi_d = \phi_r + \pi$).

$$2 \cdot (\Delta_{RDS} + \tau_m) = n \cdot T_{clk} \tag{4.19}$$

compensate for interconnect skews [183]. However, this would introduce significant overheads in system complexity, lock-in time, power and area.

Like before, the maximum static error between the clock signals at the PD's input is ϵ_{pd} . However, neither of these clock signals correspond to the controlled clock phase (ϕ_d). The circuit's symmetry guarantees that ϕ_d is exactly in between of PD's input signals, so the maximum static error is $0.5\epsilon_{pd}$. Although this represents a significant accuracy improvement, ϕ_d is not the clock phase distributed in SDc. So, the variability that affect the clock distribution network in SDc are not compensated by the loop, and absolute skew in this design is given by (4.20).

$$S_{c_{RDS}} = 0.5\epsilon_{pd} + \delta_{\tau_c} = 0.5\epsilon_{pd} + \alpha_c \cdot \delta_{\Box}$$
(4.20)

To mitigate the impact of PVT variability, the delay introduced by each DCDL should be sufficient to guarantee the condition in (4.19) and accommodate the unexpected interconnect path skew (δ_{τ_m}), as shown in (4.21). Note that unlike a LDS, the DCDL dynamic range is now unrelated to the size of SDc - it depends only on the interconnect skew that it is supposed to eliminate.

$$\Delta_{m_{RDS}} \ge \Delta_{RDS} + \delta_{\tau_m} = (n \cdot T_{clk}/2 - \tau_m) + \delta_{\tau_m}$$
(4.21)

A longer line is more exposed to PVT variability and will probably need more repeater stages, which increases the line's sensitivity to PSN. To compute the uncertainty associated with clock lines, two parameters are here defined: the average clock line skew per unit length (δ_l); and the average clock line jitter per unit length (σ_l). This unit length corresponds the line's electrical length (delay) and not to its physical length. Using those metrics, interconnect path skew (δ_{τ_m}) and jitter (σ_{τ_m}) can be written as shown in (4.22).

$$\sigma_{\tau_m} = \sigma_l \cdot \tau_m \quad \text{and} \quad \delta_{\tau_m} = \delta_l \cdot \tau_m \tag{4.22}$$

The uncertainty (U_{RDS}) and lock-in time ($t_{L_{RDS}}$) in a RDS are given in (4.23) and (4.24). Note that $t_{L_{RDS}}$ depends only on the magnitude of low-frequency components of skew introduced by one of the interconnects, because DCDLs are simultaneously adjusted.

$$\mathcal{U}_{RDS} = 0.5\epsilon_{pd} + \alpha_c \cdot \delta_{\Box} + g_\delta \cdot \rho \left(\mathbf{Y}_n \cdot v_n \left(T_{clk} / 2 + (\delta_l - 1) \cdot \tau_m \right) + \sigma_l \cdot \tau_m + \alpha_c \cdot \sigma_{\Box} \right)$$
(4.23)

$$t_{L_{RDS}} = T_{dsk} \cdot \lfloor \gamma \cdot \tau_m \cdot \delta_l / d \rfloor$$
(4.24)

A similar design can be obtained if the position of matched lines is exchanged with DCDLs [186], physically separating the lines from the SC. This solution requires one extra long interconnect (the control line), which could compromise the system's precision if ACDLs are used. In a digital implementation the main drawback is the compromise between routing cost and lock-in time. A parallel control scheme is prohibitively costly in terms of routing resources, while a series scheme increases the lock-in time, reducing the system's ability to mitigate dynamic variations (e.g., authors in [139] propose a scheme with series control, but is can be used only for static or periodic deskewing).

The choice between LDS and RDS depends on the expected interconnect and intradomain distribution uncertainty. A LDS can eliminate clock distribution skew inside SDs, but is highly affected by interconnect uncertainty. Thus, it performs better when there is a small number of large SDs placed in close proximity to each other and deskewing blocks can be placed simultaneously close to SDr and to some leaf in the controlled domains. On the contrary, a RDS reduces interconnect skews but cannot eliminate distribution skews inside SDs. Thus, it performs better when there are many small SDs scattered throughout the chip as long as interconnect routing does not become prohibitively expensive. These and other scalability issues are discussed in the next section.

4.3.3 Impact of Synchronisation Topologies

Deskewing schemes are either based on LDS or RDS. However, multi-domain systems may employ a parallel, series or mesh synchronisation topology, which determines the maximum uncertainty between SDs that share sequentially-adjacent registers. This section investigates the impact of synchronisation topology on deskewing uncertainty.

Parallel Synchronisation

With parallel synchronisation, all modules are synchronised to a common reference clock. If the modules are topologically close to each other, the reference domain can be centralised and located in close proximity to the deskewing blocks. This is shown in Fig. 4.16a, which employs several LDSs in parallel. With this scheme, the maximum uncertainty between two SDs (U_{cpc}) is just twice the uncertainty in a LDS and the lock-in time ($t_{L_{cp}}$) is the same as in a LDS. This is shown in (4.25), where subscripts stand for centralised parallel scheme.



Figure 4.16: Parallel synchronization, with: a) centralised SDr; and b) distributed SDr.

$$\mathcal{U}_{cp} = 2 \cdot \mathcal{U}_{LDS}$$
 and $t_{L_{cp}} = t_{L_{LDS}}$ (4.25)

Although this architecture has been successfully used in commercial chips [157, 182], the neighbouring condition is hard to meet in current large VLSI systems. To synchronise large chip areas, parallel synchronisation can be extended using the concept of distributed reference clock. This concept was first presented using transmission line interconnects [187], which were able to synchronise arbitrarily located modules. The same concept was latter adapted to micro-electronic circuit design [188]. The key idea is that the clock can be distributed along the chip area using two matched lines in a ring configuration. Since the lines are routed in parallel and in opposite directions, the midpoint of clock phases travelling the lines is constant at any point on the ring. This clock ring can also be replaced with a distributed PLL [189] or DLL [186].

The architecture proposed in [186] is shown in Fig. 4.16b, using two RDSs in cascade. The first distributes a global reference clock while the second synchronises local domain clocks. Here, these loops are referred to as the global and local deskewing loops. Note that local clocks are taken from the middle point of local DCDLs, which is equivalent to having two DCDLs, one in the forward path and another in the return path. The clock ring allows a greater flexibility in floorplanning multiple SDs, but the global precision is now affected by two cascaded deskewing circuits. In the following analysis, the global ring is assumed to introduce a delay $2\tau_m$ in each direction (a total delay of $4\tau_m$ in the global ring), to match the forward interconnect delay of a simple RDS.

To derive an expression for the worst-case system uncertainty, the two most electrically distant SDs are here considered: one close to the reference clock input (best location) and the other close to the global DCDL (worst location). Note that these domains are physically close to each other (and close to the SC), and can therefore share sequentiallyadjacent registers. Considering a global loop delay equal to one clock period, for the clock signals to be aligned at the PD's input the global DCDL must be able to insert a delay (Δ_G) equal to $T_{clk} - 4\tau_m$. Moreover, it must be able to compensate for skew introduced in the global ring, which can be represented as $4\tau_m \cdot \delta_l$. The maximum DCDL delay can thus be expressed as shown in (4.26).

$$\Delta_{m_G} = \Delta_G + 4\tau_m \cdot \delta_l = T_{clk} - 4\tau_m \cdot (1 - \delta_l) \tag{4.26}$$

Using the same reasoning, the circuit's symmetry forces the local deskewing circuits to adjust their DCDLs to $\Delta_{m_{L,b}} = T_{clk}$, for the best located loop, and $\Delta_{m_{L,w}} = \Delta_{m_G}$ for the worst located loop. Using these expressions, uncertainty in the best and worst located local loops was derived as shown in (4.27) and (4.28), respectively. Skew is proportional to $1.5\epsilon_{pd}$, as it corresponds to the sum of static errors introduced by global and local PDs. Also, note that jitter in local DCDLs depends only on half of their insertion delay because the domain clock is extracted from the line's middle point.

$$\mathcal{U}_{L,b} = 1.5\epsilon_{pd} + \alpha_c \cdot \delta_{\Box} + \rho \cdot g_\delta \left(\mathbf{Y}_n \cdot \mathbf{v}_n \cdot T_{clk} / 2 + \alpha_c \cdot \sigma_{\Box} \right)$$
(4.27)

$$\mathcal{U}_{L,w} = 1.5\epsilon_{pd} + \alpha_c \cdot \delta_{\Box} + \rho \cdot g_\delta \left(2\tau_m \cdot \sigma_l + Y_n \cdot v_n \cdot \Delta_{m_G} / 2 + \alpha_c \cdot \sigma_{\Box} \right)$$
(4.28)

The worst-case system uncertainty corresponds to the sum of absolute uncertainties obtained for the most distant local loops. However, the static error introduced by the global loop should not be considered, as it equally affects the precision of both local loops. The complete expression for the maximum system uncertainty (U_{dp}) is shown in (4.29), where subscripts stand for distributed parallel. Note that the resulting expression is twice the uncertainty in a RDS, as expected from the system's architecture.

$$\mathcal{U}_{dv} = \mathcal{U}_{L,b} + \mathcal{U}_{L,w} - 2\epsilon_{vd} = 2 \cdot \mathcal{U}_{RDS}$$
(4.29)

To avoid instability, global and local loops should operate in hierarchy. Some kind of arbitration should then be used to guarantee that the global loop locks-in before the local loops start operating. So, the worst case system lock-in time corresponds to the sum of global and local loop's lock-in time, considering the worst location for the local loop (close to the global DCDL). The total lock-in time ($t_{L_{dp}}$) is shown in (4.30).

$$t_{L_{dn}} = T_{dsk} \cdot \left\lfloor \left(\gamma \cdot 8\tau_m \cdot \delta_l\right) / d \right\rfloor \tag{4.30}$$

Series Synchronisation

In a series synchronisation scheme, the deskewing circuit is placed at the input of each domain to compensate for skews introduced by the CDN in that domain. This is appropriate when intra-domain skew is negligible and each domain can obtain its reference clock from a nearby clock leaf, on an adjacent domain. This scheme is depicted in Fig. 4.17a, for a cascaded hierarchy. The reference clock is fed to the reference domain (SDr), from which the closest deskewing system gets its clock (DSK1). Similarly, other deskewing systems get their clocks from their neighbours, although those clocks are not the real reference clock. Nevertheless, if intra-domain skew is negligible, those clocks can be considered to be good time references.

The cascaded series topology is based on a common LDS, but its precision is deteriorated by the number of synchronisation levels (L_s) separating SDs with sequentiallyadjacent registers. Thus, both the uncertainty (U_{cs}) and lock-in time ($t_{L_{cs}}$) depend on L_s ,



Figure 4.17: Series synchronization with: a) cascaded hierarchy; b) H-Tree hierarchy.

as shown in (4.31). In these expressions, subscripts stand for cascaded series.

$$\mathcal{U}_{cs} = L_s \cdot \mathcal{U}_{LDS}$$
 and $t_{L_{cs}} = L_s \cdot t_{L_{LDS}}$ (4.31)

In Fig. 4.17b, an alternative series synchronisation scheme with an H-tree hierarchy is shown [190]. Each tree level employs a LDS, which reduces skew between adjacent domains to within the PD's guard-band. The main advantage of this scheme is that clock domains are synchronised to each other, so there is no need for the DCDLs to add delay in the clock path to compensate for the clock distribution delay. As a consequence, it is enough that $\sum \Delta_{m_i} \ge \delta_{\Box}$, which reduces jitter inserted by DCDLs and simultaneously saves power and area. However, this scheme requires a global clock distribution network and thus, $\sigma_{\tau_f} \neq 0$ (unlike a common LDS).

Furthermore, the hierarchical nature of deskewing makes static uncertainty dependent on the number of levels between the most electrically distant domains, which can be physically close to each other in this configuration. If the chip is divided in N_c local clock domains, the H-tree has $L_s = \sqrt{N_c}$ levels. To reduce uncertainty, authors in [134] proposed a similar configuration with a ring control scheme in each quadrant, which reduces L_s to $\sqrt[4]{N_c}$. In either case, the final expression for worst-case system uncertainty (\mathcal{U}_{ts}) is shown in (4.32), where subscripts stand for tree series. The maximum accumulated skew depends on the maximum number of PDs between two neighbouring SDs, which is $2L_s - 1$. Because two neighbouring SDs may have totally separate clock paths, the maximum jitter is twice the jitter in a common LDS, with $\Delta_m = \delta_{\Box}$ and $\sigma_{\tau_f} \neq 0$.

$$\mathcal{U}_{ts} = \epsilon_{pd} \left(2L_s - 1 \right) + 2g_{\delta} \cdot \rho \left(\alpha_c \cdot \sigma_{\Box} + \sigma_{\tau_f} + \mathbf{Y}_n \cdot v_n \cdot \delta_{\Box} \right)$$
(4.32)

The lock-in time ($t_{L_{ts}}$) depends on the total skew that the DCDLs are supposed to eliminate in each branch, because domains at level *i* have to be deskewed before the domains in level *i* – 1, and so forth (hierarchical deskewing). Thus $t_{L_{ts}}$ is given by (4.33).

$$t_{L_{ts}} = T_{dsk} \cdot \lfloor \gamma \cdot \delta_{\Box} / d \rfloor \tag{4.33}$$

Mesh Synchronisation

If DCDLs are allowed to be controlled by several Skew Controller (SC) outputs, a mesh synchronisation topology can be implemented as schematically represented in Fig. 4.18 [190]. It employs an H-tree clock distribution, but other clock distribution structures are possible. The domain clocks are compared to their neighbours (up to four) and a control signal is generated to adjust the DCDL's delay. To ensure a stable lock, the line's dynamic range should be enough to accommodate chip-wide clock distribution skew, so again $\Delta_{m_m} = \delta_{\Box}$. When the system arrives to a stable state, the maximum uncertainty between neighbouring domains (\mathcal{U}_m) is shown in (4.34).



Figure 4.18: Mesh synchronization: a) global H-tree; b) local SDs; and c) deskewing units.

$$\mathcal{U}_{m} = \epsilon_{pd} + 2g_{\delta} \cdot \rho \left(\alpha_{c} \cdot \sigma_{\Box} + \sigma_{\tau_{f}} + Y_{n} \cdot v_{n} \cdot \delta_{\Box} \right)$$
(4.34)

Compared to a tree series deskewing scheme, this has a smaller static uncertainty because there is only one PD between neighboring domains. However, it may have higher dynamic uncertainty due to its longer lock-in time (which increases g_{δ}). The bound for

			0
	Skew (S_{ij})	Jitter (J _{ij})	t_L/T_{dsk}
\mathcal{U}_{cp}	$2\epsilon_{pd}$	$2g_{\delta} \cdot \rho \left(\mathbf{Y}_{n} \cdot v_{n} \left(T_{clk} - \tau_{c} + \alpha_{c} \cdot \delta_{\Box} \right) + \alpha_{c} \cdot \sigma_{\Box} \right)$	$\lfloor \gamma \cdot \alpha_c \cdot \delta_{\Box} / d \rfloor$
\mathcal{U}_{dp}	$\epsilon_{pd} + 2\left(\alpha_c \cdot \delta_{\Box}\right)$	$2g_{\delta} \cdot \rho \left(\tau_{m} \cdot \sigma_{l} + Y_{n} \cdot v_{n} \left(T_{clk}/2 - \tau_{m} \left(1 - \delta_{l}\right)\right) + \alpha_{c} \cdot \sigma_{\Box}\right)$	$\lfloor \gamma \cdot 8\tau_m \cdot \delta_l / d \rfloor$
\mathcal{U}_{cs}	$L_s \cdot \epsilon_{pd}$	$L_{s} \cdot g_{\delta} \cdot \rho \left(\mathbf{Y}_{n} \cdot v_{n} \left(T_{clk} - \tau_{c} + \alpha_{c} \cdot \delta_{\Box} \right) + \alpha_{c} \cdot \sigma_{\Box} \right)$	$L_s \cdot \lfloor \gamma \cdot \alpha_c \cdot \delta_{\Box} / d \rfloor$
\mathcal{U}_{ts}	$\epsilon_{pd} \left(2 \cdot L_s - 1 \right)$	$2g_{\delta} \cdot \rho \left(\alpha_{c} \cdot \sigma_{\Box} + \sigma_{\tau_{f}} + Y_{n} \cdot v_{n} \cdot \delta_{\Box} \right)$	$\lfloor \gamma \cdot \delta_{\Box} / d \rfloor$
\mathcal{U}_m	ϵ_{pd}	$2g_{\delta} \cdot \rho \left(\alpha_{c} \cdot \sigma_{\Box} + \sigma_{\tau_{f}} + \mathbf{Y}_{n} \cdot v_{n} \cdot \delta_{\Box} \right)$	$k_1 + k_2 \cdot \lfloor \gamma \cdot \delta_\Box / d \rfloor^*$

Table 4.3: Model for the worst-case static and dynamic deskewing uncertainty.

*: For the mesh scheme, $k_1 = N_c + 2\sqrt{N_c}$ and $k_2 = 2\sqrt{N_c} - 2$

the worst case lock-in time was empirically derived in [190] as $k_1 + M \cdot k_2$, where $k_1 = N_c + 2\sqrt{N_c}$ and $k_2 = 2\sqrt{N_c} - 2$. If *M* is replaced with $m = \lfloor \gamma \cdot \delta_{\Box}/d \rfloor$, which is the number of delay steps needed to mitigate the expected low-frequency distribution skew, the after-boot lock-in time becomes the expression shown in (4.35).

$$t_{L_m} \approx T_{dsk} \cdot (k_1 + k_2 \lfloor \gamma \cdot \delta_{\Box} / d \rfloor)$$
(4.35)

4.3.4 Comparative Analysis

The presented analysis shows that DLL-based deskewing systems are either implemented as LDSs or RDSs, in spite of the multiple schemes that can be found in literature. LDSs are used to eliminate skew between two adjacent SDs, while RDSs eliminate only clock distribution skew. These fundamental schemes can be found in five different topologies: centralised parallel; distributed parallel; cascaded series; tree series; and mesh. This section compares their static and dynamic uncertainty, using the proposed uncertainty model. For the reader's convenience, it is summarised in Table 4.3. For each topology, it shows the worst-case skew (S_{ij}) and jitter (J_{ij}) between adjacent domains, as well as the after-boot lock-in time (t_L).

In mesh and tree series topologies, jitter is directly proportional to the expected onchip distribution skew (δ_{\Box}). Thus, these topologies directly trade static for dynamic uncertainty, which is a clear disadvantage when compared to others. Compared to each other, they apparently have the same expression for jitter. However, g_{δ} is higher in mesh because it has a higher t_L . This means that it is less capable of mitigating quasi-static variations. Yet, the mesh topology has a lower and constant skew, which is a good feature if one needs to synchronise a large number of domains. On the contrary, skew in a tree series topology is higher and proportional to the number of synchronisation levels (L_s).

The distributed parallel topology is the only where jitter is not proportional to δ_{\Box} . Instead it depends on interconnect variability (δ_l) and delay (τ_m). Also, this scheme requires DCDLs with a nominal delay of only $T_{clk}/2$, as opposed to T_{clk} in centralised parallel and cascaded seires, which results in less jitter insertion. However it has two significant disadvantages: 1) it cannot mitigate intra-domain skew; and 2) it requires a ring to be routed throughout the chip, which may not be feasible.

The centralised parallel topology is one of the best in skew reduction, but cannot be implemented when the domains are not physically close to each other and close to the deskewing units. This imposes stringent limits on the maximum number of SDs in this scheme. When N_c is large, an hybrid solution with cascaded arrangements of centralised parallel systems can be used, as proposed in [191]. Yet, this degrades the overall static uncertainty and increases t_L . Regarding jitter, both centralised parallel and cascaded series schemes need DCDLs with significant insertion delay, which may insert significant dynamic uncertainty in high PSN environments.

To illustrate the advantages and disadvantages of different synchronisation topologies, the proposed model will now be used to compute static and dynamic uncertainty in SDs. A reference synchronous circuit is considered to be equally divided in N_c domains, with $\alpha_c = 1/N_c$ and $L_s = \sqrt{N_c}$. Design parameters and performance metrics are shown in Table 4.4. Fig. 4.19a shows results for $N_c = 4$. Note that the initial global circuit skew (δ_{\Box}) is reduced by all topologies, but at the cost of higher jitter (jitter is much higher than σ_{\Box}). The mesh topology is the one with the best overall performance, while centralised parallel and cascaded series are the ones with the worst jitter performance. This occurs because they require a DCDL with significant insertion delay and thus, with higher jitter insertion. However, they are more effective in mitigating skew than distributed parallel or tree series.

		Design	Param	Performance Metrics							
$\epsilon_{pd} = d$	T_{clk}	$ au_{\Box}$	γ	ρ	$ au_c$	τ_m	v_n	δ_{\Box}/T_{clk}	σ_{\Box}/T_{clk}	δ_l	σ_l
5ps	200ps	200ps	30%	70%	τ_{\Box}/N_c	$\tau_{\Box}/4$	4%	10%	2 %	10%	2%

Table 4.4: Design parameters and performance metrics for model evaluation.



Figure 4.19: Skew and jitter as a percentage of T_{clk} : a) reference scenario; b) higher N_c ; c) higher δ_{\Box} and σ_{\Box} ; and d) higher δ_{\Box} , σ_{\Box} and v_n .

Fig. 4.19b shows results for $N_c = 16$. The performance of series topologies is shown to significantly decrease: the cascaded series introduces higher jitter while the tree series ends up inserting more skew than it was supposed to eliminate. On the contrary, the distributed parallel topology introduces less skew and jitter, because they are proportional to the SD's size. Nevertheless, the mesh topology was the one identified as best suited to deal with increasing number of synchronous domains.

Fig. 4.19c compares results when δ_{\Box} and σ_{\Box} are increased, for $N_c = 4$. It can be seen that skew is mitigated again at the cost of increased jitter. The distributed parallel

topology shows the worst skew performance, because it cannot be eliminated inside each SD. However, it is the one with lower jitter. The same occurs when noise levels (v_n) are also increased, as shown in Fig. 4.19d. Again, the topology with the best overall performance is the mesh, which is able to mitigate skew with comparably small jitter insertion. Yet, jitter is shown to be about $4 \times$ larger than it was with a single SD.

Results shown for this illustrative example were obtained using a simple model for g_{δ} , given in (4.36). Because some topologies are sufficiently fast to partially eliminate jitter ($g_{\delta} < 1$), while others may be so slow that even quasi-static variations cannot be mitigated ($g_{\delta} > 1$), g_{δ} should be expressed as a function of t_L . The proposed expression was derived considering the following: a) sufficiently high-frequency jitter components always exist, so jitter is never completely eliminated ($g_{\delta} > 0.6$); b) for the longest t_L , jitter increases at most by 50%; c) skew is always larger than jitter, so eliminating low-frequency components of skew has a residual impact on t_L .

$$g_{\delta} = 0.6 + \sqrt{(\log_{10} t_L)/2} \tag{4.36}$$

Besides being a tool to compare precision in feedback synchronisation schemes, the proposed model can also be used to determine the maximum allowable noise and interference levels for each scheme. For a reliable system operation, jitter should be kept as a small percentage of the clock period. Thus, the model can be used to compute the maximum allowable noise level (v_n) at DCDLs, according to their insertion delay, and evaluate the need for additional PSN filtering. In chapter 6, this model will also be used to evaluate deskewing precision trends with technology scaling.

4.4 Conclusions

T HIS Chapter discussed clock uncertainty in clocking structures. Section 4.1 described different architectures of Digitally Controlled Delay Lines (DCDLs) and compared their performance, considering different metrics. Clock tree design was also briefly addressed. Because clock trees and DCDLs are built with several cascaded Clock Repeater Cells (CRCs), section 4.2 proposed a scalable jitter accumulation model for CRC lines. Finally, uncertainty in deskewing systems was discussed in section 4.3. Models proposed in section 4.2 and 4.3 will be useful in chapter 6, to investigate uncertainty trends with technology scaling. The main conclusions drawn in each of these sections are summarised next.

Simulation results presented in section 4.1 showed that uncertainty in DCDLs is almost constant, regardless the circuit's implementation details and selected delay. This means that for a given technology, jitter in these structures depends mostly on their dynamic range. Based on this observation, a normalised uncertainty parameter was defined. Also, the conventional statistical accumulation model was shown to fail PSN jitter accumulation predictions, even when individual contributions are obtained from simulation results. This happens because it disregards the dual nature of PSN impact on delay a difference in the supply voltage between a driver and receiver pair creates either a positive or negative time shift in the perceived signal transition at the receiver, depending on noise correlations.

To replace time-consuming transient noise simulations when evaluating jitter in clock distribution systems, section 4.2 proposed a modified statistical accumulation model to predict PSN jitter accumulation in cascaded CRCs. Along with the scalable jitter model proposed in chapter 3, this model provided PSN jitter predictions for clock trees that were within 10% of simulation results. This is a much better accuracy than the conventional statistical accumulation model could provide. On the other hand, the proposed accumulation model can give the designer a valuable insight regarding the impact of noise correlations on jitter accumulation. This can be useful to promote floorplan-based power and clock distribution design (to minimise jitter accumulation), which can be particularly effective in bump-bonded and low inductance package styles, where cascaded circuit blocks may be subjected to significantly different power distribution parasitics.

Finally, section 4.3 presented a model to evaluate uncertainty in digital deskewing circuits. DLL-based deskewing systems have been shown to be either LDSs or RDSs. LDSs are used to eliminate skew between adjacent SDs, while RDSs eliminate only clock distribution skew. This fundamental difference impacts both their skew and jitter performance, which can be evaluated using the proposed analytical model. As it depends only on parameters that can be easily obtained from design or early simulation data, it can be incorporated in an automatic tool to determine the best topology for a given application or to evaluate the system's tolerance to power-supply noise.

Chapter 5 Experimental Results

This chapter describes the experimental setup, the methods and timing measurement techniques used to experimentally evaluate the accuracy of the jitter models proposed in chapters 3 and 4. These models can be used to predict jitter in clock repeaters and repeater lines, whether they are part of an IC or a PCB, as long as the key circuit parameters can be extracted. For practical reasons, the proposed models were experimentally evaluated using discrete clock repeaters. As a consequence, results presented in here concern only to PSN jitter.

5.1 Experimental Setup

T HIS section provides detailed information on the repeaters and measurement hardware used to evaluate the proposed models. The PSN framework, relevant measurement techniques and board design solutions are also described.

5.1.1 Hardware and Equipment

Encapsulated digital gates (Little Logic), from Texas Instruments [192], were used to validate the conclusions regarding jitter and uncertainty performance in different Static Delay Repeaters (SDRs). These devices are available in different technology families, with different voltage ranges and timing performance. Results presented here were obtained using gates from Low-Voltage Complementary Metal Oxide Semiconductor (LVC) and Advanced Ultra-low-voltage (AUC) families, hereafter called InvL and InvA, and LVC NAND gates. The supply voltage was set to $V_{dd} = 1.8V$, which is the recommended supply voltage for the AUC family, although the LVC family is optimised for 3.3V operation. Unfortunately, these gates could not be used to evaluate jitter insertion models because they are not single gate repeaters. They include an unknown number of internal tapered repeaters, and thus, output jitter depends both on insertion and accumulation mechanisms.

To evaluate jitter insertion models, inverters were built with matched pair small signal MOSFET arrays from Advanced Linear Devices (ALD1115). This monolithic complementary N-channel and P-channel transistor pair is intended for a broad range of analog applications, including signal switching (CMOS inverter), and can support supply voltages up to $V_{dd} = 13V$. For reasons latter explained, V_{dd} was set to 6V. Transistors have similar threshold voltages ($V_{th} = 0.7V$) and input capacitance ($C_{in,n} = C_{in,p} = 1pF$), but rather different drain to source ON resistance ($R_{ds,n} = 350\Omega$ and $R_{ds,p} = 1200\Omega$ at $V_{ds} = 0.1V$ and $V_{gs} = 5V$). Thus, the resulting inverter gate has a mean channel resistance $R_{on} = 775\Omega$ and an input capacitance $C_{in} = 2pF$. These gates are hereafter called analog inverters as opposed to digital Little Logic inverters.

Different Printed Circuit Boards (PCBs) were designed for the repeaters' characterisation and jitter evaluation phases. Because digital signals have significantly high spectral content, several signal integrity related issues had to be addressed in the PCB's design to ensure reliable results. First, boards were driven by semi-rigid coaxial cables with straight Sub-Miniature version A (SMA) connectors and 50Ω terminations. Second, board interconnects were designed as 50Ω micro-strip lines, to reduce reflections and maintain a good signal integrity. Finally, boards were designed with built-in voltage-divider passive probes in the input and output nodes of the Device Under Test (DUT). This is illustrated in Fig. 5.1, for a PCB with a cascade of five InvA cells, where the intermediate repeater is the DUT.



Figure 5.1: Repeater chain PCB with passive probes: a) schematic; b) photograph.

Passive voltage-divider probes are attractive for their simplicity, with a good performance up to several GHz. Moreover, being purely passive, they do not add any random noise and jitter to the signal being measured (except for the intrinsic resistive thermal noise). However, they reduce the amplitude of the signal transmitted to the measurement instrument, so the instrument's noise has a large relative impact. On the contrary, active probes have no loading-amplitude trade-off but will always add some random noise and jitter to the signal, as well as some parasitic capacitance load. Although neither of these are ideal probes, measurements were performed with passive probes only due to the unavailability of active probes compatible with the measurement equipment.

Using passive probes with analog inverters was tricky due to their rather large channel resistance. Passive probes consist of a resistive voltage divider, made up by a highimpedance resistor (R_s) in series with 50 Ω scope impedance. For Little Logic gates an $R_s = 450\Omega$ was used, which resulted in a reasonable probe ratio of 1:10. However, this R_s is too low for analog inverters, which have a comparable mean channel resistance (R_{on}). To reduce the impact of the probe's resistance and still have a reasonable dividing ratio, an $R_s = 1820\Omega$ was selected for those inverters, which resulted in a ratio of 1:37. This R_s is the one that minimises the probe's impact on circuit operation and still allows us to take advantage of the scope's full vertical span (although at the minimum vertical resolution). Despite the careful selection of R_s , the voltage divider reduced the clock signal dynamic range at the inverter's output (3.3*V*) compared to the supply voltage (6*V*). This resulted in a lower switching speed than was expected from these inverters, but had no other consequence.

Different circuit boards were tested using the setup illustrated in Fig. 5.2. The noise generator apparatus is explained in section 5.1.2. To generate the input clock signal, an Agilent HP81130A pulse generator was used. It generates single-ended clock pulses from 1kHz up to 400MHz, with selectable transition times of 800ps or 1.6ns and configurable amplitude levels. According to the manufacturer data-sheet [193], typical RMS period jitter and baseline noise are equal to 0.001%+15ps and 4mV, respectively, which are too small to interfere with our measurements. To measure jitter and timing parameters, a 20GHz bandwidth Tecktronics digital phosphor oscilloscope (DPO72004) was used, with

50GS/s real time sample rate and 1.43ps delta time measurement accuracy (RMS) [194]. It has a minimal vertical resolution of 10mV/div and a typical vertical noise below 0.6% of full scale (0.6mV). Although this is a very good accuracy, passive probes significantly reduced the measured signal's dynamic range in analog inverters. Thus, vertical noise had to be accounted for when analysing experimental results (as will be latter explained). Also, timing characterisation was performed in real-time sample mode with interpolation and averaging artifacts to maximise the measurement accuracy.



Figure 5.2: Setup used to measure PSN jitter in different circuit boards.

To measure jitter insertion, i.e. how much jitter does a clock repeater add, the repeater's input clock signal was used as the scope's trigger when capturing the output clock signal's histogram around the threshold crossing. The histogram standard deviation is known to be a good jitter metric as long as it follows a Gaussian distribution and sufficient samples are taken. For that reason, at least 5000 samples were taken in each measurement. With this number of samples, one can be reasonably confident that all outliers up to 3σ have been caught [159]. Other jitter metrics, like minimum, maximum or peak-to-peak spread, are not suited for this purpose because they depend on a single data point, even when the distribution itself is made up from millions of data points. On the contrary, the standard deviation depends on all data points together and thus, it is a extremely stable parameter (especially for large data sets).

5.1.2 Supply Noise Generator

To evaluate PSN induced jitter, power and ground random noise sources were built using a Xilinx Field-Programmable Gate Array (FPGA) and two custom Digital to Analog Converter (DAC) boards, as shown in Fig. 5.3. Two independent Gaussian random sequences were generated in MATLAB, following a standard normal distribution (μ =0 and σ =1) with 4096 samples (12-bit resolution), and stored in the FPGA's internal memory. The DAC boards were then used to generate two different noise voltage waveforms (V_{n1} and V_{n2}). Their magnitude, spectral content and relative mode (MMN, CMN or DMN) can be digitally configured by external switches. Each DAC board includes a single-supply 12-bit resolution DAC (AD9762) followed by a high-bandwidth rail-to-rail amplifier (AD8061), both from Analog Devices. Separate analog and digital PDNs reduce the impact of digital switching noise coupling to the output signal waveforms (V_{n1} and V_{n2}), so that they can keep the spectral and statistical characteristics defined by the random sequences stored in the FPGA.



Figure 5.3: Noise generator built with a Xilinx FPGA, custom DAC and daughter boards.

Although V_{n1} and V_{n2} sources are sufficient to evaluate the proposed jitter insertion model, at least fourteen independent noise sources are necessary to evaluate jitter accumulation in a clock tree with 3 stages (N=7). Thus, two additional daughter-boards were necessary, which connect to both the FPGA and DAC boards. Each board receives and demultiplexes the analog noise source V_{ni} into eight independent outputs V_{nij} , using one Field Effect Transistor (FET) demultiplexer (SN74CBTLV3251), eight capacitors and eight high-bandwidth rail-to-rail amplifiers (AD8061). This can be done because the noise samples stored in the FPGA are not correlated in time and thus, their time demultiplexed samples are also independent between each other. To guarantee that V_{nij} have the same bandwidth as the noise waveforms used for individual inverters, the DAC clock frequency was increased eight times when using daughter-boards. DAC and daughter boards are shown in Fig. 5.4.



Figure 5.4: PCBs for a) custom DAC board; and b) custom daughter board.

To evaluate the impact of PSN on jitter, noise waveforms were coupled to the repeater's supply and ground rails. To minimise the impact of the necessary changes in the PDN and thus, on the repeater's output waveform, the noise and signal paths were separated using discrete reactances (L_1 , L_2 , C_1 and C_2), as shown in Fig. 5.5a. When switching, the repeater needs to push/pull current from/to the power supply network. On the other hand, noise has to be coupled to the repeater's supply pins but not to the entire PDN. Thus, the PDN must have a low impedance path for the clock signal's spectral components and a high impedance to PSN components. Because the clock signal has discrete spectral components and the relevant PSN spectral components are below the clock frequency ($f_n < f_{clk}$), that was done using disjoint pass-band (L_2 , C_2) and notch filters (L_1 , C_1), as shown in Fig. 5.5b and Fig. 5.5c.



Figure 5.5: a) Repeater's supply network with noise coupling; b) signal path and signal's transfer function ($H_s(f)$); c) noise path and noise's transfer function ($H_n(f)$).

Noise and signal paths were analysed separately to compute the required value for L_1 , L_2 , C_1 and C_2 . Thevenin's theorem was used to transform the repeater's current source into a voltage source, which has been conveniently transferred to the supply input node. Thus, the clock signal is shown to pass through a notch filter (5.1), which allows all the relevant clock frequencies to be delivered to the repeater's load, while the noise waveform passes through a pass-band filter (5.2), which blocks its DC component and their high-frequency content. Here, $\tau = R_{on}C_L$, $Z_{11} = L_1C_1$, $Z_{22} = L_2C_2$ and $Z_{12} = L_1C_2$.

$$H_s(s) = \frac{v_o}{v_s} = \frac{(1+s^2Z_{11})(1+s^2Z_{22})(1+s\tau)}{(1+s\tau)((1+s^2Z_{11})(1+s^2Z_{22})+s^2Z_{12})+s^2L_1C_L(1+s^2Z_{22})}$$
(5.1)

$$H_n(s) = \frac{v_o}{v_n} = \frac{s^2 Z_{12}(1+s\tau)}{(1+s\tau)((1+s^2 Z_{11})(1+s^2 Z_{22})+s^2 Z_{12})+s^2 L_1 C_L (1+s^2 Z_{22})}$$
(5.2)

The clock frequency was selected according to the repeater's switching speed ($T_{clk} > 10t_{sw}$), to guarantee a good signal integrity. Thus, $f_{clk} = 20MHz$ was selected for Little Logic gates and $f_{clk} = 2MHz$ for analog inverters. Noise frequency was then adjusted so that PSN sources have low-frequency content when compared to the clock signal. Thus, PSN sources were limited to $f_n = 6.25MHz$ and $f_n = 750kHz$, for Little Logic gates and analog inverters, respectively. The appropriate inductors and capacitors for the supply and noise coupling network were then obtained, so that the relevant clock frequency components ($f = nf_{clk}$, with n = 0, 1, 2, ...) may pass through the notch filter, while the low-frequency noise components are coupled to the repeater's supply rails.

5.2 Uncertainty and Jitter Evaluation

This section experimentally demonstrates that uncertainty is rather constant in digital clock repeaters and repeater chains, supporting the conclusions in sections 3.1 and 4.1. Also, the accuracy of the reference jitter model proposed in section 3.2 is evaluated.

							5	-	,						
		Time [ps]		Currer	Current [µA]		Jitter [ps]				Uncertainty [%]				
SDR		t _d	t_{sw}	I_p	I _{eff}		MMN	CMN	DMN		MMN	CMN	DMN		
InvA	FO1	768	1100	411	186		52	42	54		6.71%	5.51%	7.02%		
	FO4	1070	1480	1100	575		69	65	65		6.41%	6.07%	6.04%		
T. T	FO1	2900	1720	502	282		183	93	234		6.33%	3.19%	8.06%		
IIIVL	FO4	4370	3550	930	136		337	207	427		7.71%	4.74%	9.76%		
NAND	FO1	5470	2010	137	128		514	156	790		9.41%	2.86%	14.43%		
	FO4	6830	3180	388	300		800	236	1308		11.7%	3.46%	19.14%		

Table 5.1: Performance metrics in circuit type B, with $\sigma_{psn} = 6.66\% V_{dd}$.

5.2.1 Uncertainty in SDRs

Little Logic gate repeaters are here used to experimentally evaluate the impact of PSN on clock jitter and uncertainty. In Table 5.1, experimental results are shown for each repeater. Circuit parameters were obtained under realistic conditions, with each repeater driving and being driven by similar gates. It corresponds to circuit type B in Fig. 5.6, which is here reproduced from chapter 3 for the reader convenience.



Figure 5.6: Test circuits: a) ideal driver and load; b) realistic driver and load.

Switching time and delay (t_{sw} and t_d) were obtained using the scope's timing measurement facilities, while the peak and effective currents (I_p and I_{eff}) were inferred from peak and mean slew-rate measurements. PSN jitter was evaluated for different noise modes (MMN, CMN and DMN), with $\sigma_{psn} = 6.66\% V_{dd}$ and $f_n = 6.125MHz$. This can be considered low-frequency noise because f_{clk} was set to = 20MHz.

Results show that although absolute PSN jitter significantly increases with fanout, uncertainty is much more stable. Moreover, uncertainty values have a much lower variability than jitter, when comparing different repeaters. For example, the FO1 NAND gate has around 10 times more MMN jitter than InvA but uncertainty is just \approx 2.5% higher. Simulation results provided in sections 3.1 and 4.1 led to the same conclusions. Note

also that these jitter measurements include the combined effects of jitter insertion and accumulation. The accumulation effect can be observed in CMN and DMN jitter metrics. Although CMN induces higher jitter insertion than DMN (as shown in Fig. 3.7), it accumulates slower when noise sources are totally correlated (as shown in Fig. 4.6). Because these repeaters are built with several cascaded gates, sharing the same PDN, CMN jitter values were smaller than those resulting from DMN.

Similar measurements were performed with capacitively loaded Little Logic gates, using single repeater PCBs (circuit type A). Results are shown in Table 5.2. Photographs of circuits type A and B are shown in Figure 5.7, for InvA repeaters. To guarantee balanced transitions, these boards include a source capacitance (C_s) in the input node. Moreover, C_L in circuit type A was selected as the one that induces the same propagation delay shown by the corresponding repeater in circuit type B, following the procedure described in section 3.2.1. Thus, C_L had to be found by inspection of timing parameters. Although the best efforts were made to match the delay shown by repeaters in circuit type B (Table 5.1), results show that they do not match exactly. Nevertheless, jitter measurements agree with the conclusions taken in section 3.2 that is, jitter in circuit type A is similar to jitter in circuit type B.



Figure 5.7: PCBs for InvA evaluation: a) FO1 type B; b) FO4 type B; and c) FOn type A.

Jitter and uncertainty results for Little Logic gates in circuit type A (solid lines) and circuit type B (white icons), are also shown in Fig. 5.8. Both fanout and repeater structure are shown to have a large impact on absolute jitter. NAND gates have more than 10 times the jitter inserted by InvA gates, for the same PSN levels, and it almost doubles when fanout increases from FO1 to FO6. On the contrary, uncertainty is rather constant with fanout and much less dependent on circuit structure.

	Table 0.2. Fertofinance metrico in cheat type H , while $p_{Sh} = 0.0070 t_{all}$.												
		Time	e [ps]	Curre	nt [µA]	J	itter [ps]	Un	Uncertainty [%]			
SDR		t _d	t_{sw}	Ip	I _{eff}	MMN	CMN	DMN	MMN	CMN	DMN		
InvA	FO1	767	872	441	209	42.9	37.3	43.6	5.59%	4.86%	5.68%		
	FO4	1140	1216	926	688	65.5	64.8	58.5	5.75%	5.69%	5.14%		
TreesT	FO1	2890	1618	380	295	194	104	245	6.71%	3.61%	8.47%		
IIIVL	FO4	4350	3348	761	148	322	198	397	7.41%	4.56%	9.13%		
NAND	FO1	5360	2090	144	138	546	167	853	10.2%	3.12%	15.9%		
INAIND	FO4	6640	3393	440	326	774	248	1250	11.7%	3.73%	18.9%		
1000		PSN Jitter [ps]					PSN Uncertainty [%]						
1000 -						15%]						

Table 5.2: Performance metrics in circuit type A, with $\sigma_{vsn} = 6.66\% V_{dd}$



Figure 5.8: Circuit type A metrics for Little Logic gates: a) jitter; and b) uncertainty. Circuit type B metrics are shown with light grey, unconnected icons.

Results presented here follow conclusions taken in sections 3.1 and 4.1, regarding PSN jitter insertion and accumulation in clock repeaters. Also, they show that the simple circuit type A can be effectively used to evaluate jitter in clock repeaters driving and being driven by similar gates (circuit type B), because jitter measurements closely match.

5.2.2 PSN Jitter Evaluation

This section presents jitter results for the analog inverter and compares them with the reference jitter model, proposed in section 3.2. This model is based on four fundamental parameters: load capacitance (C_L); peak current (I_p); effective current (I_{eff}); and PSN magnitude ($\sigma_{v_{o,psn}}$). These parameters were experimentally obtained for the analog inverter, using type A and type B boards. According to the device's data-sheet, the load capacitance for FO1 was considered to be $C_L = C_{in} = 2pF$. Also, this is the capacitance

	Table 5.5. Relevant circuit model parameters for the analog inverter.												
	Circuit	type B			Circuit	type A		Error [%]					
t _d	t_{sw}	I_p	I _{eff}	t _d	t_{sw}	I_p	I _{eff}	t_d	t_{sw}	Ip	I _{eff}		
12.0ns	26.6ns	7.0µA	5.3µA	12.1ns	27.2ns	7.1µA	5.2µA	0.67	2.65	1.47	-2.58		

Table 5.3: Relevant circuit model parameters for the analog inverter

that induces the same delay in both circuits type A and B, as shown in Table 5.3.

The error was computed as the difference between parameters measured at circuit type A and the ones measured at circuit type B, as a percentage of the last. Its small magnitude proves that the analog inverter, loaded with C_L and driven by our clock source, can be used to evaluate the inverter's behaviour when embedded in a real design (being driven and driving other gates). Also, peak and effective currents (I_p and I_{eff}) were obtained from peak and mean (from 10% to 90% V_{dd}) slew-rate measurements, respectively. Peak slew-rate was obtained by measuring transition time in 10% V_{dd} time intervals during a complete signal transition. The PMOS peak current was measured around 30%-40% V_{dd} in a rising transition while NMOS peak current was measured around 60%-70% V_{dd} in a falling transition, as expected. Shown I_p and I_{eff} parameters are the mean between measured NMOS and PMOS currents.

To quantify the noise magnitude at the inverter's supply nodes, its standard deviation was measured in three different scenarios. First, noise was measured with the input clock and external noise sources turned OFF, to measure the system's baseline noise. After, the input clock signal was turned ON, to measure self-induced noise. This is the noise generated by the repeater's switching current, passing through the non-ideal PDN. Finally, external noise sources were turned ON (MMN), to characterise the total PSN at the repeater's supply rails. Noise histograms at the power rail, and their standard deviation as a percentage of V_{dd} ($v_n = \sigma_{psn}/V_{dd}$), are shown in Fig. 5.9. To compute v_n , the scope's noise floor (0.6mV at this vertical resolution (10mV/div)) was subtracted from the measured noise standard deviation σ_n .

In the first and second scenarios, similar noise standard deviations were measured. Thus, a single plot illustrates both in Fig. 5.9a. The only difference between them is the spectral component at $f_{clk} = 2$ MHz, which is nonexistent when the clock is OFF. This means that self-induced noise can be disregarded in these experiments. Also, these plots



Figure 5.9: Spectral and statistical properties of noise in the repeater's power supply nodes: a) external sources OFF; b) external sources ON.

show power supply rail noise only, because noise in the ground rail has shown similar spectral and statistical characteristics. Comparing these plots, one can see that our system's baseline noise is white and much smaller than externally generated noise. External sources induce $\approx 2.3\% V_{dd}$ noise in each power/ground rail, while baseline system noise is just $\approx 0.5\% V_{dd}$ when they are OFF (standard deviation).

Regarding spectral characteristics, external noise sources have their first spectral null well bellow the clock frequency, which is one of our model's requirement. Using (3.6) from chapter 3, the PSN standard deviation is $\sigma_{v_{o,psn}} = 4\% V_{dd}$ when external sources are ON. When they are OFF, baseline PSN is white and has a magnitude equal to $\sigma_{v_{o,w}} = 0.8\% V_{dd}$. Because this baseline PSN is relatively high, and our scope's vertical accuracy is limited by the probe's voltage divider, it was not possible measure the inverter's TCN. However, it will be shown that the TCN sensitivity metric can also be used to predict white PSN, due to their spectral similarity - both TCN and white PSN have high frequency components so jitter is proportional to I_p and not I_{eff} .

In Table 5.4, jitter measurements for MMN and white PSN are compared against model results, using (3.4) and (3.8) with $\sigma_{v_{o,tcn,max}} = \sigma_{v_{o,w}}$. Sensitivity was computed as $\beta_w = C_L/I_p$ for white PSN and $\beta_{psn} = C_L/I_{eff}$ for low-frequency PSN. The error between jitter predictions and simulation results, as a percentage of the last, is also shown. The proposed model is show to accurately predict PSN jitter, for both low- and high-frequency

		,			-			-			<i>u</i> ,		
		Lc	w-freq	uency F	'SN		White PSN						
Fanout	FO1	FO2	FO3	FO4	FO5	FO6	FO1	FO2	FO3	FO4	FO5	FO6	
β [ns/V]	391	475	546	618	724	825	281	341	392	443	520	592	
$\sigma_{t_d,mea}$ [ns]	1.32	1.68	1.95	2.16	2.56	2.90	0.19	0.24	0.27	0.30	0.34	0.39	
$\sigma_{t_d,mod}$ [ns]	1.38	1.68	1.93	2.16	2.52	2.89	0.20	0.24	0.28	0.31	0.37	0.42	
Error [%]	4.2	-0.1	-1.9	-0.3	-1.6	-0.3	4.4	0.4	1.5	4.2	5.5	6.7	

Table 5.4: PSN jitter measurements ($\sigma_{t_d,mea}$) and model predictions ($\sigma_{t_d,mod}$)

PSN. The error for white PSN induced jitter is higher because absolute values are very small and thus, more prone to measurement errors.

Fig. 5.10 compares measurement (symbols) and model results (dashed lines), for PSN jitter and output currents (I_p and I_{eff}). I_{eff} was computed using the model expression given in (3.7), with $V_{th} = 0.7$ and $\xi = 0.92$. Although ξ is a fitting parameter, it has shown a value close to one, as expected. Model predictions are shown to closely follow experimental results, for both jitter and I_{eff} . Also, I_{eff} is shown to follow the same trend shown by I_p , but with smaller absolute value, as expected from simulations (Fig. 3.12).



Figure 5.10: Measurement and model results: a) PSN jitter; b) peak and effective currents.

The impact of unbalanced transitions on output jitter was also experimentally evaluated. The capacitors at the inverter's input and output nodes were changed, so that PSN jitter could be obtained for different transition time ratios (r_{io}). In Fig. 5.11, jitter and timing parameter measurements are shown for r_{io} in the range [0.4, 1.4], normalised to their values when $r_{io} = 1$. As expected, jitter follows the trends of timing parameters (t_d and t_{out}), but increases/decreases slightly faster when r_{io} is varied.



Figure 5.11: Impact of unbalanced transitions on delay (t_d), output switching time (t_{out}) and PSN jitter: a) FO1 inverter; b) FO6 inverter.

5.2.3 CRT Jitter Evaluation

In section 3.2, simulations were used to show that crosstalk has a two-fold impact on PSN jitter. When aggressors are able to increase a victim's line effective capacitance, PSN jitter is expected to increase because t_{out} increases. However, jitter increases less than expected because $r_{io} = t_{in}/t_{out}$ decreases. In this section, experimental measurement are shown to support those simulation results.

Crosstalk induced jitter is here evaluated by measuring its impact on PSN jitter, using the board shown in Fig. 5.12. It has two aggressor lines (La1 and La2) driven by an analog inverter, similar to the one driving the victim line (Lv). The circuit includes a driver to align the aggressor and victim clock signals and guarantee that they have similar switching times. This maximises the impact of crosstalk, which results in more accurate measurements. Moreover, waveforms in aggressor lines can be configured to switch in the same or opposite directions, compared to the victim's clock signal.

Fig. 5.13 shows the victim's timing parameters (r_{io} , t_{out}) and PSN jitter for different aggressor transitions and coupling capacitance (C_c). Results were normalised to values obtained when aggressors are quiet (no crosstalk). The symbol '-' means that the aggressor is not switching, while symbols '/' and '\' represent the aggressor switching in the same and opposite direction compared to the victim's signal, respectively. Values are normalised to parameters obtained for quiet aggressors.



Figure 5.12: Circuit board to evaluate the impact of CRT: a) schematic; b) photograph.



Figure 5.13: Crosstalk jitter measurements with $C_g = 8pF$ and: a) $C_c = 8pF$; b) $C_c = 15pF$.

As expected, jitter is higher (lower) when aggressors switch in the opposite (same) direction and is proportional to the number of aggressors. This linear relationship reflects the linear dependence on the effective output capacitance. This capacitance is known to be minimum ($C_{eff} = C_g$) when aggressors switch in the same direction as the victim, maximum ($C_{eff} = C_g + 4C_c$) when they switch in the opposite direction, and half way inbetween ($C_{eff} = C_g + 2C_c$) when aggressors are quiet [163]. Note also that jitter increases slower than would be expected if it was predicted based only on the C_{eff} variation (represented by the dashed line), because r_{io} decreases for increasing C_{eff} (as discussed in section 3.2). Finally, it can be observed that with a higher coupling capacitance (C_c) jitter has a larger variability in respect to its value for no crosstalk.

5.3 Scalable Jitter Model Validation

This section presents experimental results for jitter insertion and accumulation models, proposed in sections 3.3 and 4.2, respectively.

5.3.1 Jitter Insertion

The scalable Clock Repeater Cell (CRC) jitter insertion model, presented in section 3.3, is based on two key assumptions. First, it assumes that for jitter prediction purposes, an RC loaded repeater is equivalent to a capacitively loaded repeater. This equivalent capacitance (C_{eq}) is the one that captures the CRC's output slew rate up to the 50% V_{dd} threshold. Second, it assumes that design choices having an impact on the CRC's timing, have a proportional impact on output jitter. This means that the timing characterisation of a reference repeater can be used as a scaling function to determine jitter in CRCs with different design parameters (r_c and r_{io}). Experimental results here presented, support both assumptions.

The equivalent circuit model accuracy, for the purpose of predicting jitter in general CRCs, was first evaluated. Jitter was measured in the output node of a single balanced inverter, loaded with a π -model RC network and a load capacitor. The RC network models the interconnect, while the load capacitor models the input capacitance of subsequent stages. For practical reasons, the capacitance values were fixed ($C_L = C_{int} = 2C_{in}$) while the line resistance was varied between zero and 2.6 R_{on} . For each R_{int} , C_{eq} was found as described in section 3.3.1. This C_{eq} was then used to load a similar inverter, and the correspondent jitter insertion measured. For both circuits, transitions were kept balanced using a source capacitor C_s . Fig. 5.14 shows circuit board's pictures and schematics.

In Fig. 5.15a, jitter measurements in the RC loaded inverter (symbols) are compared to results measured in the capacitively loaded inverter (dashed lines). The design space to which the presented results correspond is defined by $r_r \in [0, 2.6]$, $r_{io} = 1$ and $r_c \in$ [4, 8.5]. Despite the difficulties to guarantee balanced transitions and the intrinsic board differences, the error between jitter measurements in these circuits was inferior to 8%. This means that the C_{eq} circuit model can be used for jitter estimation purposes and does



Figure 5.14: Circuit boards to evaluate the equivalent circuit model: a) inverter followed by an interconnect π -model and load; and b) inverter followed by C_{eq} .

not introduce significant errors compared to the interconnect's π -model. Note also that this capacitance is the one that produces the same output switching time during the initial charging/discharging period (t_{swl}) and not the one that produces the same cell delay, as shown in Fig. 5.15b.



Figure 5.15: Comparison between RC inverter and C_{eq} inverter measurements: a) PSN jitter; and b) delay (t_d) and switching time (t_{swl}).

To evaluate the proposed PSN jitter insertion model, jitter and timing parameters were measured in a capacitively loaded inverter with increasing r_c and r_{io} . Normalised results are shown in Fig. 5.16. When $r_{io} = 1$, all three metrics increase linearly with r_c , although the delay trend is the one that most closely matches the trend of jitter. On the contrary, when $r_{io} \neq 1$, jitter increases/decreases much faster than timing parameters. This is shown in superimposed plots, for $r_c \in \{1, 4, 6\}$. This effect has been accounted for in our model (specifically, in the scaling function proposed in (3.20)). Note that Γ_d was defined as the product of both timing parameters (t_d and t_{out}) when $r_{io} \neq 1$.



Figure 5.16: Normalised dynamic jitter, delay and switching time, for different r_c and r_{io} .

5.3.2 Jitter Accumulation

The jitter accumulation model, proposed in section 4.2, depends on the characterisation of gain functions for DMN and CMN, considering correlated and uncorrelated noise sources in cascaded cells. To experimentally evaluate its accuracy, these functions had to be first obtained for the analog inverters. To do that, a cascade of six similar FO1 inverters ($r_c = r_{io} = 1$) was used, with $r_r = 0$. The characterisation procedure explained in section (4.2.2) was then followed, using the board shown in Fig. 5.17.

Gain results are shown in Fig. 5.18. These values were obtained for CMN and DMN, with $v_n = 4\%$, considering uncorrelated and correlated noise sources along the line. Comparing these plots with simulation results presented in section 4.2.3, one can see that the gain functions follow the same trends, although here, DMN gain for correlated



Figure 5.17: Board with a FO1 inverter line, used to characterise gain functions.

noise sources is substantially lower. Note that measurements are not expected to match simulation results. The measured gain parameters correspond to discrete analog inverters (built with encapsulated MOSFETs) while simulation results were obtained for 90nm integrated inverters, with no port or package parameters considered.



Figure 5.18: Measured gain functions, with $r_c = 1$, $r_r = 0$ and $v_n = 4\% V_{dd}$, for: a) uncorrelated noise sources; and b) correlated noise sources.

These functions were then used to predict jitter in a binary tree with three stages (N=7), using the board shown in Fig. 5.19. Each tree node includes two inverters, built with MOSFETs similar to the ones used before. The only difference is that now, a package with two transistor pairs (ALD1105) is used as it simplifies the binary tree construction. Each tree node has its own PDN and an independent passive probe. The board was also built in such way that it is possible to shunt PDNs (in the back side), so that the same PSN

source can be applied to all cells simultaneously. Thus, it can be used to evaluate jitter with uncorrelated and correlated PSN sources.



Figure 5.19: Binary tree board with three stages, i.e., with 7 cascaded inverters.

The same PSN source was first applied to repeaters along the tree (with $v_n = 4\% V_{dd}$), to measure jitter at each junction. Then, daughter-boards were used to generate four-teen independent power and ground noise waveforms and jitter measurements were repeated. Fig. 5.20 compares measurement results (symbols) with model predictions (dashed lines). MMN jitter predictions were computed as the mean value between CMN and DMN bounds. The proposed model's predictions are shown to be very accurate (within 10% of measurement results), despite the tolerances associated with discrete circuit elements and the imperfections associated with the experimental framework. Note also that inverters used in these experiments may come from different wafers and/or different lots and thus, are not as matched as repeaters would be in integrated clock trees.

Trends in Fig. 5.20 and Fig. 4.12 are very similar. If noise sources are uncorrelated, CMN jitter defines the upper-bound while DMN jitter is relatively smaller. On the contrary, DMN jitter grows much faster than CMN jitter if noise sources are totally correlated. Also MMN is shown to fall between CMN and DMN bounds, as expected from simulation results. However, simulated inverters have shown a higher sensitivity to DMN, when sources are totally correlated. As a consequence, the DMN jitter bound for correlated noise sources in Fig. 5.20b is smaller than the bound for uncorrelated sources. Nevertheless, DMN jitter grows faster for correlated noise sources ($8 \times$ after seven CRCs) than any other jitter bound for uncorrelated sources in long repeater lines, where noise sources are highly correlated.



Figure 5.20: Jitter measurements in a binary tree, compared to model predictions, for: a) uncorrelated noise sources; and b) correlated noise sources.

5.4 Conclusions

THIS chapter evaluates the accuracy of models presented in chapters 3 and 4, using experimental results. Section 5.1 describes the hardware, the experimental setup and measurement techniques. The proposed reference jitter model is evaluated in section 5.2, while the scalable jitter insertion and accumulation model is evaluated in section 5.3. Uncertainty has also been shown to be rather constant in SDRs, supporting conclusions taken in sections 3.1 and 4.1. Next, conclusions drawn along this chapter are summarised.

Section 5.1 describes clock repeaters, evaluation boards, laboratory equipment and measuring techniques used in the presented experiments. It discusses the experimental limitations which justify why results in this chapter are limited to PSN jitter. Details regarding the design of evaluation boards and noise generation hardware are also provided, explaining the approach taken to generate multiple noise sources with specific statistical and spectral characteristics.

The reference jitter model, proposed in section 3.2, was evaluated in section 5.2. Experimental results have shown that a repeater loaded with a single capacitance, and driven by an ideal clock source (clock generator equipment), can be used to evaluate jitter insertion in repeaters embedded in real circuits (which are usually driven and drive other gates). This simple circuit model was then used to experimentally evaluate the accuracy of PSN jitter sensitivity metrics. Results have shown that the model can provide very accurate results, with an error below 4% for low-frequency PSN sources and below 7% for high-frequency PSN. Finally, variations in the effective load capacitance of a given repeater, induced by crosstalk delay, have been shown to have a beneficial impact on PSN jitter.

Section 5.3, evaluated the scalable jitter insertion and accumulation models, proposed in sections 3.3 and 4.2. The assumptions on which the jitter insertion model is based were experimentally evaluated, showing the model's accuracy and applicability. Model errors were measured to be within 8% of experimental measurements. The proposed jitter accumulation model was also used to predict jitter in a binary tree. Model predictions have shown an error of less than 10%, compared to experimental results. These can be considered sufficiently accurate results, given the fact that experimental measurements have their own uncertainty sources. First, clock repeaters are discrete elements and thus, inter-repeater circuit parameters have a much higher variability than in integrated circuits. This is relevant because the models depend on a preliminary circuit characterisation. Second, imperfections in the experimental framework (e.g, in noise generator or evaluation boards), introduce errors that do not exist in a simulation environment. Finally, errors can be partially attributed to the granularity imposed by the use of discrete capacitors (in C_{eq} and C_s) and difficulties in measuring accurate timing parameters (due to signal integrity problems).
Chapter 6

Limits and Trends in Synchronous Clocking

As processes shrink, clock speed increases, and die size grows, an increasingly larger percentage of the clock period is being lost to skew and jitter budgets. This chapter evaluates clock precision trends and its impact on synchronous design, considering different scaling scenarios. Jitter insertion and accumulation models, proposed in chapters 3 and 4, will be used, coupled with models for variability sources and their evolution with technology scaling. Results show that the limits are ultimately imposed by dynamic clock uncertainty, which is increasing with technology scaling and cannot be mitigated without significant power and routing overheads. Therefore, technology scaling alone should not be the main driver of the virtuous cycle of decreasing the cost per function in electronic circuits. Solutions at other abstraction levels may enable designers to reduce the ratio cost/function and increase system performance without depending exclusively on dimensional scaling, and all its variability, reliability and power consumption issues.

6.1 Clock Repeaters

THIS section evaluates the impact of technology scaling in circuit parameters affecting the repeater's sensitivity to variability sources. Simulation results are presented, using Predictive Technology Modelss (PTMs) [195] and commercial MOSFET models, showing that jitter sensitivity is increasing in scaled devices. Also sensitivity metrics proposed in section 3.2, are shown to follow simulation results with reasonable accuracy.

6.1.1 Scaling and Circuit Parameters

The continuous increase in the integration density, captured by Moore's Law [196], has been made possible by a dimensional scaling of transistors. The scaling theory developed by Mead [197] and Dennard [198] shows that one can obtain at the same time a higher speed and a reduced power consumption in a CMOS circuit as long as critical dimensions are reduced, while keeping the electrical field constant. According to this theory, the ratio between the original and the new parameter (*S*=old parameter/new parameter) is 1.44 for the feature size (*S*_L) and 0.7 for the supply voltage (*S*_V = $\sqrt{S_L}$) [199]. This implies a *V*_{dd} and *V*_{th} reduction in each new technology generation.

In the last decade, the scaling theory has faced difficulties in keeping the electrical field constant, because V_{th} scaling is limited by off-current requirements and physical constants [4]. To cope with this problem, alternatives to the existing materials and structures have been introduced. These include strain-induced mobility enhancement, high-k gate dielectrics, metal gates and non-classical devices (e.g., Ultra-Thin Body (UTB) Fully Depleted Silicon-On-Insulator (FDSOI), or Multi Gate (MG) MOSFETs)[200]. Also, different techniques can be used to trade speed for power in a given technology [201]. This means that one can no longer assume that S_L and S_V have the values previously referred, nor that they are fixed numbers for a given technology node.

This can be observed in Table 6.1, showing the scaling factors associated with key inverter parameters built with PTMs (180nm down to 16nm) [195] and commercial MOSFET models from UMC (180nm and 130nm) and IBM (90nm and 65nm). Note that most factors have quite different values from the ones assumed by general rules (scaling theory). Thus, general rules can only give a very coarse approximation to reality in nanometric technologies. These factores were computed from simulation results in each technology node, using balanced FO1 minimum-sized inverters. The NMOS transistor was chosen to have the minimum size allowed in each technology, while the PMOS was sized to obtain balanced transitions. This choice on transistor sizing is advantageous for two main reasons. First, these inverters exhibit the worst case TCN jitter, which minimises simulation errors when obtaining this metric. Second, it is the most straightforward way to compare different technologies, which may have different restrictions in respect to NMOS and

		-			-		-				-
				PTM				UMC	IB	M	General
Tech. [nm]	130	90	65	45	32	22	16	130	90	65	Rules
SL	1.38	1.44	1.38	1.44	1.41	1.45	1.38	1.38	1.44	1.38	1.43
S_V	1.38	1.08	1.09	1.10	1.11	1.13	1.14	1.50	1.00	1.00	1.20
$S_{V_{th}}$	1.17	0.98	0.97	0.78	1.06	1.10	1.24	1.24	0.76	0.88	1.20
S_{I_p}	1.24	1.27	1.41	0.93	1.44	1.73	1.68	1.73	0.99	1.39	1.43
$S_{C_{in}}$	1.67	1.53	1.65	1.65	1.42	1.69	1.44	1.73	1.91	1.94	1.43
$S_{t_{sw}}$	1.92	1.30	1.27	1.95	1.06	1.08	0.99	1.53	1.96	1.41	1.43
S_{t_d}	1.80	1.31	1.27	1.84	1.10	1.14	1.09	1.38	1.78	1.36	1.43

Table 6.1: Scaling factors for key inverter parameters, in different technologies.

PMOS sizing. To keep the consistency of results, similar standard performance MOSFETs were chosen in each technology, with the recommended V_{dd} .

Fig. 6.1 shows the most relevant circuit parameters for the simulated inverters, normalised to the PTM 180nm inverter. Solid lines correspond to PTM inverters and dashed lines correspond to inverters implemented with commercial models from UMC and IBM. The left plot includes the inverter's supply voltage (V_{dd}), the threshold voltage as a percentage of the supply voltage ($v_T = v_{th}/V_{dd}$) and the inverter's input capacitance (C_{in}). Parameter V_{th} was obtained as a mean value between NMOS and PMOS threshold voltage with $V_{gs} = V_{ds} = 0.75V_{dd}$, following the model proposed in [75]. The plot in the righthand side, includes the peak current (I_p), time delay (t_d) and switching time (t_{sw}), for a fanout of one inverter ($C_L = C_{in}$). As considered in previous chapters, these parameters correspond to mean values for rising and falling transitions.



Figure 6.1: Parameters for FO1 inverters implemented with predictive and commercial models, normalised to the PTM 180nm inverter.

Presented data shows that CMOS inverters (the basic building block of clock repeaters) are expected to follow the increasing speed trend with technology scaling, but with an increasingly slower pace. The main difference between commercial and predictive technology models is their higher v_T and lower drivability (I_p), which is translated into higher switching time and delay in 180nm and 130nm technologies from UMC. IBM inverters show a better performance because their supply voltage is not scaled (65nm devices use the same V_{dd} as 130nm devices). Regarding PTM parameters, a sudden change in v_T and I_p can be observed between 65nm and 45nm results. This happens because predictive models are partially based on early stage silicon data from industrial applications, which do not necessarily follow smooth theoretical scaling rules.

6.1.2 Trends in Jitter Sensitivity

Trends in physical and environmental sensitivity metrics are investigated in this section, using simulation results for inverters implemented in different technology nodes. Results show that the reference jitter model, proposed in chapter 3, can be used to accurately predict jitter trends with scaling, as long as trends in key circuit parameters can be obtained.

TCN Jitter Sensitivity

TCN jitter was evaluated using the simulation framework described in section 3.1, for the reference inverter. To guarantee consistency of simulation results across technology nodes: a) the seed in the random noise generator was the same for all simulations; b) TCN jitter simulation was performed for 1000 clock cycles, at room temperature; c) switching time as a percentage of clock period was kept constant ($t_{sw} = 20\% T_{clk}$); and d) balanced transitions were guaranteed ($t_{in} = t_{out}$).

Fig. 6.2 shows results for TCN jitter ($\sigma_{t_{d,ten}}$) and uncertainty (U_{ten}) in inverters with increasing fanouts, using predictive and commercial models. Both absolute and relative jitter metrics are shown to generally increase with scaling. This means that intrinsic clock precision deteriorates with scaling because faster devices generate more noise, which is not fully compensated by their higher peak slew-rate.



Figure 6.2: TCN precision metrics for inverters implemented with predictive and commercial models: a) absolute jitter ($\sigma_{t_{d,tcn}}$); and b) uncertainty (U_{tcn}).

		UMC	& IBM									
Tech. [nm]	180	130	90	65	45	32	22	16	180	130	90	65
$\mu\epsilon_{tcn}$ [%]	0.39	0.14	-0.22	0.17	-0.60	0.41	0.12	-0.14	-0.18	-0.27	0.17	-0.66
$\sigma \epsilon_{tcn}$ [%]	0.41	0.75	0.99	0.88	1.84	1.84	0.96	0.79	0.53	0.44	0.37	0.18
k _i	0.48	0.47	0.47	0.45	0.42	0.42	0.41	0.41	0.46	0.45	0.45	0.44

Table 6.2: Reference inverter's TCN jitter model error.

Simulation results have also shown to fit the proposed reference jitter model's predictions, across technology nodes. The error (ϵ_{tcn}) was computed as the difference between jitter predictions and simulation results, as a percentage of the last. Its mean ($\mu\epsilon_{tcn}$) and standard deviation ($\sigma\epsilon_{tcn}$), for fanouts in {1...6}, are shown in Table 6.2. It also provides the exact input voltage parameter $k_i = V_{in}/V_{dd}$ used to measure peak noise, at each technology node. One can see that both $\mu\epsilon_{tcn}$ and $\sigma\epsilon_{tcn}$ are very small, meaning that the proposed model is sufficiently accurate to predict TCN jitter trends with technology scaling. Also, no significant correlation between the error and fanout was observed.

PSN Jitter Sensitivity

PSN sensitivity with technology scaling has also been evaluated using simulation results. Although PSN is expected to increase in high-performance digital circuits [202], the focus here is on PSN sensitivity. Therefore, PSN sources were scaled according to V_{dd} in each technology node (keeping noise within $10\% V_{dd}$), and shaped to have low-frequency PSDs. Like before, consistency across simulations was maintained: a) simulation time was defined as $T_{sim} = 3000T_{clk}$; b) switching time was set to $20\% T_{clk}$; and c) similar input and output switching times were used ($t_{in} = t_{out}$).

Fig. 6.3 shows simulation results for PSN jitter ($\sigma_{t_{d,psn}}$) and uncertainty (U_{psn}). As before, results for different fanouts are shown using a grey scale. Because PSN was considered to scale with V_{dd} , absolute jitter is shown to decrease with technology scaling, following power supply scaling. Nevertheless, jitter grows faster than the inverter's delay and thus, uncertainty increases in each generation. This means that if repeaters are used to introduce delay, they will do an increasingly worse job as technology scales.



Figure 6.3: PSN precision metrics for inverters implemented with predictive and commercial models: a) absolute jitter ($\sigma_{t_{d,nsn}}$); and b) uncertainty (U_{psn}).

Like before, simulation results were compared with the reference jitter model predictions. Table 6.3 presents statistical data on the error between predictions and simulation results, compared to simulation results (ϵ_{psn}). It gives the error's mean ($\mu\epsilon_{psn}$) and standard deviation ($\sigma\epsilon_{psn}$) for fanouts in {1...6}, and the effective current fitting parameter (ξ), used in each technology node. Contrary to TCN results, the model has consistently shown to over-estimate jitter for fanouts higher than three or four, and under-estimate jitter for lower fanouts. Also, this effect was more evident in 22nm and 16nm technologies, which have shown $\sigma\epsilon_{psn} \approx 5\%$. This is a consequence of considering a very simple I_{eff} model, with a weak dependence on fanout. However, the model has been shown

PTM											UMC &	z IBM	
Tech. [nm]	180	130	90	65	45	32	22	16		180	130	90	65
$\mu\epsilon_{psn}$ [%]	0.53	-0.87	-0.74	-0.94	0.35	-0.44	0.50	-0.37		-0.67	-0.10	1.49	0.78
$\sigma \epsilon_{psn}$ [%]	2.42	2.22	2.12	2.80	2.92	2.89	4.68	5.01		1.44	2.37	4.84	4.69
ξ	1.05	1.02	1.02	1.03	1.06	1.03	1.02	1.01		1.15	1.18	1.13	1.14

Table 6.3: Reference inverter's PSN jitter model error.

to reasonable predict jitter trends with scaling and thus, the cost of lower accuracy is compensated with the model's generality and simplicity.

The trends associated with the normalised PSN uncertainty (Y_{psn}) were also evaluated. This parameter was proposed in section 4.1.3, as the ratio between PSN uncertainty (U_{psn}) and the noise level (v_n). For the reader's convenience, it is reproduced in (6.1). Because it was shown to be fairly constant in circuits implemented in a given technology, it was considered to represent the technology's sensitivity to PSN. To evaluate the trends of Y_{psn} , and to see if that assumption holds with scaling, several simulations were conducted using PTM inverters. Results for increasing σ_{psn} , are shown Fig. 6.4a. It can be observed that Y_{psn} is fairly constant for low noise magnitudes and increases exponentially for larger σ_{psn} . This behaviour is similar for all technology nodes, but the exponential rise starts earlier for scaled technologies (the dotted line indicates where $v_n = 8\%$).

$$Y_{psn} = U_{psn} / v_n = \left(\sigma_{t_{d,psn}} / t_d\right) \cdot \left(\sigma_{psn} / V_{dd}\right) = \left(\sigma_{t_{d,psn}} \cdot V_{dd}\right) / \left(t_d \cdot \sigma_{psn}\right)$$
(6.1)

Results for noise samples with increasing cut-off frequencies ($f_n = 1/T_n$) are shown in Fig. 6.4b. Each curve corresponds to a different technology node, where $T_{clk} = 20t_{sw}$. Thus, the frequencies where small peaks occur ($T_n = T_{clk}$ and $T_n = 0.5T_{clk}$) are not the same for all technologies. Nevertheless, results presented in these plots confirm that Y_n in a given technology is almost constant for low-frequency noise ($f_n < f_{clk}$) with typical noise levels ($v_n < 10\%$). Thus, it can be considered to reflect the implementation technology's PSN sensitivity. Moreover, Y_n is shown to increase with technology scaling, meaning that sensitivity to PSN increases with scaling regardless the circuit architecture.



Figure 6.4: Normalised PSN uncertainty (Y_{psn}) scaling trends with increasing: a) noise standard deviation (σ_{psn}); and c) cut-off frequencies (f_n).

CRT Jitter Sensitivity

To investigate CRT sensitivity trends, a reference inverter loaded with a single capacitance (C_v) was used. The input transition time (t_{in}) was kept constant and equal to the output transition time when $C_v = \mu_c$ (situation with no crosstalk). Then, C_v was varied in $[0.5\mu_c, 1.5\mu_c]$ and the delay as a function of output capacitance obtained, for each technology node. Simulation results were fitted into linear functions of the type $y = m \cdot x + b$, where m represents the inverter's delay sensitivity to variability in C_v . Table 6.4, shows parameters m and b for different technologies, along with the fitting Root Mean Square Error (RMSE). One can see that the slope is almost constant across technology nodes ($m \approx 0.5$), with only a slight decreasing trend. This means that scaling has no significant impact on the inverter's sensitivity to load variability. Thus CRT jitter is expected to increase with technology scaling only if C_v variability increases, which depends essentially on the ratio between coupling and ground capacitance and on the probability of simultaneous switching in neighbouring wires.

Variability in C_v also affects the inverter's balance and thus, has an impact on output jitter (e.g., PSN jitter). To investigate this, the scaling trends in the inverter's delay and balance ratios ($r_d = t_d/t_{d,nom}$ and $r_{io} = t_{in}/t_{out}$) were evaluated, in FO4 inverters implemented with PTM transistors. Plots in Fig. 6.5 show that the impact of unbalanced transitions in PSN jitter is not expected to increase significantly with technology scaling.

PTM										UMC & IBM				
Tech. [nm]	180	130	90	65	45	32	22	16	18) 130	90	65		
т	0.63	0.60	0.59	0.58	0.53	0.51	0.49	0.50	0.5	0 0.50	0.50	0.47		
b	0.36	0.40	0.41	0.42	0.47	0.49	0.50	0.50	0.4	8 0.50	0.50	0.53		
RMSE [×10 ⁻³]	6.80	5.86	6.23	6.47	4.23	4.52	5.71	7.12	7.1	2 2.43	3 1.55	1.04		





Figure 6.5: Performance metrics in FO4 inverters: a) normalised PSN jitter as a function of r_{io} ; and b) $r_d = t_d/t_{d,nom}$ and r_{io} , as a function of C_v/μ_c .

6.2 Clocking Structures

TNTERCONNECT has become a major bottleneck in current digital circuits, because the properties of the wires do not scale with technology in a favourable way [203]. Thus, distributing a high precision clock signal to electrically distant circuit blocks has become a difficult problem. This section investigates the precision limits of traditional CDNs and the underlying synchronous design paradigm.

6.2.1 Clock Trees

Jitter insertion and accumulation models, proposed in chapters 3 and 4, are used here to discuss the performance limits in direct clock distribution networks. Models were implemented in Scilab, which is an open source numerical computational package, to evaluate the impact of alternative designs. Fig. 6.6 presents the flow-graph of this simulation framework. The only topological restriction is that trees must be a cascade of CRCs and thus, they must not include grids, links between regions nor other passive/active deskewing systems. The simulation loop can be manually interrupted or automatically suspended by a predefined stop condition. Because it is based on analytical models, this loop-based approach is computationally inexpensive.



Figure 6.6: Scilab simulation framework to evaluate precision in clock trees.

First, the clock tree is synthesised according to the user's design options, implementation technology data, chip size and load distribution. The synthesis algorithm computes the repeater's sizes, the sink capacitance driven by each branch and the interconnect parasitics. Using this clock tree netlist (or any other previously synthesised), CRCs are identified according to their definition. Each cell is characterised by the repeater's size, π -model parasitics and load capacitance. The methodology presented in section 3.3 is then used to compute the equivalent capacitance and obtain the key circuit parameters (r_c , r_r and r_{io}) associated with each CRC. This framework makes use of an additional resistive model, similar to the one proposed in [85], to obtain accurate values for the input switching time (from 10% to 90% of full swing) of cascaded repeaters.

With these parameters and k-factor equations describing the reference repeater's delay (t_d) and switching times (t_{sw}), path delay (t_D) and skew can be easily computed. Note that these k-factor equations are usually already available in technology library files. After this step, jitter performance is evaluated using the proposed scalable jitter insertion and accumulation models. These models require a pre-characterisation of dynamic and static jitter on different structures: a) a reference repeater, for the CRC's jitter insertion model; b) a reference RC interconnect in each metal layer used in the clock tree, for the C_{eq} variability mapping; and c) a line with a few cascaded repeaters, for the CRC's jitter accumulation model. Note that although characterisation data must be obtained with rather computationally expensive Monte Carlo and/or transient noise simulations, it is required for reference structures only and is performed only once for a given technology.

To estimate the clock tree's power consumption a scalable current model proposed in [20] was used. It is based on the characterisation of the current profile of a reference repeater. To reduce the amount of storage requirements, the profile is matched to the symmetrical triangular approximation, characterised by a peak current (I_p), duration (D_p) and position (P_p). Design ratios are then used to compute each cell's current consumption and the overall power consumption.

Table 6.5, provides performance results for different H-trees, using characterisation data obtained for an IBM 90nm reference repeater and top-level interconnects. The number of stages (N_{stg}), fanouts and interconnect widths were varied. For each set of options, the simulation framework selects the design solution that provides the maximum clock frequency and computes the correspondent performance. The framework can also be easily modified to search the best design solution for low-power instead of high-speed. Design options are the chip size (A_{\Box}), load capacitance per unit area (C_{\Box}), maximum clock repeater size compared to the reference repeater (S_{max}) and type of repeaters. In these experiments, inverters and buffers (with tapering factor ζ) were used as repeaters, with geometric wire sizing in the top metal layer and Gaussian MMN sources, with a

	Desig	n Option	IS		-	Timing				Power		Are	Area		
Ch	ip	Repe	ater	Tree	PSN	t_D	f _{clk} [0	GHz]	P_m	Pp	P.				
A_{\Box}	C_{\Box}	Туре	S_{max}	Nstg	(ho)	[ps]	nom	max	[mW]	[mW]	$\frac{1}{P_m}$	S_r	W _r		
1	10	Inv	120x	3	0	170	2.58	2.18	16.4	34.4	2.1	342x	1x		
				2	1	135	2.40	1.78	26.7	43.8	1.6	449x	2x		
1	10	Inv	60x	3	0	198	2.41	2.00	9.36	20.2	2.2	589x	1x		
1	10	ші	00X	2	1	157	2.22	1.58	15.8	23.3	1.5	741x	2x		
n	10	Inte	120.	3	0	193	2.09	1.78	18.1	36.8	2.0	249x	1x		
2	10	ші	120X	2	1	166	1.85	1.41	27.4	41.8	1.5	470x	2x		
4	10	Int	2404	4	0	270	1.63	1.40	21.1	57.2	2.7	622x	1x		
4	10	ші	240X	3	1	223	1.53	1.17	29.0	52.0	1.8	794x	1x		
4	F	Int	2404	4	0	256	1.67	1.43	22.4	78.1	3.5	587x	1x		
4	5	ШV	240X	2	1	170	1.55	1.23	46.1	82.9	1.8	832x	4x		
1	10	Buf	120.	3	0	292	3.48	2.59	11.0	32.5	2.9	372x	1x		
1	10	$\zeta = 2$	120x	2	1	233	2.89	1.41	17.8	31.4	1.8	438x	3x		
1	10	Buf	2405	2	0	281	4.19	2.77	22.3	48.8	2.2	895x	3x		
1	10	$\zeta = 4$	240X	2	1	265	4.26	1.69	20.7	47.0	2.3	760x	2x		

Table 6.5: H-tree performance for different design options.

Note: Chip area (A_{\Box}) and capacitance (C_{\Box}) are given in $[mm^2]$ and $[pF/mm^2]$, respectively.

standard deviation equal to $10\% V_{dd}$.

Timing performance was evaluated in terms of path insertion delay (t_D) and clock frequency (f_{clk}) for: a) best case jitter - PSN sources are uncorrelated between adjacent CRCs ($\rho = 0$); and b) worst case jitter - PSN sources in adjacent cells are totally correlated ($\rho = 1$). The nominal clock frequency corresponds to the maximum ideal frequency when no jitter is considered while the maximum real frequency takes jitter into account. Results show that the design that achieves the maximum clock frequency for $\rho = 1$ is different from the faster design when $\rho = 0$. If $\rho = 1$, a tree with less stages can distribute a higher frequency clock because jitter accumulation is smaller. On the contrary, a higher number of stages can be used if $\rho = 0$, with savings in power, implementation and routing areas.

Power performance is evaluated with mean (P_m) and peak (P_p) power consumption, which are important metrics not only in what concerns to power savings but also to autoinduced PSN. Resistive voltage drops are proportional to the total current flowing to clock repeaters (evaluated with mean and peak power consumption metrics), while inductive drops depend on current consumption variations (evaluated with the peak-to-mean ratio). These drops may induce higher variability in the clock tree and further reduce the circuit's maximum frequency, so power and timing metrics should be analysed simultaneously. Area performance metrics are also shown, as the total repeater's size ratio and interconnect's width ratio. The former corresponds to the sum of repeater sizes in a clock path compared to the reference repeater size ($S_r = S_{path}/S_{ref}$), while the latter is the minimum used interconnect width compared to the minimum possible width in the target technology ($W_r = W_{int}/W_{min}$).

The proposed framework can also be used to evaluate the impact of choosing a different maximum repeater size, repeater type or pipeline effort (which varies C_{\Box}). Results in Table 6.5 show that buffers increase the tree's nominal f_{clk} , when compared to inverters. Different tapering factors can trade-off area and power for similar results in terms of f_{clk} . However, the real maximum frequency in these trees is much lower than the nominal frequency due to higher jitter insertion and accumulation. Jitter insertion is higher in tapered buffers because internal inverters are unbalanced. Also, jitter accumulates faster in these repeaters because internal inverters are affected by correlated PSN sources.

Another issue is the impact of the synchronisation area (A_{\Box}) on the clock tree's performance. To increase system integration, improve performance and reduce design cycles, most high-performance VLSI systems today include multiple, simpler and more efficient SDs (e.g., in multicore processors or SoCs). This reduces the clock load and simplifies the clock distribution problem. Nevertheless, the proposed framework can help achieving a better global solution for the CDN inside each SD, evaluating the performance impact of using different repeater architectures and spacing, fanouts or alternative interconnect design styles.

Fig. 6.7 shows the impact of A_{\Box} on the most relevant performance metrics. Reducing A_{\Box} can simultaneously increase the clock frequency (both nominal and maximum metrics) and reduce power consumption and power ratio. However, jitter as a percentage of clock cycle (i.e., uncertainty) can increase with chip partitioning. For example, the best solution for a circuit with $A_{\Box} = 6mm^2$ is a two stage H-tree, where uncertainty is around 8% of the clock period (with $\rho = 1$). If this area is partitioned in 3 smaller areas (with $A_{\Box} = 2mm^2$ each), the clock frequency increases. However, the best solution is still a

two stage tree and uncertainty is now twice the value it was before (increases from 8.5% to 17%). This examples shows that in some situations, uncertainty can limit speed gains offered by chip partitioning.



Figure 6.7: H-tree performance for increasing synchronous domain area (A_{\Box}).

As die area is not expected to decrease, nor is circuit density, clock trees alone cannot distribute a high precision clock signal in modern chips. To reduce clock uncertainty, most CDNs today are hybrid structures with clock trees associated to clock meshes, spines or links between regions [117]. Alternative paths created by these additional interconnects, smooth out the difference between clock arrival times and reduce delay variability. However, they are used at the cost of additional power and routing resources. Thus, better results could be achieved if uncertainty could be reduced in the first place - it is always easier to improve a good system than a fair one. Even if uncertainty cannot be reduced, having accurate information regarding the CDNs performance, prior to introducing averaging structures, can save power and routing resources.

6.2.2 Trends in Jitter Accumulation

Uncertainty in clocking structures depends both on jitter insertion and accumulation. Section 6.1 has shown that jitter insertion in clock repeaters is expected to increase with technology scaling, due to a higher sensitivity to variability sources. Thus, although repeaters may switch faster, the uncertainty associated with those transitions becomes a larger percentage of the clock period. This section will show that jitter amplification also increases in scaled devices. This means that jitter accumulates faster in scaled CDNs, even if the number of clock repeaters is not increased.

Jitter along a repeater line is here evaluated with three FO1 inverters in open- and close-loop configurations. In open-loop, the circuit is driven by a clean reference clock signal, while the close-loop arrangement is just a three inverter ring oscillator. Circuits were simulated using the same TCN and MMN sources described before. Simulation results include jitter and uncertainty after the third inverter in the open-loop circuit, and the average clock frequency (μ_{fosc}) and its standard deviation (σ_{fosc}) for the close-loop. Fig. 6.8, compares the open-loop time uncertainty ($U_{OL} = \sigma_{t_d}/t_d$) with the close-loop frequency uncertainty ($U_{CL} = \sigma_{fosc}/\mu_{fosc}$). Results show that noise sources have an increasing impact on both open- and close-loop clocking structures, with technology scaling.



Figure 6.8: Open- and close-loop uncertainty for: a) TCN sources; and b) PSN sources.

For the open-loop circuit, jitter at the output of the third cell ($\sigma_{t_{d3}}$) depends on jitter accumulated along the line. Thus, TCN and MMN jitter generated in each cell were measured to compute the expected output jitter, using the conventional statistical accumulation model. Parameter $\sigma_{t_{d3,ten}}$ was obtained as the square root of the sum of individual TCN variances (uncorrelated noise sources), while $\sigma_{t_{d3,psn}}$ was computed as the sum of individual standard deviations (correlated sources). The ratio between measured and expected jitter results is shown in Table 6.6, for TCN (η_{ten}) and PSN (η_{psn}) induced jitter.

The increasing trend in parameters η_{tcn} and η_{psn} show that the error introduced by the conventional statistical accumulation model actually decreases with scaling. This

10010 0101 10000 2		11100100		en p e e e			nee m	0110101
Technology Node	180nm	130nm	90nm	65nm	45nm	32nm	22nm	16nm
η_{tcn}	0.93	0.96	0.99	0.96	0.97	0.99	1.01	1.05
η_{psn}	0.51	0.53	0.54	0.59	0.71	0.76	0.78	0.78

Table 6.6: Ratio between measured and expected jitter after three inverters.

may result from one of two phenomena: 1) the impact of CMN may be decreasing with scaling and thus, the conventional statistical accumulation model is increasingly more accurate; or 2) scaling brings CMN and DMN jitter bounds closer and thus, the error between the conventional model predictions and measurement results becomes smaller. To investigate these hypothesis, several circuit simulations were performed with cascaded PTM inverters, using $v_n = 5\%$, $r_c = r_r = 1$, $T_{clk} = 20t_{sw}$ and $T_n = 4T_{clk}$.

Fig. 6.9 shows the ratio between MMN jitter and the sum of CMN and DMN jitter bounds, measured along the repeater line. It has been shown in chapter 3 that MMN jitter is midway between bounds, so this ratio should be around 50% for all technology nodes. However, results show that this assumption actually depends on noise correlations and on the implementation technology. When sources are uncorrelated, scaling brings the MMN jitter ratio closer to 50%, meaning that the impact of CMN and DMN is increasingly balanced. On the contrary, the impact of DMN is higher than CMN (MMN jitter ratio is higher than 50%) when sources are totally correlated, and it increases with scaling. This means that the first hypothesis in the previous paragraph is correct, and the beneficial impact of CMN for correlated sources decreases with scaling.



Figure 6.9: Ratio betwen MMN jitter and the sum of CMN and DMN jitter bounds, in cascaded repeaters, and: a) uncorrelated noise sources; and b) correlated noise sources.

To test the second hypothesis, jitter accumulation gain parameters were obtained for correlated (g_i^c) and uncorrelated noise sources (g_{ij}^u), according to definitions given in chapter 4. Figures 6.10 and 6.11 show the trends associated with these parameters. CMN gain is shown to slightly increase while DMN gain decreases noticeable. Results for the 16nm node do not exactly follow this trend, but it would be premature conclude anything different because these are only predictive technology models. Thus, the second hypothesis can also be considered correct - the conventional statistical accumulation model error becomes smaller with scaling because jitter bounds become closer to each other.



Figure 6.10: Scaling impact on jitter amplification gain for correlated noise sources, and: a) CMN sources; b) DMN sources; and c) MMN sources.



Figure 6.11: Scaling impact on jitter amplification gain for uncorrelated noise sources, and: a) CMN sources; b) DMN sources; and c) MMN sources.

Finally, it is important to notice that MMN gain is shown to increase for both correlated

and uncorrelated noise sources. Thus, considering the results shown here and in section 6.1, it is reasonable to expect higher dynamic clock uncertainty in scaled technologies due to both jitter insertion and accumulation mechanisms.

6.3 Discussion

RESULTS presented in sections 6.1 and 6.2 show that dynamic uncertainty is expected to increase with technology scaling. Because it cannot be successfully mitigated without significant power and routing overheads (which are increasingly expensive resources), it poses a fundamental limit on the synchronous clocking paradigm. This section discusses dynamic uncertainty trends with technology scaling, using different scenarios for the evolution of variability sources and technology scaling trends.

6.3.1 Jitter Trends in Clock Repeaters

Jitter trends depend on the evolution of variability sources and the system's sensitivity to those sources. Although the first cannot be generally predicted without specific information regarding system's architecture, it is possible to predict the trends associated with sensitivity. This section evaluates trends of jitter sensitivity, using the reference jitter model proposed in section 3.2, coupled with models for variability sources and their evolution with technology scaling. Because the reference model depends only on parameters that can be easily obtained, it can be used to predict performance degradation in advance to technology migration, allowing the designers to consider beforehand the necessary counter-measures. Also, because it has been heuristically derived, predictions are fairly accurate in nanometric technologies where the multitude of second-order effects prohibitively increases the complexity of analytical models.

For the reader's convenience, the reference jitter model is shown in (6.2) and (6.3). It will be used in this section to predict TCN, PSN and CRT jitter trends using two scenarios for the evolution of $\sigma_{v_{o,ten}}$, $\sigma_{v_{o,psn}}$ and crosstalk parameter k_c/\sqrt{M} . In scenario A, variability sources are considered to be constant with technology scaling. In this situation, jitter evolution depends only on sensitivity factors. In scenario B, more realistic scaling trends

Parameters	Interconnect	Interconnect [nm]		ctric	C	Capad	citance [fF/mm]	
Tech	$W_{int} = S_{int}$	T_{int}	h[nm]	k _{eff}	(Cc	Cg	β_{crt}/t_d
180nm	320	640	672	3.75	9	2.9	63.1	0.236
130nm	225	360	315	3.30	5	9.1	81.2	0.187
90nm	138	234	206	3.35	6	3.7	77.5	0.197
65nm	68	122	109	3.10	6	4.3	67.6	0.207
45nm	45	81	72	2.75	5	7.0	60.0	0.207
32nm ⁽¹⁾	27	51	46	2.60	5	7.6	53.6	0.216
22nm ⁽²⁾	19	38	34	2.30	5	4.2	45.0	0.223
22nm ⁽²⁾	14	27	24	2.15	5	0.7	42.1	0.223

Table 6.7: ITRS intermediate interconnect's parameters and capacitances.

(1) Manufacturable solutions are known.

(2) Manufacturable solutions are not known.

are considered: TCN is inversely proportional to channel length, so its scaling factor is $S_{tcn} = 1/S_L$; PSN follows slew-rate scaling trends, so $S_{psn} = S_V/S_{tsw}$; and both k_c and M increase with feature size shrinkage so crosstalk scales with $S_{crt} = \sqrt{S_L}/S_L$. In both scenarios, T_{clk} is considered to scale with t_{sw} .

$$\sigma_{t_d} = f\left(\sigma_{v_{o,tcn}} \cdot \beta_{tcn} ; \sigma_{v_{o,psn}} \cdot \beta_{psn} ; \left(k_c / \sqrt{M}\right) \cdot \beta_{crt}\right)$$
(6.2)

$$\beta_{tcn} = \frac{C_L}{I_p} \; ; \; \beta_{psn} = \frac{C_L}{I_{eff}} \; ; \; \beta_{crt} = t_d \cdot \frac{C_{crt}}{C_{gt} + C_{crt}} \cdot \sqrt{\frac{t_{sw}}{T_{clk}}} \tag{6.3}$$

In these scenarios, interconnect parasitics are considered to scale according to ITRS predictions for minimum sized intermediate interconnects in high-performance integrated circuits. Table 6.7 shows the ITRS data for the interconnect's geometry (width (W_{int}), spacing (S_{int}), thickness (t_{int})) and inter/intra layer dielectric characteristics (height (h) and effective constant (k_{eff})). It also gives C_c , C_g and the ratio β_{crt}/t_d , for a constant $T_{clk} = 10t_{sw}$, computed with the ITRS general interconnect model for RC delay evaluation [4]. Note that the column on the right is almost constant, meaning that the ratio $C_{crt}/(C_{crt} + C_{gt})$ is not expected to increase with scaling.

To discuss jitter evolution in these scenarios, a reference synchronous system will be here considered. It operates at a reference clock frequency ($T_{clk} = 1$) and is implemented in a 250nm technology (technology node immediately before the 180nm node). In this system, the clock signal is affected by variability sources, so that TCN jitter is $0.1\% T_{clk}$, and both PSN and CRT jitter are $1\% T_{clk}$. Fig. 6.12 shows the expected clock jitter and uncertainty trends for scenario A, where variability sources do not scale (S = 1). Solid lines correspond to sensitivity metrics computed with PTM scaling factors (shown in Table 6.1), while dashed lines were obtained applying ITRS speed scaling predictions (17% frequency increase per year down to the 45nm node and 8% increase thereafter) [4].



Figure 6.12: Scaling trends considering constant variability sources (scenario A) for: a) absolute jitter; and b) uncertainty.

Although absolute jitter is shown to decrease in every new generation, clock precision steadily deteriorates with scaling for PSN and TCN. In these plots, clock precision is represented by uncertainty, computed as the ratio between jitter and the clock period. The uncertainty increasing trend results from the fact that sensitivity to noise sources decreases slower than T_{clk} . On the contrary, CRT sensitivity is shown to follow clock period scaling because it depends almost exclusively on t_d scaling. Data used to compute CRT sensitivity is shown in Table 6.7.

Jitter and uncertainty trends for scenario B are shown in Fig. 6.13, using PTM and ITRS scaling factors. In this scenario, both absolute and relative clock precision metrics deteriorate with scaling except for absolute CRT jitter, which follows the decreasing trend of t_d . PSN and TCN uncertainty are shown to increase by one order of magnitude between the 180nm and the 45nm node. This means that dynamic jitter increases faster than the device's switching speed and can virtually eliminate the performance gain introduced by

technology scaling. Note also that in this scenario, TCN uncertainty becomes comparable to PSN uncertainty at smaller technologies. This is relevant because contrary to PSN, intrinsic noise sources cannot be mitigated by design.



Figure 6.13: Scaling trends considering increasing variability sources (scenario B) for: a) absolute jitter; and b) uncertainty.

Regarding the mismatch between predictive and ITRS results, the difference results essentially from t_d and t_{sw} scaling predictions. While ITRS predicts a constant decrease in timing parameters (supporting the desired speed scaling), measured values for I_p and I_{eff} in PTM inverters were smaller than it would be necessary to guarantee ITRS speed scaling predictions. Thus, TCN and PSN jitter sensitivity parameters do not decrease as fast as the clock period is expected to, and uncertainty is higher than ITRS predicts.

Results presented here show that sensitivity to dynamic variability sources will increase in scaled devices. Because PSN jitter accumulation is also expected to increase, clock precision will deteriorate with technology scaling. This means that clock frequency can only be improved if variability sources can be reduced below current levels, which is not a reasonable assumption in current multi-million gate designs. The easiest solution to extent the fully synchronous design paradigm is thus to divide the system in multiple SDs, controlled by a high-level synchronisation scheme. However, dynamic uncertainty will also affect the precision of these schemes, as discussed next.

6.3.2 Jitter Trends in Synchronous Systems

The last resource to maintain the synchronous paradigm in current high-performance VLSI designs is feedback-based clock distribution architectures, i.e., using multiple SDs associated to a clock deskewing scheme. To evaluate the impact of technology scaling on the precision of these schemes, this section investigates the trends of deskewing uncertainty considering four different scaling scenarios. Fig. 6.14 illustrates these scenarios.



Figure 6.14: Scaling scenarios with higher clock frequency or more SDs.

First, scaling is assumed to be used to accommodate larger functionality on-chip, so the chip size and clock frequency are maintained with technology migration. Design reuse is commonplace to reduce development costs. Thus, larger functionality means a higher number of SDs (Intellectual Property (IP) modules). This is hereafter called scenario D (for higher number of domains). Second, dimensional scaling is assumed to be used to reduce chip area and increase its clock frequency. In this case, the functionality is the same and thus, the chip integrates the same number of SDs. This will be called scenario F (for higher frequency). For each scenario, two possible noise evolutions are further considered, described in section 6.1 as scenario A and B. The first considers that noise sources do not scale, while the second considers them to increase with scaling.

To allow a direct comparison between deskewing architectures, T_{dsk} is considered to be the same for all schemes and SDs are considered to have the same size ($\alpha_c = 1/N_c$). Regarding scaling, some assumptions are also made. First, scaled devices are considered to be faster by S_V and not S_L , to reflect the slower speed scaling trends in nanometric devices (S_{t_d} =1.19). Second, CDNs and interconnects are considered to be optimally buffered,

	0				-						
Parameter	A_{\Box}	T_{clk}	$\epsilon_{pd} = d$	α _c	τ_{\Box}	$ au_m$	δ_{\Box}	σ_{\Box}	δ_l	σ_l	v_n
Scaling factor	'S										
S_{DA}	1.00	1.00	1.00	1.41	0.92	0.92	0.65	0.77	0.71	0.84	1.00
S_{DB}	1.00	1.00	1.00	1.41	0.92	0.92	0.65	0.65	0.71	0.71	0.84
Example											
90nm node	1mm ²	500ps	5ps	0.25	500ps	125ps	25ps	10ps	5%	2%	4%

Table 6.8: Scaling factors for model and circuit parameters in scenarios DA and DB.

so that the wire delay matches the gate delay (t_d) in each stage. Because wire delay is proportional to the square of wire length, the number of buffers necessary per unit length increases with the root of t_d . Third, parameters γ and ρ are assumed to be constant with scaling ($S_{\gamma}=S_{\rho}=1$). Finally, both process variations and sensitivity to variability sources are considered to increase with scaling, for all scenarios considered.

Table 6.8 shows the scaling factors (S=old parameter/new parameter) considered in scenarios with more domains (DA and DB). Because the clock frequency is considered to remain constant, there is no need to improve the deskewing accuracy and $S_d = S_e = 1$. Also, improving accuracy has the undesirable side effect of increasing the lock-in time and noise sensitivity. Following the second assumption, the clock distribution latency (τ_{\Box}) and interconnect delay (τ_m) increase with the root of $1/S_{t_d}$ in this scenario (constant A_{\Box}). Chip-wide distribution skew and jitter are considered to increase proportionally to variability sources, sensitivity to those sources, and distribution latency. They increase faster than interconnect skew and jitter, because these metrics are given per unit delay, which also increases in this scenario. In DA (constant noise levels), skew (δ_{\Box} and δ_l) increases faster than jitter (σ_{\Box} and σ_l) because process variability is assumed to be the only PVT source increasing with scaling. The number of domains is considered to increase at the same rate as density, so S_{α_c} is equal to S_L =1.41.

For illustration purposes, the table also shows absolute values for model parameters in an hypothetical 90nm system. This circuit has $A_{\Box} = 1mm^2$, four SDs, $f_{clk} = 2GHz$ and d = 5ps. If it had a single global clock distribution network, τ_{\Box} would be 500ps (one clock period). A global interconnect ring would have the same delay in this circuit $(4\tau_m = \tau_{\Box})$. Clock distribution skew and jitter, were considered to be equal to 5% and 2% of the clock period, respectively. Likewise, interconnect skew and jitter were defined as 5% and 2% of interconnect delay. Parameter Y_n was obtained from simulation results shown in Fig. 6.4, with $v_n = 4\%$. Other parameters were defined as follows: $\tau_c = \alpha_c \cdot \tau_{\Box}$, $\gamma = 30\%$ and $\rho = 0.8$. Different absolute values for model parameters would give different results, but the purpose of this example is just to show how the model can be used to evaluate the uncertainty trends associated with different deskewing schemes and their ability to increase the system's performance and/or reliability, with technology scaling.

Fig. 6.15 presents skew and jitter results for scenarios DA and DB, using five different deskewing topologies: centralised parallel; distributed parallel; cascaded series; tree series; and mesh (refer to section 4.3.4). All schemes are shown to reduce skew well below its maximum value without deskewing $(2\delta_{\Box})$, represented by a dotted line. Skew as a percentage of clock period is constant (or almost constant) for parallel and mesh topologies, because chip partitioning fully compensates the increase in variability sources. On the contrary, series topologies show skew degradation with scaling because a higher number of SDs increase the number of hierarchical levels, and thus, increase skew accumulation.



Figure 6.15: Deskewing scaling trends in scenarios DA and DB for: a) skew as a percentage of clock period; and b) jitter as a percentage of clock period.

Parameter	A_{\Box}	T_{clk}	$\epsilon_{pd} = d$	α _c	τ_{\Box}	$ au_m$	δ_{\Box}	σ_{\Box}	δ_l	σ_l	v_n
Scaling facto	rs										
S_{FA}	2.00	1.19	1.19	1.00	1.30	1.30	0.92	1.09	0.71	0.84	1.00
S_{FB}	2.00	1.19	1.19	1.00	1.30	1.30	0.92	0.92	0.71	0.71	0.84

Table 6.9: Scaling factors for model and circuit parameters in scenarios FA and FB.

Jitter in scenario DB is almost twice as high as in scenario DA, but exhibits similar trends. Mesh and series schemes are the ones where jitter increases faster with scaling, because they have a higher lock-in time and consequently, higher g_{δ} . This reflects their inability to mitigate low-frequency skews when more SDs are used. On the contrary, jitter grows slowly in parallel schemes, specially for the one with a distributed reference domain. However, worst case jitter between two domains is always higher than it would be without deskewing ($2\rho\sigma_{\Box}$) and thus, all schemes trade static for dynamic uncertainty.

When scaling is used to increase the clock frequency (scenario F), both T_{clk} and A_{\Box} are considered to decrease with scaling. The scaling factors for each circuit parameter in this scenario are shown in Table 6.9. T_{clk} is now assumed to scale with t_d , total area to scale with S_{L}^2 , N_c is not considered to scale ($S_{\alpha_c}=1$) and deskewing accuracy (d and ϵ_{pd}) is considered to follow T_{clk} . Distribution and interconnect delays are considered to scale simultaneously with the root of $S_{A_{\Box}}$ and the root of $1/S_{t_d}$. Clock distribution and interconnect uncertainty have the same dependencies as explained for scenario D. However, their scaling factors are now higher because path delays decrease with scaling.

Results for static and dynamic uncertainty in scenarios FA and FB are shown in Fig. 6.16. The same 90nm reference system described above, was used in these plots. The centralised parallel and cascaded series topologies are the ones with higher dynamic uncertainty, because they require DCDLs with larger delays (proportional to T_{clk}). In terms of static uncertainty, the worst scheme is the distributed parallel, because it cannot mitigate inter-domain skews. Although uncertainty values are smaller than those obtained in scenario D, the same trends can be observed - while skew is kept within comfortable levels, jitter increases exponentially with scaling. Like other clocking structures, feedback-based synchronisation systems are here shown to be limited by dynamic uncertainty, which can mitigate their ability to increase clock precision in large synchronous designs.

Figure 6.16: Deskewing scaling trends in scenarios FA and FB for: a) skew as a percentage of clock period; and b) jitter as a percentage of clock period.

6.3.3 The Synchronous Paradigm

The microprocessor is a key system driver for semiconductor products, since it often uses the most aggressive design styles and manufacturing technologies. This section will use it to describe the impact that technology scaling has had on the synchronous design paradigm, and discuss future trends.

Increasing circuit complexity, power consumption and variability have been the main constraints affecting the miniaturisation virtuous circle of the semiconductor industry. This cycle, represented in Fig. 6.17, is powered by the continuous decreasing cost-perfunction, which leads to significant improvements in economic productivity and further investments in technology scaling. According to Moore's Law, with a feature scaling of n one can get $O(n^2)$ more transistors each generation, running O(n) faster. In the past, this enabled the first microprocessors to scale performance with $O(n^3)$, but the system's complexity and size soon reduced that performance increase to $O(n^2)$ and then to O(n). At the turn of the century, microprocessors became so complex and dynamic power consumption so high, that microprocessor could no longer take advantage of device speed scaling and performance stalled [204].

Figure 6.17: The miniaturisation virtuous circle of the semiconductor industry.

To overcome excessive power consumption and system complexity, circuit designers started exploiting parallelism. MPUs now incorporate multiple cores per die, which are smaller and faster to counter global interconnect scaling, and optimised for reuse across multiple applications and configurations. On the other hand, modern MPU platforms have stabilised maximum power dissipation at approximately 120W due to package cost, reliability, and cooling cost issues. Thus, further increases in clock frequency require designers to reduce power waste using multiple circuit level techniques, as multiple V_{dd} domains, clock distribution optimisation, frequency stepping, new interconnect architectures, multiple V_{th} devices, well biasing and block shutdowns among others [4].

Parallelism allowed designers to relax the need for increasing clock frequencies, as relatively cheap parallel hardware resources can be used to increase performance. However, this is not a free lunch. The cheaper hardware provided by technology scaling is also more vulnerable to process variability and noise. On the other hand, circuit-level techniques to reduce power have increased localised power and temperature variations that changed the traditional on-chip PSN profile. As a consequence, dynamic clock uncertainty increased and its impact on circuit performance became less predictable. To circumvent this, most MPUs today employ hybrid clock distribution networks and rely on loosely synchronous design styles. For example, it is common to find independent clock frequencies and distribution styles, with different averaging structures (e.g., spines, grids, etc.), at different parts of the chip [127, 130].

To reduce the impact of variability, deskewing units have been proposed as an alternative to time averaging structures, with lower power and routing requirements. Although their lengthy response times limits their ability to mitigate dynamic skew, they are very effective in eliminating static and quasi-static skew [205]. For this reason, most current cutting-edge MPUs employ some sort of active deskewing [130]. However, section 6.2 has shown that all deskewing schemes trade static for dynamic uncertainty. Thus, their usage is usually complemented with techniques to reduce PSN levels (e.g., dedicated power supplies or on-die voltage regulators) [8]. Moreover, these deskewing units are not used to guarantee chip-wide synchronicity. Instead, they are used inside localised SDs in GALS architectures. GALS design is a natural choice for both MPUs and SoCs, as they are frequently designed with existing synchronous IP modules/cores to improve design productivity and reduce cost.

The challenges faced by MPUs and SoCs depend largely on the application and product markets. Table 6.10 presents the main high-performance system drivers, their architecture, requirements and performance trends [4]. The low-power and multi-technology SoCs segment is here neglected, as they are out of the scope of this thesis. According to these system drivers, it can be observed that the era of sequential computing, where technology scaling was the main driving force, gave way to a new era in which parallelism is at the forefront. Thus, high performance system success is now based more on software breakthroughs in parallel programming than simply on hardware. Nevertheless, achieving the desired performance trends described in Table 6.10 will also require hardware advances in multiple design abstraction levels. These levels, along with digital design domains, are represented in Fig. 6.18 using the traditional Y-chart [206].

Transistor-level optimisation can be a practical solution to reduce the cost per func-

	Examples	High-end gaming and networking applications				
	Architecture	Multiple cores with accelerator engines, and with on- board switch fabric, L3 caches and connectivity modules				
High		Die areas are constant				
Soc		Number of cores increases by $1.4 \times /$ year				
560	Requirements	Core frequency increases by $1.05 \times /$ year				
		Accelerator engine frequency increases by $1.05 \times /$ year				
		Underlying fabrics scales consistently with the increase in number of cores				
	Performance Trends	Processing performance ⁽¹⁾ increases $1000 \times$ between 2009 and 2024				
	Examples	High-end gaming				
SoC	Architecture	A main general-purpose processor, a number of Data Pro- cessing Engines (DPEs) and I/O for memory and chip-to- chip interfaces				
Consumer Stationary		Design productivity improves $10 \times$ for newly designed logic over the next ten years to 2019				
		A main processor is to able to control up to 8 DPEs				
	Requirements	Superior functional flexibility to support adding or mod- ifying functions				
		Die areas are constant (≈ 220 mm ²)				
		Main processor and DPEs have constant circuit complex- ity, so that layout areas decrease with scaling				
	Performance Trends	Processing performance ⁽¹⁾ increases $250 \times$ between 2009 and 2024				
	Examples	Desktop (Cost-Performance (CP)), server systems (High- Performance (HP)) and embedded MPUs as cores in SoC applications (Power-Connectivity-Cost (PCC))				
	Architecture	General-purpose instruction-set architecture				
Microprocessor		Die areas are constant (140mm ² for CP, 260mm ² for HP, 70100mm ² for PCC)				
		The number of logic cores increases by a factor of $1.4\times$ with each technology generation				
	Requirements	The number of logic transistors per processor core increases $1.4 \times$ with each technology generation				
		Memory content (like logic content) doubles with each successive technology generation				
		Layout density doubles with each technology generation				
	Performance Trends	Clock frequency increases by a factor of at most $1.25 \times$ per technology generation				

Table 6.10: Syste	em drivers in	the high-perforn	nance circuit segment.

(1) results from the product of number of cores, core frequency, and accelerator engine frequency.

tion and increase performance. It relies on the fact that, if the logic synthesis tool can use any possible logic function and size, the resulting technology mapping can drastically reduce the number of transistors, improving timing, power and area [207]. Using less

Figure 6.18: Y-chart with digital design domains and levels of abstraction.

transistors per function, one can obtain better performance and simultaneously reduce the cost/function ratio without resorting to technology migration. On the other hand, these area gains can be used to improve computational integrity. Computational integrity involves multiple design considerations, such as testability, reliability, serviceability, recoverability, fail-safe computation, and security. Although multiple specific techniques exist already in these domains, system architects need to integrate them into their designs under stringent power budgets.

At the architecture level, there are two different approaches to reduce the complexity of current high-performance designs and increase their performance. The first is to go back to fully synchronous design using novel clock distribution technologies. These include dedicated clock distribution chips for three-dimensional ICs, travelling-wave, optical or RF distribution [208]. Although more robust to dynamic variability, none of these schemes are currently a viable alternative to GALS in commercial products for three main reasons. First, they have reduced or none CAD tool support; second, they typically require auxiliary circuits in the receiver end, which may eliminate precision gains offered by those structures; and finally, they are efficient in mitigating global clock distribution uncertainty only, which is also mitigated using a GALS architecture.

The second approach would be to accept dynamic uncertainty as an unavoidable

constraint and opt for asynchronous communication between SDs, i.e., NoC architectures [209]. Besides design and verification benefits, NoCs have been advocated to address clocking, signal integrity, and wire delay challenges. In fact, it is nowadays widely recognized that they represent the most viable solution to cope with scalability issues of future systems and to meet performance, power and reliability requirements [210]. Nevertheless, further research is necessary to better understand design trade-offs and accuratly evaluate their performance gains.

System-level optimisations are also possible, but require better use of parallelism. In MPUs, this can be achieved with non-von Neumann architectures for some specific applications [211] or with better architectures for cooperating von Neumann machines (multicore processors). To maximise the performance of multicore processors, it is still necessary to improve the communication scheme between cores and the memory configuration [212]. Also, application-driven selection of the optimum number of cores, and their nature (homogeneous or heterogeneous cores), can bring further performance improvements [213].

Finally, a word on power consumption reduction. Although this is not an obvious priority in high-performance applications, which are generally free from battery life issues, high-performance system designers will have to continue addressing this issue if they want to take advantage of transistor speed scaling. This can be done with traditional circuit level techniques, as Dynamic Voltage and Frequency Scaling (DVFS) [214], or/and new technology solutions, like SOI and Multi Gate (MG) devices. However, the ability to push forward the limits of synchronous systems will also depend on their impact/sensitivity on/to dynamic variability sources, which is yet to be investigated.

6.4 Conclusions

T HIS chapter discussed limits and trends of clock precision in synchronous systems, using the models proposed in chapters 3 and 4. Section 6.1 focused on clock repeaters and their circuit model parameters, while clocking structures were analysed in section 6.2. Different scaling scenarios were used in section 6.3 to predict clock pre-

cision trends in direct and feedback clock distribution systems. These results were then used to discuss future trends in synchronous clocking, considering current ITRS highperformance system drivers. Next, the main conclusions in this chapter are summarised.

Section 6.1.1 evaluated the trends associated with key circuit parameters using inverters implemented in Predictive Technology Models (PTM) and commercial technologies. It has been shown that difficulties in keeping the electrical field constant have changed their traditional scaling trends. Also, circuit parameters were shown not to be fixed numbers for a given technology node and thus cannot be estimated using the generalised scaling theory. In this scenario, the proposed reference jitter model has the advantage of depending only on circuit parameters for which actual scaling factors can be easily obtained.

In section 6.1.2, both TCN and PSN uncertainty were shown to increase with technology scaling. Also, the proposed reference model was shown to provide very accurate jitter predictions, compared to simulation results. The error for Thermal Channel Noise (TCN) jitter predictions was within 3% of simulation results, considering the mean plus 3 times the standard deviation ($\mu \epsilon_{tcn} + 3\sigma \epsilon_{tcn}$) for fanouts in {1..6}. Power Supply Noise (PSN) predictions were less accurate, with $\mu \epsilon_{tcn} + 3\sigma \epsilon_{tcn}$ within 16% of simulation results for the same fanout span, due to the simplicity of the proposed model for I_{eff} . Nevertheless, it has shown to be better than other I_{eff} models, for this purpose. This section has also shown that the normalised PSN uncertainty can be seen as a constant parameter in each technology node, reflecting its PSN sensitivity.

Section 6.2 discussed precision limits in clocking structures. The proposed scalable jitter insertion and accumulation models were used in section 6.2.1 to evaluate jitter in clock trees. Results have shown that those models can help the designer selecting the best solution for its clock distribution tree, avoiding circuit over-design and unnecessary consumption of power and routing resources. Finally, section 6.2.2 showed that PSN jitter amplification in clocking structures is expected to increase with technology scaling, for both open- and close-loop clocking structures. This further supports the key chapter conclusion - that the synchronous design paradigm will be increasingly limited by dynamic uncertainty, as technology scales.

Considering different scaling scenarios and evolution trends for variability sources,

section 6.3 discussed clock uncertainty trends. The analysis presented in section 6.3.1 led to two significant conclusions. First, that sensitivity to dynamic jitter sources will continue increasing with technology scaling. This means that clock repeaters will insert increasing amounts of jitter even if variability sources do not increase. Second, that if variability sources increase as expected, uncertainty will increase exponentially with scaling. This means that uncertainty will reduce the designer's ability to use the potential performance gains offered by device speed scaling. In section 6.3.2, the proposed deskewing uncertainty model was also used to evaluate the performance of alternative deskewing topologies, in different scaling scenarios. Results have shown that regardless of the system architecture, deskewing schemes trade static for dynamic uncertainty, with the additional disadvantage of area and power overheads.

Given these conclusions, section 6.3.3 discussed the trends of high-performance synchronous systems and identified recent techniques at different abstraction levels that can support the virtuous cycle in the electronic industry, in this domain. However, none of those techniques has yet proved to be effective in mitigating the impact of dynamic uncertainty and thus, the synchronous digital design paradigm is expected to become restricted to the design of small and simple modules/cores. This alleviates the clock uncertainty problem within SDs, but may introduce new challenges in their interfaces.

Chapter 7 Conclusions and Future Directions

The main goal of this thesis is to better understand the sources of clock uncertainty in highperformance synchronous systems in order to identify opportunities and strategies for performance improvement and evaluate the limits of the synchronous design paradigm. This chapter summarises the main conclusions of this thesis and proposes possible directions for additional research.

7.1 Conclusions

Technology scaling and the demand for ever-increasing performance has driven CMOS system complexity and power consumption up. Because voltage scaling had to be slowed down in modern technologies, performance evolution has been depending on multiphase clock solutions and parallel processing architectures rather than clock frequency in sequential processing structures. Nevertheless, even with loose requirements in clock frequency increase, the requirement for tight timing control of clock precision has not been alleviated. In fact, higher complexity in power distribution networks, higher process variability and reduced noise margins have increased the difficulty in maintaining clock uncertainty within the traditional 10% budget. To better understand the mechanisms through which clock uncertainty is generated, and evaluate the limits of the synchronous design paradigm, this thesis investigated jitter insertion and accumulation in active circuits at different design levels.

Clock uncertainty is mainly inserted by repeaters in the clock distribution paths from the source to clock sinks, as a result of Process, Voltage and Temperature (PVT) variations along those paths. One contribution of this thesis has been the analysis of static and dynamic uncertainty (i.e, static and dynamic jitter) in the most common Static Delay Repeaters (SDRs) and Tunable Delay Repeaters (TDRs). Although much of the focus has been on the contrast between different repeater structures, the conclusion is that uncertainty (evaluated as the delay variation as a percentage of propagation delay) is rather constant in these circuits. This means that for a given clock path delay, precision is determined essentially by the implementation technology and not by the clock repeater design. To further investigate the jitter insertion mechanism in clock repeaters, analytical jitter models were developed and presented in this thesis. They depend only on simple circuit parameters that can easily be obtained and thus, provide a valuable insight regarding the repeater's key design parameters responsible for jitter insertion, including the gate, load and interconnects.

Another contribution is the analysis of clock uncertainty in clocking structures. The methodology used to evaluate jitter insertion in clock repeaters was extended to repeater chains, showing again that clock precision depends marginally on implementation details. It is essentially determined by path delay and the correlation among noise sources in individual clock repeaters. This thesis proposes a model for dynamic jitter accumulation in delay lines and clock trees, which has a much better accuracy than the conventional statistical accumulation model. Moreover, it gives the designer a valuable insight regarding the impact of noise correlations on jitter accumulation, which can be useful to promote floorplan-based power and clock distribution design (with the objective of minimising jitter accumulation).

Because deskewing systems can be effectively used to mitigate static uncertainty and increase clock precision in synchronous systems, they could not be disregarded. This thesis has shown that DLL-based deskewing systems are either implemented as Local Deskewing Systems (LDSs) or Remote Deskewing Systems (RDSs). However, despite the implementation structure, they end up trading static for dynamic clock uncertainty. To quantify this effect, this thesis proposes a model to evaluate uncertainty in deskewing systems, considering both floorplanning and scalability issues. As it depends only on parameters that can be easily obtained from design or early simulation data, the model can be incorporated in an automatic tool to determine the best topology for a given application or to evaluate the system's tolerance to power-supply noise.
In pursuit of the thesis's main goal, the proposed models have been used to predict clock precision trends using different scaling scenarios and evolution trends for variability sources. Results were then used to discuss future trends in synchronous clocking, considering current ITRS high-performance system drivers. More specifically, this thesis identified dynamic uncertainty as the main impairment to fully synchronous designs for three main reasons. First, sensitivity to dynamic jitter sources is increasing with technology scaling. This means that clocking structures insert increasing amounts of jitter even if variability sources are not considered to increase. Second, deskewing circuits trade static for dynamic uncertainty, which can ultimately render them useless. Finally, alternative solutions to the current aggressive technology scaling have still not proven effective in supporting the virtuous cycle in the electronic industry nor in mitigating the impact of dynamic uncertainty.

As a consequence of these conclusions, the synchronous digital design paradigm is expected to become restricted to the design of small and simple modules/cores, where dynamic uncertainty can be kept within tolerable levels. In this Globally Asynchronous Locally Synchronouss (GALSs) paradigm, the system's performance becomes more dependent on the asynchronous interfaces and functional partition than on the clock frequency of individual synchronous modules. Thus, the focus of high-performance circuit designers will naturally shift from clocking structures to communication modules, and from clock uncertainty to transmission errors.

7.2 Future Directions

This thesis investigated the sources of clock uncertainty in synchronous designs using static CMOS design styles and on-die electrical clock distribution. Although it proposes uncertainty models for both clock repeaters and clocking structures, complementary research in the following topics could help in further understanding the sources of clock uncertainty and the limits of the synchronous design paradigm.

New Device Structures and Materials

For four decades, the semiconductor industry has achieved continuous performance enhancements by shrinking the bulk MOSFET device dimensions, as described by Moore's Law. However, it has become clear that the conventional transistor materials have been pushed to fundamental material limits and new materials, techniques and structures are needed to improve scaled CMOS devices. In the next decade, either extensions of bulk CMOS technology or new approaches such as fully depleted Fully Depleted Silicon-On-Insulator (FDSOI) and Multi Gate (MG) devices will be required to further reduce the cost-per-function and increase the performance of integrated circuits. Although the proposed models are expected to be applicable with these new device structures and materials (as they depend only on circuit-level parameters), further research would be necessary to demonstrate this assumption.

Clock Averaging Structures

Clock averaging structures, like clock meshes or spines, are ubiquitous in most highperformance circuits today. A clock mesh is a grid composed by wires to which the sequential elements are directly connected, while a clock spine can be seen as a one dimensional clock mesh. Clock spines are usually used to take the clock signal from a clock driver, across the chip, to one or more local clock trees/meshes. Both structures can be used to smooth out undesirable variations between signal nodes spatially distributed over a SD, although their ability to mitigate delay variations is highly related to power consumption. It would be interesting to extend our jitter insertion and accumulation models to include hybrid CDNs in different configurations, like spine-grid or tree-grid clock distribution.

Differential CDNs

This thesis focused only on single-ended electrical clock distribution styles, which is the most common approach in commercial chips. However, differential clock distribution styles can have less sensitivity to power supply noise and to manufacturing variations,

which leads to significant savings in skew and jitter [63]. In the current scenario of increasing PVT variability with technology scaling as well as increasing sensitivity to those variations (as shown in this thesis), differential clock distribution may become a viable alternative. Thus, it would be interesting to extend the proposed jitter insertion and accumulation models to include differential buffers and interconnects, as well as differential to single-ended converters.

Alternative CDNs

Most CDNs today employ clock grids and/or clock trees. However, a number of alternative clocking strategies have been proposed, as standing-wave, travelling-wave, optical clock distribution, package clock distribution, among others. These approaches may either be forgotten, become popular in niche applications, or take over as the dominant clock method if technology evolves to makes them more attractive. They are particularly efficient for global clock distribution and thus, can avoid the usage of deskewing schemes when synchronisation is needed among different SDs. In this situation, the limits of synchronous design would be limited by their floor-planning and scalability issues. Thus, further investigation on their timing precision could establish new limits for the synchronous design paradigm.

Bibliography

- [1] R.J. Riedlinger, R. Bhatia, L. Biro, B. Bowhill, E. Fetzer, P. Gronowski, and T. Grutkowski. A 32nm 3.1 billion transistor 12-wide-issue itanium processor for mission-critical servers. In *Solid-State Circuits Conference Digest of Technical Papers* (ISSCC), 2011 IEEE International, pages 84 – 86, Feb. 2011.
- [2] S. Sawant, U. Desai, G. Shamanna, L. Sharma, M. Ranade, A. Agarwal, S. Dakshinamurthy, and R. Narayanan. A 32nm westmere-ex xeon enterprise processor. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pages 74–75, Feb. 2011.
- [3] Joel Hruska. Blood in the water: Nvidia, qualcomm, samsung, and ti prepare for arm war. Technical report, Extreme Tech, 2012.
- [4] ITRS. The international technology roadmap for semiconductors. Technical report, ITRS Website [Online]. Available: http://public.itrs.net, 2010.
- [5] Herb Sutter. The free lunch is over: A fundamental turn toward concurrency in software. Technical report, Microsoft, 2009.
- [6] Kelin J. Kuhn. Cmos transistor scaling past 32nm and implications on variation. Technical report, Intel Corporation, 2010.
- [7] E.G. Friedman. Clock distribution networks in synchronous digital integrated circuits. *Proceedings of the IEEE*, 89(5):665 –692, May 2001.

- [8] Shenggao Li, A. Krishnakumar, E. Helder, R. Nicholson, and V. Jia. Clock generation for a 32nm server processor with scalable cores. *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2011 IEEE International, pages 82–83, 2011.
- [9] Frank P. O'Mahony. 10GHz global clock distribution using coupled standing-wave oscillators. PhD thesis, Stanford University, CA, 2003.
- [10] G. Le G. de Mercey. 18GHz-36GHz Rotary Traveling Wave Voltage Controlled Oscillator in a CMOS technology. PhD thesis, Bundeswehr Munchen University, 2004.
- [11] A. Mule, S. Schultz, T. Gaylord, and J. Meindl. An optical clock distribution network for gigascale integration. In *IEEE 2000 International Interconnect Technology Conference*, pages 6–8, Jun. 2000.
- [12] Woonghwan Ryu, A.L.C. Wai, Fan Wei, Wai Lai Lai, and Joungho Kim. Over ghz low-power rf clock distribution for a multiprocessor digital system. *Advanced Packaging*, *IEEE Transactions on*, 25:18–27, 2002.
- [13] Ashok Narasimhan, Shantanu Divekar, Praveen Elakkumanan, and Ramalingam Sridhar. A low-power current-mode clock distribution scheme for multi-ghz nocbased socs. VLSI Design, International Conference on, 0:130–133, 2005.
- [14] Q. Zhu and S. Tam. Package clock distribution design optimization for high-speed and low-power vlsis. *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging*, 20(1):56–63, 1997.
- [15] Jens Spars. Principles of asynchronous circuit design A systems perspective. Kluwer Academic Publishers, 2001.
- [16] M. Alioto, G. Palumbo, and M. Pennisi. Understanding the effect of process variations on the delay of static and domino logic. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 18(5):697–710, May 2010.
- [17] M. Figueiredo and R.L. Aguiar. Time precision comparison of digitally controlled delay elements. In *Circuits and Systems*, 2009. ISCAS 2009. IEEE International Symposium on, May 2009.

- [18] M. Figueiredo and R.L. Aguiar. Predicting noise and jitter in cmos inverters. In IEEE PhD. Research in Microelectronics and Electronics, 2007.
- [19] M. Figueiredo and Rui L. Aguiar. A dynamic jitter model to evaluate uncertainty trends with technology scaling. *Integration, the VLSI J.*, 45(2):162 – 171, 2012.
- [20] M. Figueiredo and R.L. Aguiar. Clock repeater characterization for jitter-aware clock tree synthesis. In José Monteiro and René van Leuken, editors, *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation,* volume 5953 of *Lecture Notes in Computer Science*, pages 46–55. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-11802-9_9.
- [21] M. Figueiredo and R.L. Aguiar. A jitter insertion and accumulation model for clock repeaters. submitted to IEICE Trans. on Fundamentals.
- [22] M. Figueiredo and R.L. Aguiar. Noise and jitter in cmos digitally controlled delay lines. In *IEEE Conf. on Electronics, Circuits and Systems*, 2006.
- [23] M. Figueiredo and R.L. Aguiar. Noise induced jitter performance of digitally controlled cmos delay lines. In *Conf. on Telecommunications*, 2007.
- [24] M. Figueiredo and R.L. Aguiar. A study on cmos time uncertainty with technology scaling. In Lars Svensson and José Monteiro, editors, *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation,* volume 5349 of *Lecture Notes in Computer Science,* pages 146–155. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-540-95948-9_15.
- [25] M. Figueiredo and R.L. Aguiar. Dynamic jitter accumulation in clock repeaters considering power and ground noise correlations. In *Circuits and Systems*, 2011. *ISCAS 2011. IEEE International Symposium on*, pages 2565 – 2568, 2011.
- [26] M. Figueiredo and R.L. Aguiar. Clock uncertainty model for deskewing schemes. to be submitted to PATMOS2012.
- [27] V.G. Oklobdzija, V.M. Stojanovic, D.M. Markovic, and N.M. Nedovic. Digital System Clocking: High-Performance and Low-Power Aspects. Wiley-IEEE Press, 2003.

- [28] D. Harris. Skew-tolerant circuit design. Morgan Kaufmann Publishers, 2001.
- [29] S. Tam. Clocking in Modern VLSI Systems, Chapter 2 Modern Clock Distribution Systems. Springer Science+Business Media, 2009.
- [30] P.K. Green. A ghz ia-32 architecture microprocessor implemented on 0.18 mu;m technology with aluminum interconnect. In *Solid-State Circuits Conference*, 2000. *Digest of Technical Papers*. *ISSCC*. 2000 IEEE International, pages 98–99, 449, 2000.
- [31] P.J. Restle, C.A. Carter, J.P. Eckhardt, B.L. Krauter, B.D. McCredie, K.A. Jenkins, A.J. Weger, and A.V. Mule. The clock distribution of the power4 microprocessor. In *Solid-State Circuits Conference*, 2002. *Digest of Technical Papers*. *ISSCC*. 2002 IEEE International, volume 2, pages 108–424, 2002.
- [32] J. Silberman, N. Aoki, D. Boerstler, J.L. Burns, Sang Dhong, A. Essbaum, U. Ghoshal, D. Heidel, P. Hofstee, Kyung Tek Lee, D. Meltzer, Hung Ngo, K. Nowka, S. Posluszny, O. Takahashi, I. Vo, and B. Zoric. A 1.0-ghz single-issue 64bit powerpc integer processor. *Solid-State Circuits, IEEE Journal of*, 33(11):1600–1608, Nov. 1998.
- [33] S. Tam, S. Rusu, U. Nagarji Desai, R. Kim, Ji Zhang, and I. Young. Clock generation and distribution for the first ia-64 microprocessor. *Solid-State Circuits, IEEE Journal* of, 35(11):1545–1552, Nov. 2000.
- [34] D.W. Boerstler. A low-jitter pll clock generator for microprocessors with lock range of 340-612 mhz. *Solid-State Circuits, IEEE Journal of*, 34(4):513 –519, Apr. 1999.
- [35] Z. Bobxing, G. Huimin, Z. Hong, and C. Tie. Design and optimization of an integrated 1ghz pll ip for microprocessors. In *Solid-State and Integrated Circuits Technol*ogy, 2004. Proceedings. 7th Int. Conf. on, volume 2, pages 1535 – 1538, Oct. 2004.
- [36] N.A. Kurd, J.S. Barkarullah, R.O. Dizon, T.D. Fletcher, and P.D. Madland. A multigigahertz clocking scheme for the pentium(r) 4 microprocessor. *Solid-State Circuits*, *IEEE Journal of*, 36(11):1647 –1653, Nov. 2001.

- [37] J. Tierno, A. Rylyakov, D. Friedman, A. Chen, A. Ciesla, T. Diemoz, G. English, D. Hui, K. Jenkins, P. Muench, G. Rao, G. Smith, M. Sperling, and K. Stawiasz. A dpll-based per core variable frequency clock generator for an eight-core power7 microprocessor. In VLSI Circuits IEEE Symp. on, pages 85–86, Jun. 2010.
- [38] R. Ho, K.W. Mai, and M.A. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4):490 –504, Apr. 2001.
- [39] H. Masuda, S. Okawa, and M. Aoki. Approach for physical design in sub-100 nm era. In *Circuits and Systems*, 2005. ISCAS 2005. IEEE International Symposium on, volume 6, pages 5934–5937, 23-26 2005.
- [40] X. Zhang and X. Bai. Emerging Technologies and Circuits, Process Variability-Induced Timing Failures - A Challenge in Nanometer CMOS Low-Power Design. Springer Science and Business Media, 2010.
- [41] S. S. Sapatnekar. Overcoming variations in nanometer-scale technologies. *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, 1(1):5–18, Mar. 2011.
- [42] K.A. Bowman, S.G. Duvall, and J.D. Meindl. Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. *Solid-State Circuits, IEEE Journal of*, 37(2):183–190, Feb. 2002.
- [43] Zhiyuan Li, Jianguo Ma, Yizheng Ye, and Mingyan Yu. Compact channel noise models for deep-submicron mosfets. *Electron Devices, IEEE Transactions on*, 56(6):1300–1308, Jun. 2009.
- [44] A. Hajimiri and T.H. Lee. A general theory of phase noise in electrical oscillators. *Solid-State Circuits, IEEE Journal of*, 33(2):179–194, Feb 1998.
- [45] J. Jeon, I. Song, I.M. Kang, Y. Yun, B.-G. Park, J.D. Lee, and H. Shin. A new noise parameter model of short-channel mosfets. In *Radio Frequency Integrated Circuits* (*RFIC*) Symposium, 2007 IEEE, pages 639–642, Jun. 2007.
- [46] A. van der Ziel. Thermal noise in field-effect transistors. *Proceedings of the IRE*, 50(8):1808–1812, Aug. 1962.

- [47] C. Hu and A. Niknejad. BSIM4.3.0 MOSFET Model. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 2003.
- [48] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variations and impact on circuits and microarchitecture. In *Design Automation Conference*, 2003. Proceedings, pages 338 – 342, Jun. 2003.
- [49] M. Hashimoto, T. Yamamoto, and H. Onodera. Statistical analysis of clock skew variation in h-tree structure. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E88-A(12):3375–3381, 2005.
- [50] A. Chakraborty, K. Duraisami, A. Sathanur, P. Sithambaram, L. Benini, A. Macii, E. Macii, and M. Poncino. Dynamic thermal clock skew compensation using tunable delay buffers. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(6):639–649, June 2008.
- [51] L.H. Chen, M. Marek-Sadowska, and F. Brewer. Buffer delay change in the presence of power and ground noise. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 11(3):461 – 473, June 2003.
- [52] Atsushi Muramatsu, Masanori Hashimoto, and Hidetoshi Onodera. Effects of onchip inductance on power distribution grid. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E88-A(12):3564–3572, 2005.
- [53] I. Kantorovich and C. Houghton. Effectiveness of on-die decoupling capacitance in improving chip performance. In *Electrical Performance of Electronic Packaging*, 2008 *IEEE-EPEP*, pages 165–168, Oct. 2008.
- [54] S. Bobba, T. Thorp, K. Aingaran, and D. Liu. Ic power distribution challenges. In Computer Aided Design, 2001. ICCAD 2001. IEEE/ACM International Conference on, pages 643 –650, 2001.
- [55] D.J. Herrell and B. Beker. Modeling of power distribution systems for highperformance microprocessors. *Advanced Packaging, IEEE Transactions on*, 22(3):240 –248, Aug. 1999.

- [56] A.V. Mezhiba and E.G. Friedman. Impedance characteristics of power distribution grids in nanoscale integrated circuits. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 12(11):1148 – 1155, Nov. 2004.
- [57] Jing Wang, D.M. Walker, Xiang Lu, A. Majhi, B. Kruseman, G. Gronthoud, L.E. Villagra, P.J.A. van de Wiel, and S. Eichenberger. Modeling power supply noise in delay testing. *Design Test of Computers, IEEE*, 24(3):226–234, May 2007.
- [58] T. Enami, S. Ninomiya, and M. Hashimoto. Statistical timing analysis considering spatially and temporally correlated dynamic power supply noise. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Trans. on*, 28(4):541–553, Apr. 2009.
- [59] V. Narang, B. Arya, and K. Rajagopal. Novel low delay slew rate control i/os. pages 189–193, Jul. 2009.
- [60] Li-Rong Zheng and H. Tenhunen. Interconnect-centric design for advanced SoC and NoC, Chapter 2 - Wires as Interconnects. Kluwer Academic Publishers, Boston, 2004.
- [61] P. Heydari and M. Pedram. Capacitive coupling noise in high-speed vlsi circuits. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 24(3):478 – 488, Mar. 2005.
- [62] X. Huang, P. Restle, T. Bucelot, Y. Cao, T-J. King, and C. Hu. Loop-based interconnect modeling and optimization approach for multigigahertz clock network design. *Solid-State Circuits, IEEE Journal of*, 38(3):457 – 463, Mar. 2003.
- [63] D.C. Sekar. Clock trees: differential or single ended? In *Quality of Electronic Design*, 2005. ISQED 2005. Sixth International Symposium on, pages 548 – 553, 21-23 2005.
- [64] N. Hedenstierna and K.O. Jeppson. Cmos circuit speed and buffer optimization. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 6(2):270 – 281, Mar. 1987.
- [65] T. Sakurai and A.R. Newton. Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas. *Solid-State Circuits, IEEE Journal of,* 25(2):584–594, Apr. 1990.

- [66] S. Dutta, S.S.M. Shetti, and S.L. Lusky. A comprehensive delay model for cmos inverters. *Solid-State Circuits*, *IEEE Journal of*, 30(8):864–871, Aug. 1995.
- [67] T. Sakurai and A.R. Newton. A simple mosfet model for circuit analysis. *Electron Devices, IEEE Transactions on*, 38(4):887–894, Apr. 1991.
- [68] M.M. Mansour and A. Mehrotra. Modified sakurai-newton current model and its applications to cmos digital circuit design. In VLSI, 2003. Proceedings. IEEE Computer Society Annual Symposium on, pages 62 – 69, 20-21 2003.
- [69] N. Chandra, A. Kumar Yati, and A.B. Bhattacharyya. Extended-sakurai-newton mosfet model for ultra-deep-submicrometer cmos digital design. In VLSI Design, 2009 22nd International Conference on, pages 247 –252, 5-9 2009.
- [70] Yangang Wang and M. Zwolinski. Analytical transient response and propagation delay model for nanoscale cmos inverter. In *Circuits and Systems*, 2009. ISCAS 2009. IEEE International Symposium on, pages 2998 –3001, 24-27 2009.
- [71] E. Yoshida, Y. Momiyama, M. Miyamoto, T. Saiki, M. Kojima, S. Satoh, and T. Sugii. Performance boost using a new device design methodology based on characteristic current for low-power cmos. In *Electron Devices Meeting*, 2006. *IEDM '06. International*, pages 1–4, 11-13 2006.
- [72] K.K. Ng, C.S. Rafferty, and Hong-Ih Cong. Effective on-current of mosfets for largesignal speed consideration. In *Electron Devices Meeting*, 2001. IEDM Technical Digest. *International*, pages 31.5.1–31.5.4, 2001.
- [73] M.H. Na, E.J. Nowak, W. Haensch, and J. Cai. The effective drive current in cmos inverters. In *International Electron Devices Meeting*, pages 121–124, 2002.
- [74] Xiaojun Yu, Shu jen Han, N. Zamdmer, Jie Deng, E.J. Nowak, and K. Rim. Improved effective switching current (ieff+) and capacitance methodology for cmos circuit performance prediction and model-to-hardware correlation. In *Electron Devices Meeting*, 2008. IEDM 2008. IEEE International, pages 1–4, 15-17 2008.

- [75] J. Hu, J.E. Park, G. Freeman, and H.S.P. Wong. Effective drive current in cmos inverters for sub-45nm technologies. In NSTI Nanotech, The Nanotechnology Conference and Trade Show, 2008.
- [76] K. von Arnim, C. Pacha, K. Hofmann, T. Schulz, K. Schriifer, and J. Berthold. An effective switching current methodology to predict the performance of complex digital circuits. In *Electron Devices Meeting*, 2007. *IEDM* 2007. *IEEE International*, pages 483–486, 10-12 2007.
- [77] A. Hirata, H. Onodera, and K. Tamaru. Analytical formulas of output waveform and short-circuit power dissipation for static cmos gates driving a crc pi load. *IEIC Trans. Fundamentals*, E00A:1–8, 1997.
- [78] Tianwen Tang. On-Chip Interconnect Noise in High-Performance CMOS Integrated Circuits. PhD thesis, University of Rochester, New York, 2000.
- [79] Sachin Sapatnekar. *Timing*. kluwer Academic Publishers, 2004.
- [80] W. C. Elmore. The transient response of damped linear networks with particular regard to wideband amplifiers. *Journal of Applied Physics*, 19(1):55–63, Jan. 1948.
- [81] C.L. Ratzlaff, S. Pullela, and L.T. Pillage. Modeling the rc-interconnect effects in a hierarchical timing analyzer. In *Custom Integrated Circuits Conference*, 1992., Proceedings of the IEEE 1992, pages 15.6.1–15.6.4, 3-6 1992.
- [82] P.R. O'Brien and T.L. Savarino. Modeling the driving-point characteristic of resistive interconnect for accurate delay estimation. In *Computer-Aided Design*, 1989. *ICCAD-89. Digest of Technical Papers.*, 1989 IEEE International Conference on, pages 512 –515, 5-9 1989.
- [83] L.T. Pillage and R.A. Rohrer. Asymptotic waveform evaluation for timing analysis. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 9(4):352 –366, Apr. 1990.
- [84] F. Dartu, N. Menezes, J. Qian, and L.T. Pillage. A gate-delay model for high-speed cmos circuits. In *Design Automation*, 1994. 31st Conf. on, pages 576–580, 6-10 1994.

- [85] J. Qian, S. Pullela, and L. Pillage. Modeling the effective capacitance for the rc interconnect of cmos gates. *Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on*, 13(12):1526 –1535, Dec. 1994.
- [86] Minglu Jiang, Qiang Li, Zhangcai Huang, and Y. Inoue. A non-iterative effective capacitance model for cmos gate delay computing. In *Communications, Circuits and Systems (ICCCAS), 2010 International Conference on*, pages 896–900, july 2010.
- [87] A. Nardi, E. Tuncer, S. Naidu, A. Antonau, S. Gradinaru, T. Lin, and J. Song. Use of statistical timing analysis on real designs. In *Design, Automation Test in Europe Conference Exhibition*, 2007. DATE '07, pages 1–6, 16-20 2007.
- [88] Todd Charles Weigandt. Low-phase-noise, low-timing-jitter design techniques for delay cell based VCOs and frequency synthesizers. PhD thesis, University of California, Berkeley, 1998.
- [89] J. McNeill. Jitter in Ring Oscillators. PhD thesis, Boston University, 1994.
- [90] B. Razavi. A study of phase noise in cmos oscillators. Solid-State Circuits, IEEE Journal of, 31(3):331–343, Mar. 1996.
- [91] A. Demir, A. Mehrotra, and J. Roychowdhury. Phase noise in oscillators: a unifying theory and numerical methods for characterization. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 47(5):655–674, May 2000.
- [92] A.A. Abidi and R.G. Meyer. Noise in relaxation oscillators. *Solid-State Circuits, IEEE Journal of*, 18(6):794–802, Dec. 1983.
- [93] T.C. Weigandt, Beomsup Kim, and P.R. Gray. Analysis of timing jitter in cmos ring oscillators. In *Circuits and Systems*, 1994. ISCAS '94., 1994 IEEE International Symposium on, volume 4, pages 27–30 vol.4, 30 1994.
- [94] M. Saint-Laurent and M. Swaminathan. Impact of power-supply noise on timing in high-frequency microprocessors. *Advanced Packaging*, *IEEE Transactions on*, 27(1):135 – 144, Feb. 2004.

- [95] J.V.R. Ravindra and M.B. Srinivas. Analytical crosstalk model with inductive coupling in vlsi interconnects. In *Signal Propagation on Interconnects*, 2007. SPI 2007. IEEE Workshop on, 2007.
- [96] Pinhong Chen, D.A. Kirkpatrick, and K. Keutzer. Miller factor for gate-level coupling delay calculation. In *Computer Aided Design*, 2000. ICCAD-2000. IEEE/ACM International Conference on, pages 68–74, 2000.
- [97] Kevin T. Tang and Eby G. Friedman. Delay and noise estimation of cmos logic gates driving coupled resistive-capacitive interconnections. *Integration, the VLSI Journal*, 29:131–165, Sep. 2000.
- [98] Yungseon Eo, Seongkyun Shin, W.R. Eisenstadt, and Jongin Shim. A decoupling technique for efficient timing analysis of vlsi interconnects with dynamic circuit switching. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 23(9):1321 – 1337, Sept. 2004.
- [99] K. Takeuchi, K. Yanagisawa, T. Sato, K. Sakamoto, and S. Hojo. Probabilistic crosstalk delay estimation for asics. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 23(9):1377 – 1383, Sept. 2004.
- [100] A. Hajimiri, S. Limotyrakis, and T.H. Lee. Jitter and phase noise in ring oscillators. Solid-State Circuits, IEEE Journal of, 34(6):790 –804, Jun. 1999.
- [101] T. Pialis and K. Phang. Analysis of timing jitter in ring oscillators due to power supply noise. In *Circuits and Systems*, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on, volume 1, pages I–685 – I–688 vol.1, 25-28 2003.
- [102] M.-J.E. Lee, W.J. Dally, T. Greer, Hiok-Tiaq Ng, R. Farjad-Rad, J. Poulton, and R. Senthinathan. Jitter transfer characteristics of delay-locked loops - theories and design techniques. *Solid-State Circuits, IEEE Journal of*, 38(4):614 – 621, Apr. 2003.
- [103] A. Tajalli, P. Muller, M. Atarodi, and Y. Leblebici. Analysis and modeling of jitter and frequency tolerance in gated oscillator based cdrs. In *Circuits and Systems*, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on, 2006.

- [104] K.L. Wong, T. Rahal-Arabi, M. Ma, and G. Taylor. Enhancing microprocessor immunity to power supply noise with clock-data compensation. *Solid-State Circuits*, *IEEE Journal of*, 41(4):749 – 758, Apr. 2006.
- [105] J. Jang, O. Franza, and W. Burleson. Period jitter estimation in global clock trees. In Signal Propagation on Interconnects, 2008. SPI 2008. 12th IEEE Workshop on, pages 1-4, 12-15 2008.
- [106] L.W. Nagel and D.O. Pederson. Spice (simulation program with integrated circuit emphasis), memorandum no. erl-m382. Technical report, University of California, Berkeley, 1973.
- [107] C. Visweswariah. Death, taxes and failing chips. In *Design Automation Conference*, 2003. Proceedings, pages 343 – 347, 2-6 2003.
- [108] A. Singhee, S. Singhal, and R.A. Rutenbar. Practical, fast monte carlo statistical static timing analysis: Why and how. In *Computer-Aided Design*, 2008. ICCAD 2008. IEEE/ACM International Conference on, pages 190–195, 10-13 2008.
- [109] H.-F. Jyu, S. Malik, S. Devadas, and K.W. Keutzer. Statistical timing analysis of combinational logic circuits. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 1(2):126–137, Jun 1993.
- [110] M. Gao, Z. Ye, Y. Peng, Y. Wang, and Z. Yu. A comprehensive model for gate delay under process variation and different driving and loading conditions. In *Quality Electronic Design (ISQED), 2010 11th Int. Symp. on,* pages 406–412, 22-24 2010.
- [111] A. Mutlu, Jiayong Le, R. Molina, and M. Celik. A parametric approach for handling local variation effects in timing analysis. In *Design Automation Conference*, 2009. DAC '09. 46th ACM/IEEE, pages 126–129, July 2009.
- [112] Sunil Walia. Primetime advanced ocv technology. Technical report, Synopsys, 2009.
- [113] R.C.H. van de Beek, E.A.M. Klumperink, C.S. Vaucher, and B. Nauta. Low-jitter clock multiplication: a comparison between plls and dlls. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 49(8):555 – 566, Aug. 2002.

- [114] B. Kim, T.C. Weigandt, and P.R. Gray. Pll/dll system noise analysis for low jitter clock synthesizer design. In *Circuits and Systems*, 1994. ISCAS '94., 1994 IEEE International Symposium on, volume 4, pages 31–34, 1994.
- [115] N. Bindal, T. Kelly, N. Velastegui, and K.L. Wong. Scalable sub-10ps skew global clock distribution for a 90nm multi-ghz ia microprocessor. In *Solid-State Circuits Conference*, 2003. *Digest of Technical Papers*. *ISSCC*. 2003 IEEE International, volume 1, pages 346 – 498, 2003.
- [116] D.W. Bailey and B.J. Benschneider. Clocking design and analysis for a 600-mhz alpha microprocessor. *Solid-State Circuits, IEEE J. of*, 33(11):1627 –1633, Nov. 1998.
- [117] Gustavo Reis Wilke. Analysis and Optimization of Mesh-based Clock Distribution Architectures. PhD thesis, Universidade Federal do Rio Grande do Sul, 2008.
- [118] F.E. Anderson, J.S. Wells, and E.Z. Berta. The core clock system on the nextgeneration ltaniumlm microprocessor. In *Solid-State Circuits Conference*, 2002. *Digest* of *Technical Papers*. *ISSCC*. 2002 *IEEE International*, volume 2, pages 110–424, 2002.
- [119] S. Tam, U. Desai, and R. Limaye. Clock generation and distribution for the third generation itanium processor. In VLSI Circuits, 2003. Digest of Technical Papers. 2003 Symposium on, pages 9 – 12, 12-14 2003.
- [120] J. Clabes, J. Friedrich, M. Sweet, J. Dilullo, S. Chu, D. Plass, J. Dawson, P. Muench, L. Powell, M. Floyd, B. Sinharoy, M. Lee, M. Goulet, J. Wagoner, N. Schwartz, S. Runyon, G. Gorman, P. Restle, R. Kalla, J. McGill, and S. Dodson. Design and implementation of the power5 microprocessor. In *Solid-State Circuits Conference, IEEE International*, volume 1, pages 56 57, 15-19 2004.
- [121] E. Fayneh and E. Knoll. Clock generation and distribution for intel banias mobile microprocessor. In VLSI Circuits, 2003. Digest of Technical Papers. 2003 Symposium on, pages 17 – 20, 12-14 2003.
- [122] P. Mahoney, E. Fetzer, B. Doyle, and S. Naffziger. Clock distribution on a dualcore, multi-threaded itanium reg;-family processor. In *Solid-State Circuits Confer-*

ence, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International, pages 292 –599 Vol. 1, 10-10 2005.

- [123] S. Tam, J. Leung, R. Limaye, S. Choy, S. Vora, and M. Adachi. Clock generation and distribution of a dual-core xeon processor with 16mb l3 cache. In *Solid-State Circuits Conference*, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International, pages 1512 –1521, 6-9 2006.
- [124] J. Dorsey, S. Searles, M. Ciraula, S. Johnson, N. Bujanos, D. Wu, M. Braganza, S. Meyers, E. Fang, and R. Kumar. An integrated quad-core opteron processor. In *Solid-State Circuits Conference*, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International, pages 102–103, 11-15 2007.
- [125] J. Friedrich, B. McCredie, N. James, B. Huott, B. Curran, E. Fluhr, G. Mittal, E. Chan, Y. Chan, D. Plass, Sam Chu, Hung Le, L. Clark, J. Ripley, S. Taylor, J. Dilullo, and M. Lanzerotti. Design of the power6 microprocessor. In *Solid-State Circuits Conference, IEEE International*, pages 96–97, 11-15 2007.
- [126] N. Sakran, M. Yuffe, M. Mehalel, J. Doweck, E. Knoll, and A. Kovacs. The implementation of the 65nm dual-core 64b merom processor. In *Solid-State Circuits Conference*, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International, pages 106–590, 11-15 2007.
- [127] A. Allen, J. Desai, F. Verdico, F. Anderson, D. Mulvihill, and D. Krueger. Dynamic frequency-switching clock system on a quad-core itanium processor. In *Solid-State Circuits Conference - Digest of Technical Papers*, 2009. ISSCC 2009. IEEE International, pages 62 –63,63a, 8-12 2009.
- [128] N. Kurd, P. Mosalikanti, M. Neidengard, J. Douglas, and R. Kumar. Next generation intel core micro-architecture (nehalem) clocking. In *Solid-State Circuits, IEEE Journal of*, volume 44, pages 1121–1129, Apr. 2009.
- [129] Simon Tam, Justin Leung, and Rahul Limaye. Clock generation and distribution for a 45nm, 8-core xeon processor with 24mb cache. In VLSI Circuits, 2009 Symposium on, pages 154–155, 16-18 2009.

- [130] N.A. Kurd, S. Bhamidipati, C. Mozak, J.L. Miller, P. Mosalikanti, T.M. Wilson, A.M. El-Husseini, M. Neidengard, R.E. Aly, M. Nemani, M. Chowdhury, and R. Kumar. A family of 32 nm ia processors. *Solid-State Circuits, IEEE Journal of*, 46(1):119–130, Jan. 2011.
- [131] Y. Elboim, A. Kolodny, and R. Ginosar. A clock-tuning circuit for system-on-chip. Very Large Scale Integration (VLSI) Systems, IEEE Trans. on, 11(4):616–626, Aug. 2003.
- [132] Jeng-Liang Tsai, Lizheng Zhang, and Charlie Chung-Ping Chen. Statistical timing analysis driven post-silicon-tunable clock-tree synthesis. In *Computer-Aided Design*, *IEEE/ACM International Conference on*, pages 575 – 581, Nov. 2005.
- [133] V. Khandelwal and A. Srivastava. Variability-driven formulation for simultaneous gate sizing and postsilicon tunability allocation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(4):610–620, Apr. 2008.
- [134] J. G. Mueller and R. A. Saleh. Autonomous, multilevel ring tuning scheme for postsilicon active clock deskewing over intra-die variations. *Very Large Scale Integration* (VLSI) Systems, IEEE Transactions on, PP(99):1–14, 2010.
- [135] F. Anceau. A synchronous approach for clocking vlsi systems. Solid-State Circuits, IEEE Journal of, 17(1):51 – 56, Feb. 1982.
- [136] E.G. Friedman and S. Powell. Design and analysis of a hierarchical clock distribution system for synchronous standard cell/macrocell vlsi. *Solid-State Circuits, IEEE Journal of*, 21(2):240 – 246, Apr. 1986.
- [137] I.A. Young, J.K. Greason, and K.L. Wong. A pll clock generator with 5 to 110 mhz of lock range for microprocessors. *Solid-State Circuits, IEEE Journal of*, 27(11):1599 –1607, Nov. 1992.
- [138] M.G. Johnson and E.L. Hudson. A variable delay line pll for cpu-coprocessor synchronization. *Solid-State Circuits, IEEE Journal of*, 23(5):1218–1223, Oct. 1988.

- [139] A. Kapoor, N. Jayakumar, and S.P. Khatri. Dynamically de-skewable clock distribution methodology. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions* on, 16(9):1220 –1229, Sept. 2008.
- [140] A. Chattopadhyay and Z. Zilic. Galds: a complete framework for designing multiclock asics and socs. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions* on, 13(6):641 – 654, Jun. 2005.
- [141] S.R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh,
 T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar.
 An 80-tile sub-100-w teraflops processor in 65-nm cmos. *Solid-State Circuits, IEEE Journal of*, 43(1):29 –41, Jan. 2008.
- [142] Jr. Watson, R.B. and R.B. Iknaian. Clock buffer chip with multiple target automatic skew compensation. *Solid-State Circuits*, *IEEE J. of*, 30(11):1267–1276, Nov. 1995.
- [143] S.-K. Kao, B.-J. Chen, and S.-I. Liu. A 62.5625-mhz anti-reset all-digital delay-locked loop. *Circuits and Systems II: Express Briefs, IEEE Trans. on*, 54(7):566–570, Jul. 2007.
- [144] Ching-Che Chung and Chen-Yi Lee. An all-digital phase-locked loop for highspeed clock generation. *Solid-State Circuits, IEEE J. of,* 38(2):347 – 351, Feb. 2003.
- [145] Ching-Che Chung and Chen-Yi Lee. A new dll-based approach for all-digital multiphase clock generation. *Solid-State Circuits, IEEE J. of*, 39(3):469 – 475, Mar. 2004.
- [146] Duo Sheng, Ching-Che Chung, and Chen-Yi Lee. An ultra-low-power and portable digitally controlled oscillator for soc applications. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 54(11):954–958, Nov. 2007.
- [147] T. Matano, Y. Takai, T. Takahashi, Y. Sakito, I. Fujii, Y. Takaishi, H. Fujisawa, S. Kubouchi, S. Narui, K. Arai, M. Morino, M. Nakamura, S. Miyatake, T. Sekiguchi, and K. Koyama. A 1-gb/s/pin 512-mb ddrii sdram using a digital dll and a slewrate-controlled output buffer. *Solid-State Circuits, IEEE Journal of*, 38(5):762–768, May 2003.

- [148] Guang-Kaai Dehng, June-Ming Hsu, Ching-Yuan Yang, and Shen-Iuan Liu. Clockdeskew buffer using a sar-controlled delay-locked loop. *Solid-State Circuits, IEEE Journal of*, 35(8):1128 –1136, Aug. 2000.
- [149] J. Kim, D.G. Kam, P.J. Jun, and J. Kim. Spread spectrum clock generator with delay cell array to reduce electromagnetic interference. *Electromagnetic Compatibility, IEEE Transactions on*, 47(4):908 – 920, Nov. 2005.
- [150] N.R. Mahapatra, S.V. Garimella, and A. Tareen. An empirical and analytical comparison of delay elements and a new delay element design. pages 81 –86, 2000.
- [151] I. Sutherland, R.F. Sproull, and D. Harris. Logical Effort: Designing Fast CMOS Circuits. Morgan Kaufmann., 1999.
- [152] J. Mueller and R. Saleh. A tunable clock buffer for intra-die pvt compensation in single-edge clock (sec) distribution networks. pages 572 –577, Mar. 2008.
- [153] M. Saint-Laurent and M. Swaminathan. A digitally adjustable resistor for path delay characterization in high-frequency microprocessors. pages 61–64, 2001.
- [154] D.K. Jeong, G. Borriello, D.A. Hodges, and R.H. Katz. Design of pll-based clock generation circuits. *Solid-State Circuits*, *IEEE Journal of*, 22(2):255 – 261, Apr. 1987.
- [155] M. Bazes. A novel precision mos synchronous delay line. *Solid-State Circuits, IEEE Journal of*, 20(6):1265 1271, Dec. 1985.
- [156] M. Maymandi-Nejad and M. Sachdev. A digitally programmable delay element: design and analysis. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 11(5):871 – 878, Oct. 2003.
- [157] G. Geannopoulos and X. Dai. An adaptive digital deskewing circuit for clock distribution networks. pages 400–401, Feb. 1998.
- [158] P. Andreani, F. Bigongiari, R. Roncella, R. Saletti, and P. Terreni. A digitally controlled shunt capacitor cmos delay line. *Analog Integrated Circuits and Signal Processing*, 18:89–96, 1999. 10.1023/A:1008359721539.

- [159] Wolfgang Maichen. *Digital Timing Measurements, From Scopes and Probes to Timing and Jitter*. Springer Netherlands, 2006.
- [160] K. Minami, M. Mizuno, H. Yamaguchi, T. Nakano, Y. Matsushima, Y. Sumi, T. Sato, H. Yamashida, and M. Yamashina. A 1 ghz portable digital delay-locked loop with infinite phase capture ranges. In *Solid-State Circuits Conference*, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International, pages 350–351, 469, 2000.
- [161] P. Maurine, M. Rezzoug, N. Azemard, and D. Auvergne. Transition time modeling in deep submicron cmos. *Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on*, 21(11):1352 – 1363, Nov. 2002.
- [162] M. Maymandi-Nejad and M. Sachdev. A monotonic digitally controlled delay element. *IEEE Journal of Solid-State Circuits*, 40 (11):2212–2219, 2005.
- [163] C.J. Akl and M.A. Bayoumi. Reducing interconnect delay uncertainty via hybrid polarity repeater insertion. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(9):1230–1239, Sep. 2008.
- [164] A. Kurokawa, T. Sato, T. Kanamoto, and M. Hashimoto. Interconnect modeling: A physical design perspective. *Electron Devices*, *IEEE Transactions on*, 56(9):1840–1851, Sept. 2009.
- [165] Y. Okajima, M. Taguchi, M. Yanagawa, K. Nishimura, and O. Hamada. Digital delay locked loop and design technique for high-speed synchronous interface. *IEICE Trans. Electron.*, E79-C, 1996.
- [166] Feng Lin, J. Miller, A. Schoenfeld, M. Ma, and R.J. Baker. A register-controlled symmetrical dll for double-data-rate dram. *Solid-State Circuits, IEEE Journal of*, 34(4):565 –568, Apr. 1999.
- [167] R.-J. Yang and S.-I. Liu. A 40-550 mhz harmonic-free all-digital delay-locked loop using a variable sar algorithm. *Solid-State Circuits, IEEE Journal of*, 42(2):361–373, Feb. 2007.

- [168] Pao-Lung Chen, Ching-Che Chung, Jyh-Neng Yang, and Chen-Yi Lee. A clock generator with cascaded dynamic frequency counting loops for wide multiplication range applications. *Solid-State Circuits*, *IEEE J. of*, 41(6):1275 – 1285, Jun. 2006.
- [169] Kwang-Hee Choi, Jung-Bum Shin, Jae-Yoon Sim, and Hong-June Park. An interpolating digitally controlled oscillator for a wide-range all-digital pll. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 56(9):2055 –2063, Sept. 2009.
- [170] B.W. Garlepp, K.S. Donnelly, Jun Kim, P.S. Chau, J.L. Zerbe, C. Huang, C.V. Tran, C.L. Portmann, D. Stark, Yiu-Fai Chan, T.H. Lee, and M.A. Horowitz. A portable digital dll for high-speed cmos interface circuits. *Solid-State Circuits, IEEE Journal of*, 34(5):632 –644, May 1999.
- [171] Byoung-Mo Moon, Young-June Park, and Deog-Kyoon Jeong. Monotonic widerange digitally controlled oscillator compensated for supply voltage variation. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 55(10):1036–1040, Oct. 2008.
- [172] M.-J. Kim and L.-S. Kim. 100mhz to 1ghz open-loop addll with fast lock-time for mobile applications. In *Custom Integrated Circuits Conference (CICC)*, 2010 IEEE, pages 1–4, Sept. 2010.
- [173] H.B. Bakoglu. *Circuits, interconnections, and packaging for VLSI*. Addison-Wesley Pub. Co., 1990.
- [174] A. Chandrakasan, W. J. Bowhill, and F. Fox. Design of High-Performance Microprocessor Circuits. IEEE Press, 2001.
- [175] E. Alon, V. Abramzon, B. Nezamfar, and M. Horowitz. On-die power supply noise measurement techniques. *Advanced Packaging, IEEE Transactions on*, 32(2):248–259, May 2009.
- [176] M. Graziano and G. Piccinini. Statistical power supply dynamic noise prediction in hierarchical power grid and package networks. *Integration, the VLSI Journal*, 41:524–538, Jul. 2008.

- [177] I. Chanodia and D. Velenis. Parameter variations and crosstalk noise effects on high performance h-tree clock distribution networks. *Analog Integrated Circuits and Signal Processing, Special issue: Selected Papers on MWSCAS 2005*, 56:13–21, 2008.
- [178] I. Kantorovich and C. Houghton. Maximum tolerable power supply noise for dataclock synchronization. In *Electrical Performance of Electronic Packaging*, 2006 IEEE, pages 167–170, Oct. 2006.
- [179] P.J. Restle and A. Deutsch. Designing the best clock distribution network. In VLSI Circuits, 1998. Digest of Technical Papers. 1998 Symposium on, pages 2 –5, Jun. 1998.
- [180] N. MohammadZadeh, M. Mirsaeedi, A. Jahanian, and M.S. Zamani. Multi-domain clock skew scheduling-aware register placement to optimize clock distribution network. In *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.,* pages 833–838, Apr. 2009.
- [181] Chopra K. Blaauw D. Agarwal, A. and V. Zolotov. Circuit optimization using statistical static timing analysis. In *Design Automation Conference*, 2005. Proceedings. 42nd, pages 321 – 324, Jun. 2005.
- [182] T. Xanthopoulos, D.W. Bailey, A.K. Gangwar, M.K. Gowan, A.K. Jain, and B.K. Prewitt. The design and analysis of the clock distribution network for a 1.2 ghz alpha microprocessor. In *Solid-State Circuits Conference*, 2001. Digest of Technical Papers. ISSCC. 2001 IEEE International, pages 402–403, 2001.
- [183] D.E. Brueske and S.H.K. Embabi. A dynamic clock synchronization technique for large systems. Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, IEEE Transactions on, 17(3):350–361, Aug. 1994.
- [184] H. Sutoh, K. Yamakoshi, and M. Ino. Circuit technique for skew-free clock distribution. In *Custom Integrated Circuits Conference*, 1995., Proceedings of the IEEE 1995, pages 163–166, may 1995.

- [185] Hyun Lee, Han Quang Nguyen, and D.W. Potter. Design self-synchronized clock distribution networks in an soc asic using dll with remote clock feedback. In ASIC/SOC Conf., 2000. Proceedings. 13th Annual IEEE Int., pages 248 –252, 2000.
- [186] R.L. Aguiar and D.M. Santos. Wide-area clock distribution using controlled delay lines. In *Electronics, Circuits and Systems, 1998 IEEE International Conference on*, volume 2, pages 63–66, 1998.
- [187] W.D. Grover. A new method for clock distribution. Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, 41(2):149–160, Feb. 1994.
- [188] A. Shibayama, M. Mizuno, H. Abiko, A. Ono, S. Masuoka, A. Matsumoto, T. Tamura, Y. Yamada, A. Nishizawa, H. Kawamoto, K. Inoue, Y. Nakazawa, I. Sakai, and M. Yamashina. Device-deviation tolerant over-1 ghz clock distribution scheme with skew-immune race-free impulse latch circuits. In *Solid-State Circuits Conf.*, 1998. Digest of Technical Papers. 1998 IEEE Int., pages 402 –403, 473, Feb. 1998.
- [189] D.R. Rolston, D.M. Gross, G.W. Roberts, and D.V. Plant. A distributed synchronized clocking method. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 52(8):1597 – 1607, Aug. 2005.
- [190] C.E. Dike, N.A. Kurd, P. Patra, and J. Barkatullah. A design for digital, dynamic clock deskew. In VLSI Circuits, 2003. Digest of Technical Papers. 2003 Symposium on, pages 21–24, Jun. 2003.
- [191] R.L. Aguiar and D.M. Santos. Highly efficient multi-point clock distribution networks. In *Electronics, Circuits and Systems, 1998 IEEE Int. Conf. on*, 2000.
- [192] Texas Instruments. Little logic guide. Technical report, available online in "http://focus.ti.com/lit/sg/scyt129c/scyt129c.pdf", 2009.
- [193] Agilent Technologies. 81100 family of pulse pattern generators. Technical report, available online in "http://cp.literature.agilent.com/litweb/pdf/5980-1215E.pdf", 2008.

- [194] Tecktronics. Digital and mixed signal oscilloscopes, dpo/ dsa/ mso70000 series data sheet. Technical report, available online, 2010.
- [195] PTM. Predictive technology models for high-performance. Technical report, http://ptm.asu.edu/, 2008.
- [196] G.E. Moore. Progress in digital integrated electronics. In *Electron Devices Meeting*, 1975 International, volume 21, pages 11 – 13, 1975.
- [197] C. Mead. Fundamental limitations in microelectronicsi. mos technology. Solid-state Electronics, 15:819–829, 1972.
- [198] R.H. Dennard, F.H. Gaensslen, Hwa-Nien Yu, V.L. Rideout, E. Bassous, and A.R. Leblanc. Design of ion-implanted mosfet's with very small physical dimensions. *Proceedings of the IEEE*, 87(4):668–678, Apr. 1999.
- [199] G. Baccarani, M.R. Wordeman, and R.H. Dennard. Generalized scaling theory and its application to a 0.25 micrometer mosfet design. *IEEE Trans. on Electron Devices*, 31:452–462, 1984.
- [200] Yu Cao Greg W. Starr Ban Wong, Anurag Mittal. Nano-CMOS Circuit and Physical Design. Wiley - IEEE Press, 2004.
- [201] A. Khakifirooz and D.A. Antoniadis. Mosfet performance scalingpart ii: Future directions. *IEEE Trans. on Electron Devices*, 55:1401–1408, 2008.
- [202] A.V. Mezhiba and E.G. Friedman. Scaling trends of on-chip power distribution noise. Very Large Scale Integration (VLSI) Systems, IEEE Trans. on, 12:386–394, 2004.
- [203] Paul Peter P. Sotiriadis. Interconnect Modeling and Optimization in Deep Sub-Micron Technologies. PhD thesis, Massachusetts Institute of Technology, 2002.
- [204] M.J. Flynn and P. Hung. Microprocessor design issues: thoughts on the road ahead. Micro, IEEE, 25(3):16 – 31, May 2005.
- [205] E.S. Fetzer. Using adaptive circuits to mitigate process variations in a microprocessor design. *Design Test of Computers, IEEE*, 23(6):476–483, Jun. 2006.

- [206] R.A. Walker and D.E. Thomas. A model of design representation and synthesis. In Design Automation, 1985. 22nd Conference on, pages 453 – 459, June 1985.
- [207] R. Reis. Physical design automation at transistor level. In NORCHIP, 2008., pages 241–245, Nov. 2008.
- [208] Kuan-Neng Chen, M.J. Kobrinsky, B.C. Barnett, and R. Reif. Comparisons of conventional, 3-d, optical, and rf interconnects for on-chip clock distribution. *Electron Devices, IEEE Transactions on*, 51(2):233 – 239, Feb. 2004.
- [209] L. Benini and G. De Micheli. Networks on chips: a new soc paradigm. *Computer*, 35(1):70–78, Jan 2002.
- [210] U.Y. Ogras, P. Bogdan, and R. Marculescu. An analytical approach for network-onchip performance analysis. *Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on*, 29(12):2001–2013, Dec. 2010.
- [211] T. Sterling and M. Brodowicz. Continuum computer architecture for nano-scale and ultra-high clock rate technologies. In *Innovative Architecture for Future Generation High-Performance Processors and Systems*, 2005, page 9 pp., Jan. 2005.
- [212] J. Nemeth, Rui Min, Wen-Ben Jone, and Yiming Hu. Location cache design and performance analysis for chip multiprocessors. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 19(1):104 –117, Jan. 2011.
- [213] M.D. Hill and M.R. Marty. Amdahl's law in the multicore era. *Computer*, 41(7):33 –38, Jul. 2008.
- [214] S. Dighe, S.R. Vangal, P. Aseron, S. Kumar, T. Jacob, K.A. Bowman, J. Howard, J. Tschanz, V. Erraguntla, N. Borkar, V.K. De, and S. Borkar. Within-die variationaware dynamic-voltage-frequency-scaling with optimal core allocation and thread hopping for the 80-core teraflops processor. *Solid-State Circuits, IEEE Journal of*, 46(1):184–193, Jan. 2011.