

Gradual Synchronization

Sandra Jackson and Rajit Manohar

Cornell University



WHY SYNCHRONIZERS?









THE FIRST SYNCHRONIZERS



- Circuits that detect metastability, and pause or stretch the clock until it is resolved
 - Pros: No chance of a metastability error
 - Cons: Unbounded Latency
- Predictive synchronizers
- Synchronizers that exploit known relationships





- Rate of Entering Metastability:
 R = F_DF_CT_w
- If F_D = 50MHz, F_C = 1GHz, T_w = 50ps, then metastability is encountered 2,500,000 times per second!



SYNCHRONIZERS IN SYSTEMS (MOTIVATION)

Flip-Flop
 Synchronizers





Cornell University Computer Systems Laboratory CS

SYNCHRONIZERS IN SYSTEMS (MOTIVATION)



FIFO based synchronizers





GRADUAL SYNCHRONIZATION







4-PHASE ATOS GRADUAL SYNCHRONIZER





FIFO BLOCK







4-PHASE STOA GRADUAL SYNCHRONIZER





FIFO BLOCK



CSI



PROOF CONCEPTS





 $P_f^{(i+1)} \le P_f^{(i+1)}(R_i) +$ $P_f^{(i+1)}(A_o) + P_f^{(i+1)}(S_i)$











































4-PHASE REQUIREMENTS



T/2

T/2

T/2

T/2

T/2

T

T

T

T

T

Synchronous to Async	Asynchronous to Synchronous				
$ au_S + au_{A_oA_i}$	<	T/2	$ au_S + au_{R_iR_o}$	<	
$ au_{\mathrm{D}}$, $\pm au_{\mathrm{I}}$	_	T/2	$ au_{A_oR_o}$	<	
$K_i A_i + K_d$		1 / 2	$ au_{AR}$	<	
$ au_{RA}$	<	T/2	$ au_{A_oA_i}$	<	
$\tau_{R:R} + \tau_d$	<	T/2	$ au_d + au_{S_i R_o}$	<	
$K_l K_0 $ u		,	$\tau_S + \tau_{R_iA_i} + \tau_{A_oR_o}$	<	
$\tau_S + \tau_{A_o R_o} + \tau_{R_i A_i} + \tau_d$	<	Т	$ au_S + au_{R_iA_i} + au_{AR}$	<	
$ au_S + au_{A_0R_0} + au_{RA}$	<	Т	$ au_{RA}+ au_{A_oR_o}$	<	
0 0			$\tau_d + \tau_{S_iA_i} + \tau_{A_oR_o}$	<	
$ au_{AR} + au_{R_iA_i} + au_d$	<	Τ,	$ au_d + au_{S_iR_i} + au_{AR}$	<	



2-PHASE GRADUAL SYNCHRONIZER







SIMULATIONS



- HSIM, 90nm process, 1.2 V
- Two Synchronous Environments
- Synchronizer Types
 - Simple Four-Phase Flip-Flop
 - Fast Four-Phase Flip-Flop
 - Fast Two-Phase Flip-Flop
 - Dual Clock FIFO
 - 2-Phase and 4-Phase Pipeline Synchronizer
 - 2-Phase and 4-Phase Gradual Synchronizer



LATENCY







THROUGHPUT







MTBF IS IMPORTANT







DETERMINING COMPUTATION

Two Factors:

- Remember those requirements, which one leads to the smallest tau_d
- Re-locatable computation available





POTENTIAL CYCLES TO MERGE







SYSTEM LATENCY ESTIMATE







GSYNC IN NOC







FLIP-FLOP SYNCHRONIZER SEND NI



CS



GRADUAL SYNCHRONIZER SEND NI







FLIP-FLOP SYNCHRONIZER RECEIVE NI



CS



GRADUAL SYNCHRONIZER RECEIVE NI







SEND INTERFACE SIMULATION RESULTS



	TX Clock	Network	MTBF	Latency (ns)		Throughput			
Sync Type	(MHz)	(MHz)	(years)	Header		Body or Tail		(ppc)	
				min	max	min	max	min	max
Fast 4-Phase	400	272	$1.84x10^{40}$	5.897	8.454	2.473	5.027	0.33	
	600	397	$4.12x10^{19}$	4.221	5.906	1.632	3.313	0.33	
	800	531	$2.05x10^9$	4.732	8.475	2.525	6.277	0.33	
Gradual	400	400	$2.04x10^{51}$	2.625	2.656	1.549	1.586	1	
	600	600	$6.20x10^{20}$	2.839	6.654	1.759	2.254	0.78	1
	800	800	2.47×10^{12}	2.997	7.971	1.973	4.836	0.63	0.78



RECEIVE INTERFACE SIMULATION RESULTS



	RX Clock	Network	MTBF	BF Latency (ns)		Throughput	
Sync Type	(MHz)	(MHz)	(years)	min	max	(fpc)	
	400	271	1.14×10^{40}	5.12	7.01	0.33	
Fast 4-Phase	600	395	3.93×10^{19}	4.51	5.096	0.33	
	800	527	1.95×10^9	3.26	3.64	0.33	
	400	400	3.98×10^{51}	2.56	6.53	1	
Gradual	600	600	1.22×10^{21}	2.54	3.31	1	
	800	800	$6.02x10^{12}$	2.65	3.06	1	



SUMMARY



Gradual Synchronization

- Computation ready FIFO
- Low latency
- Maximum throughput

Application (NI)

- Merge realistic computation
- Full system latency reduction





Gradual Synchronization

Sandra Jackson and Rajit Manohar

Cornell University

