19.01.2016

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

MOUSETRAP
Die asynchrone Pipeline-Logik

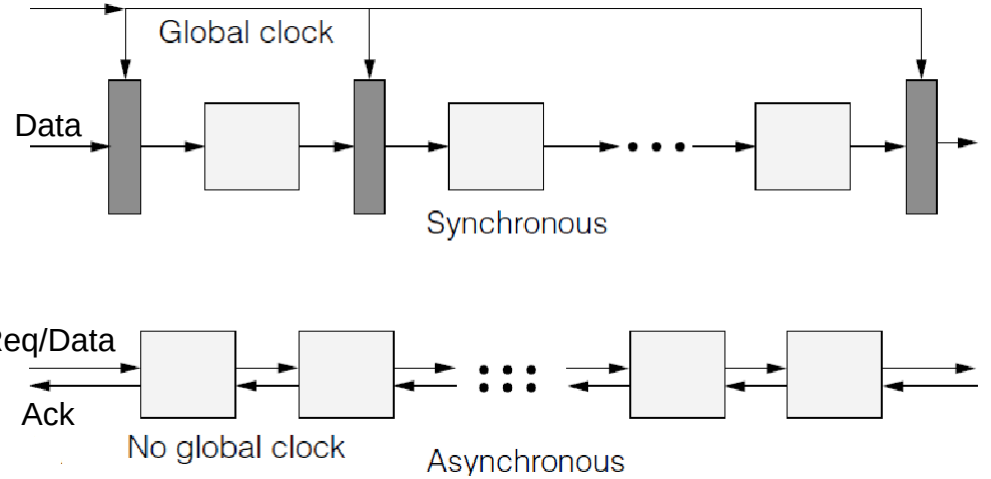
Minimal-Overhead Ultrahigh-SpEed Transition-signaling Asynchronous Pipeline

Erman Culhacik

Inhaltsverzeichnis

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Einführung



Synchrone Pipeline

- CLK Geschwindigkeit durch langsamste Stufe (Logik Delay) begrenzt
- Leistungsverbrauch auch ohne Daten

Asynchrone Pipeline

- Maximaler Durchsatz (Summe aller Logik Delays)
- Kein Leistungsverbrauch wenn keine Daten anliegen

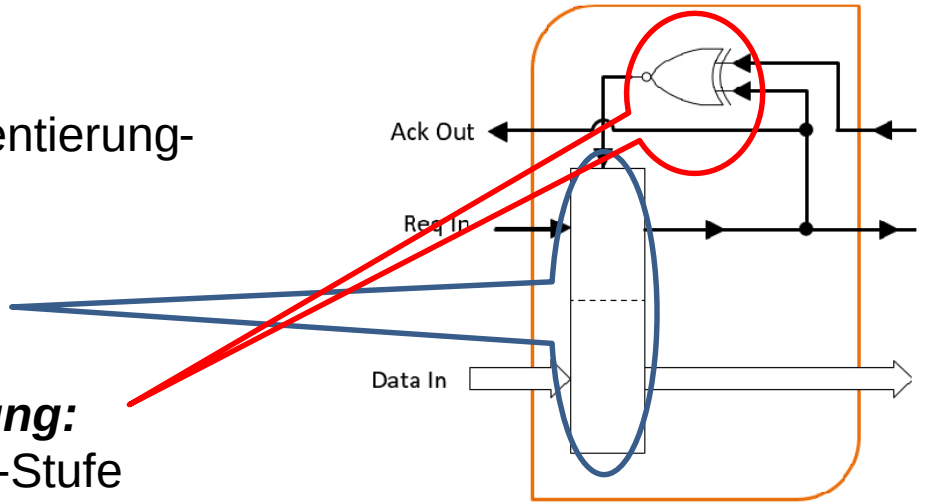
Einführung

Mousetrap Pipeline

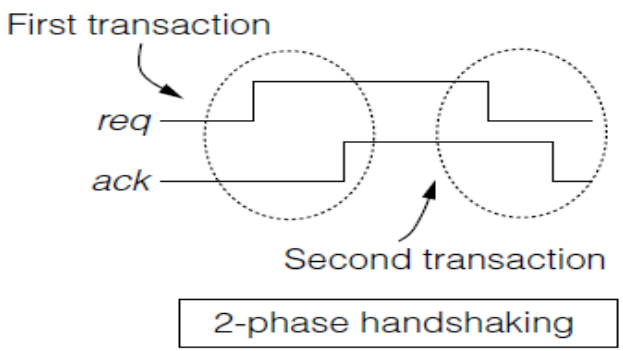
Einfache asynchrone Implementierung-Methode verwendet,

- ***Transparent Latches***
- ***Einfache Stufensteuerung:***
  1 Gatter pro Pipeline-Stufe
- ***Single-Rail bundled Data:***
  1 Leitung pro Daten

Der Zweck = sehr schnelle Taktzeit und niedriger Energieverbrauch.

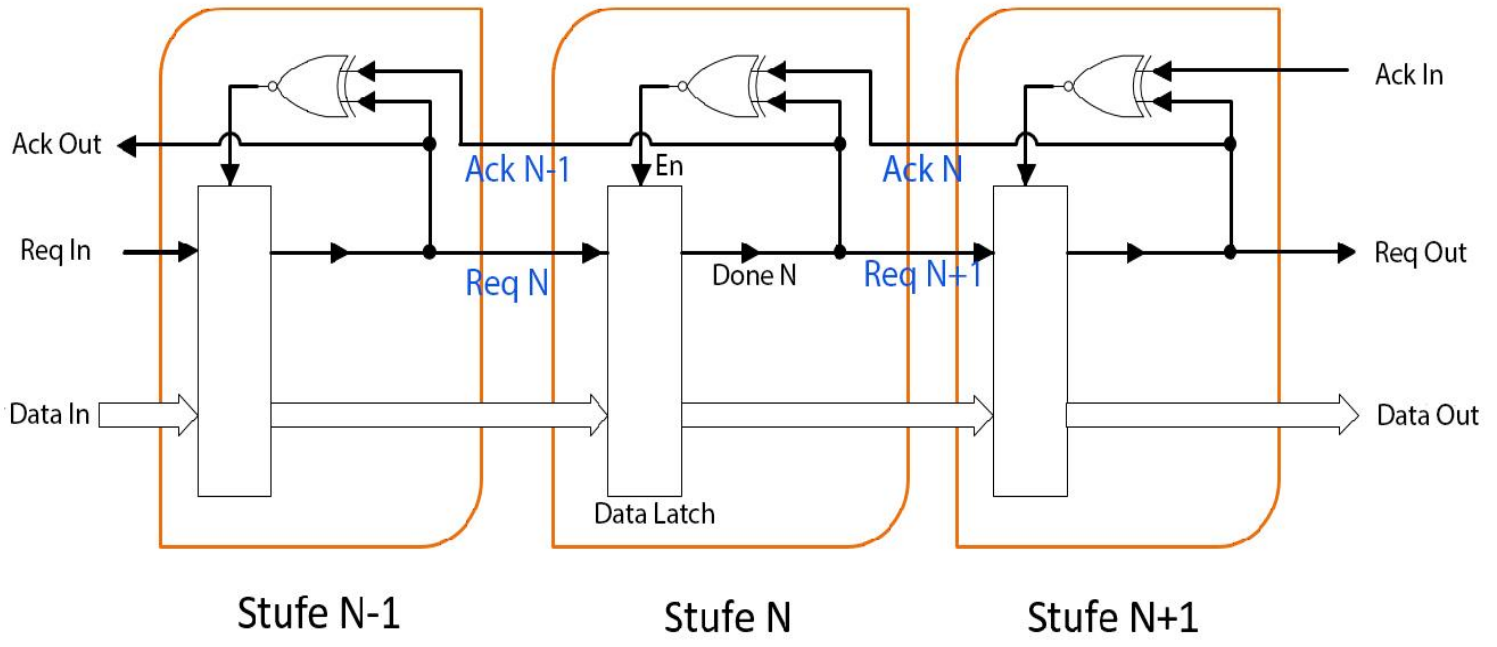


Eine Stufe von Mousetrap

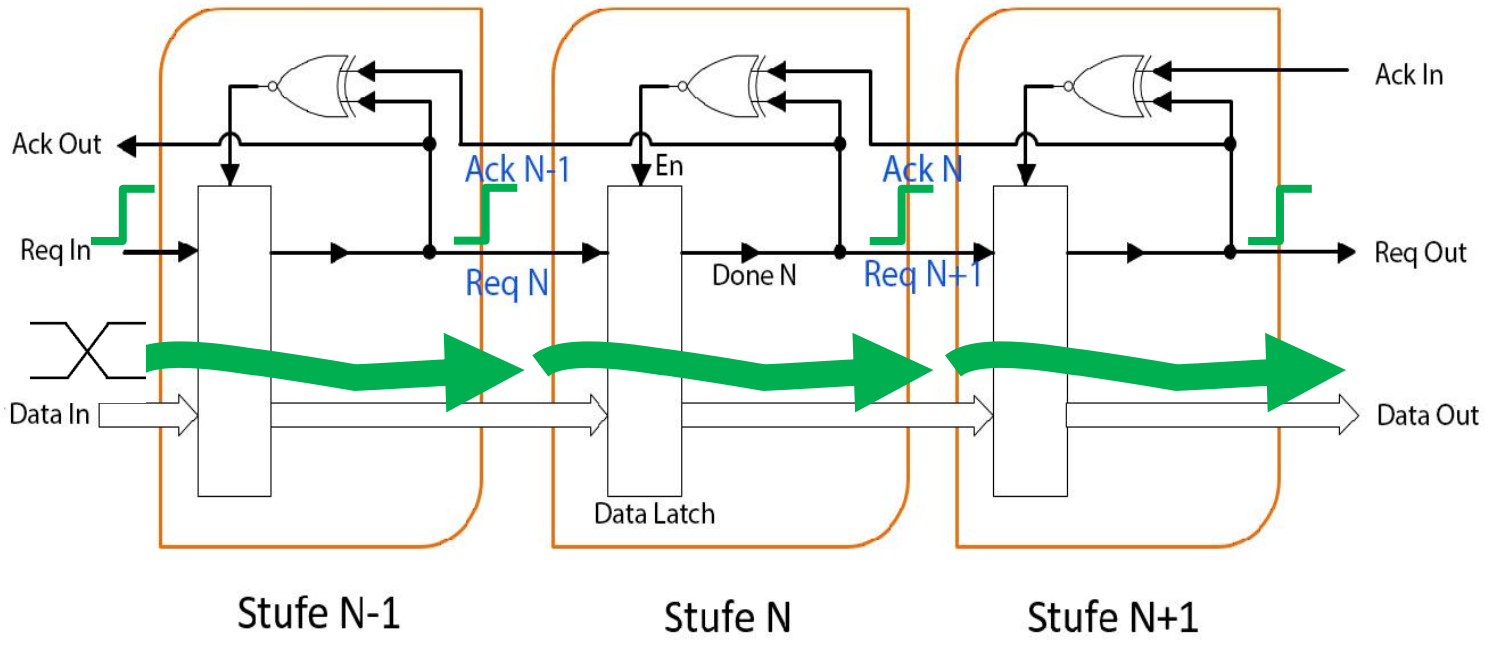


S. M. Nowick, M. Singh: High-Performance Asynchronous Pipelines: An Overview, Design & Test of Computers, IEEE vol.28, iss: 5 , p.8-22, Oct. 2011

Prinzip: Einfaches FIFO

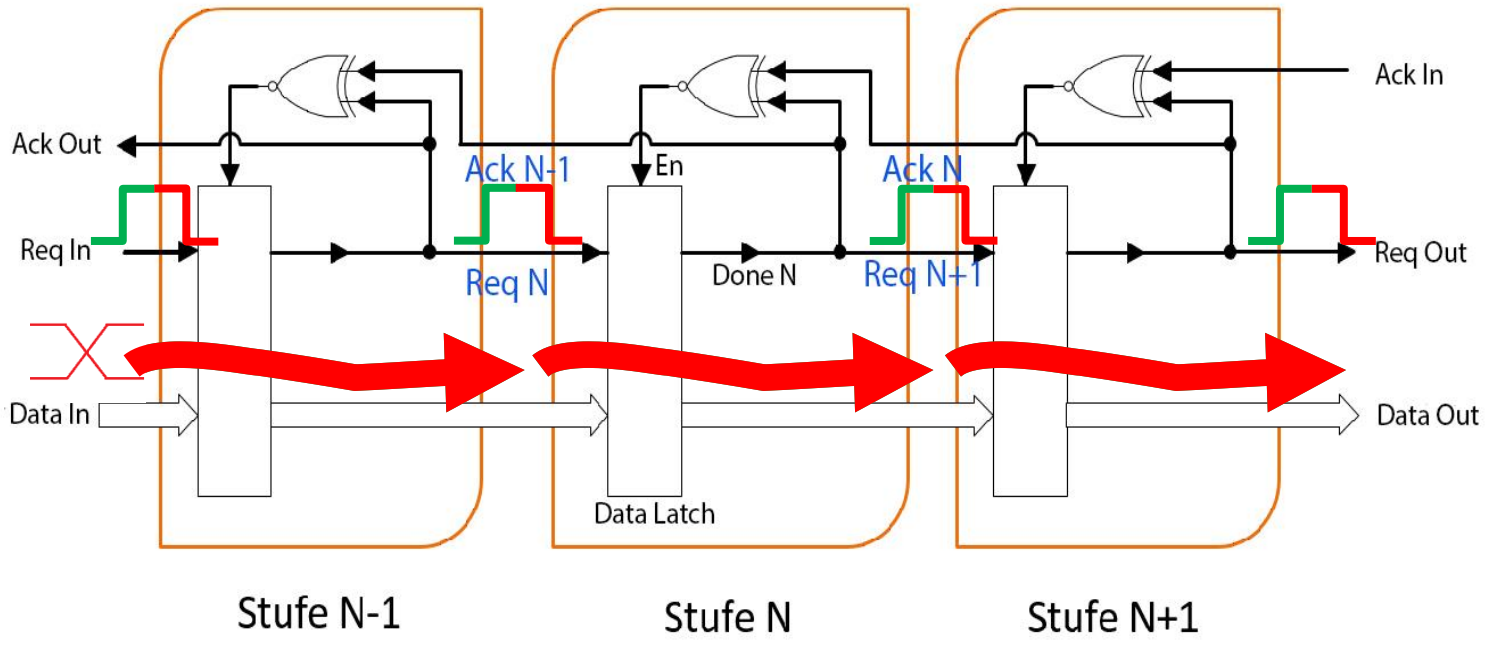


Prinzip: Einfaches FIFO



Erste Transaktion

Prinzip: Einfaches FIFO



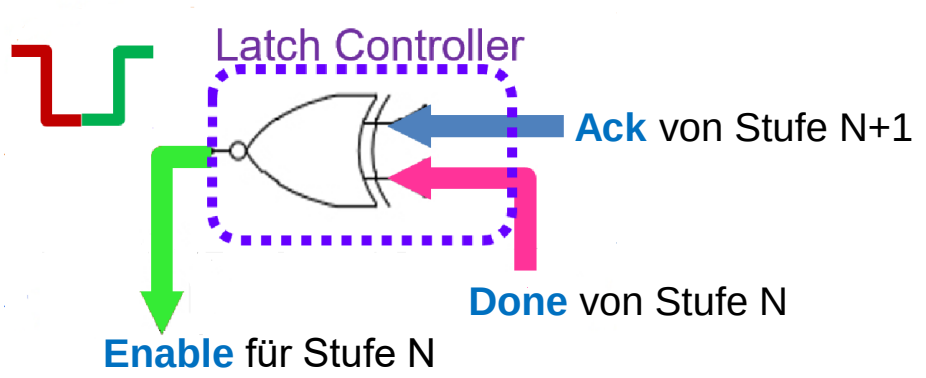
Zweite Transaktion

Prinzip : Latch Controller

XNOR-Betrieb

XNOR

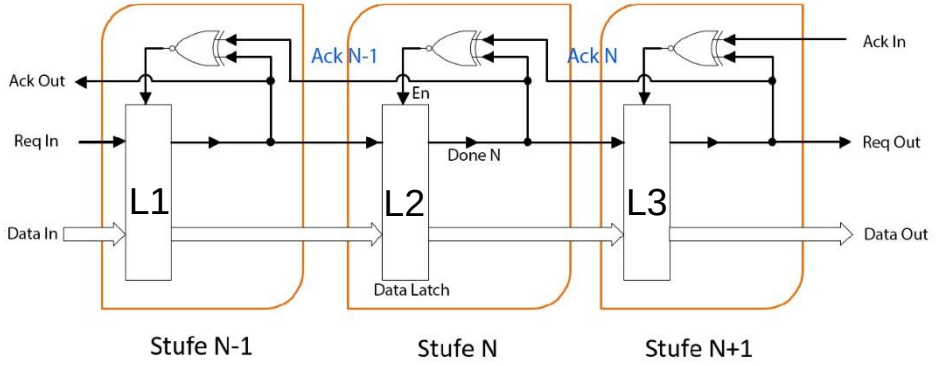
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1



- **Down transition :** durch Endsignal von die Stufe N verursacht
- **Up transition :** durch Endsignal von die Stufe N+1 verursacht

Prinzip :

Mousetrap-Verhalten

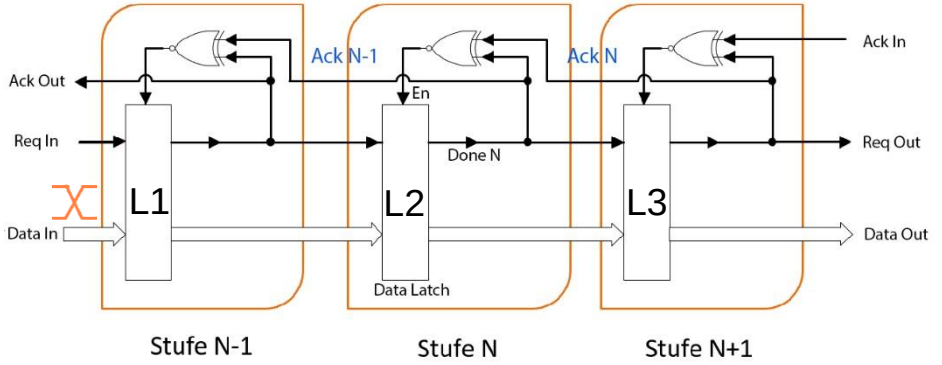


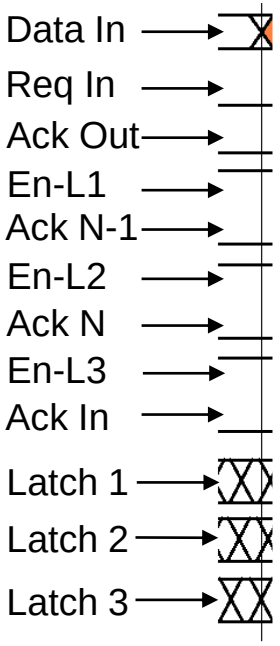
Data In ⟶
Req In ⟶
Ack Out ⟶
En-L1 ⟶
Ack N-1 ⟶
En-L2 ⟶
Ack N ⟶
En-L3 ⟶
Ack In ⟶

Latch 1 ⟶
Latch 2 ⟶
Latch 3 ⟶

Prinzip :

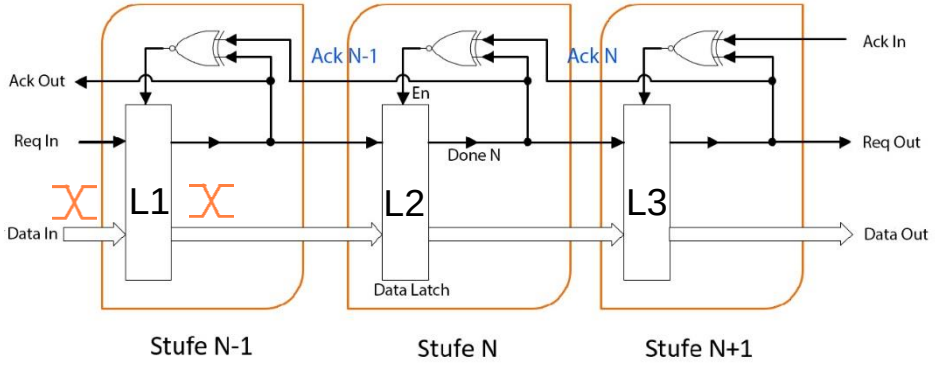
Mousetrap-Verhalten

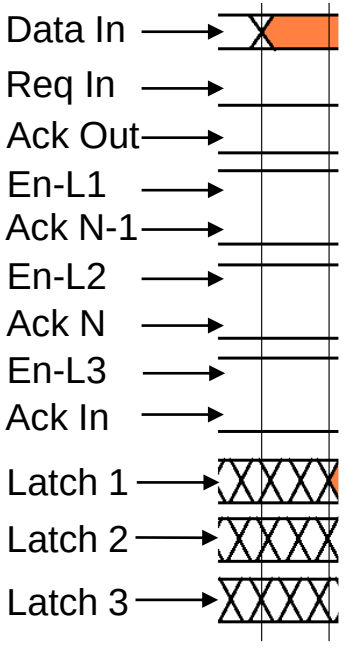




Prinzip :

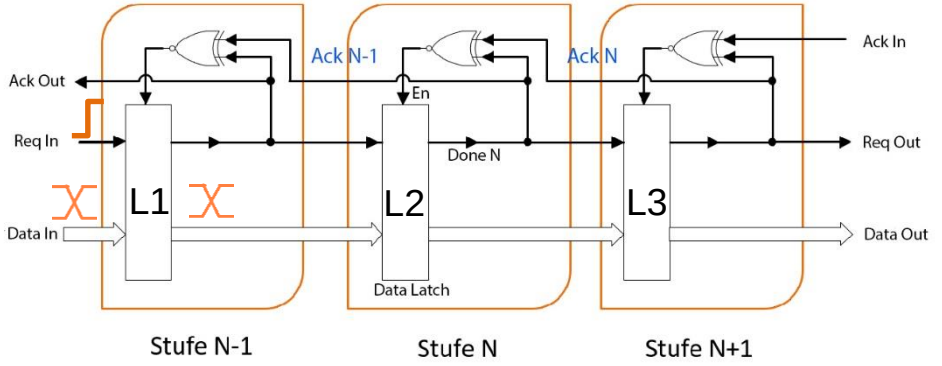
Mousetrap-Verhalten

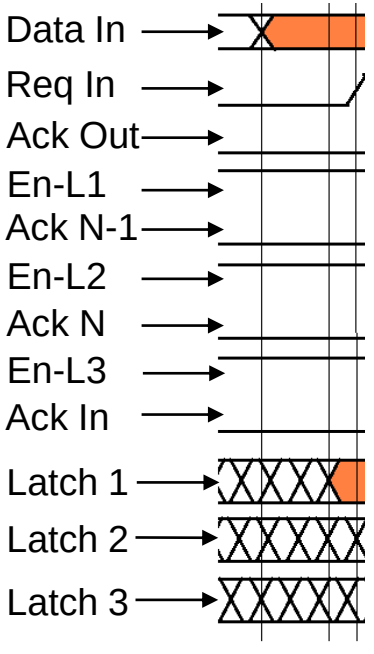




Prinzip :

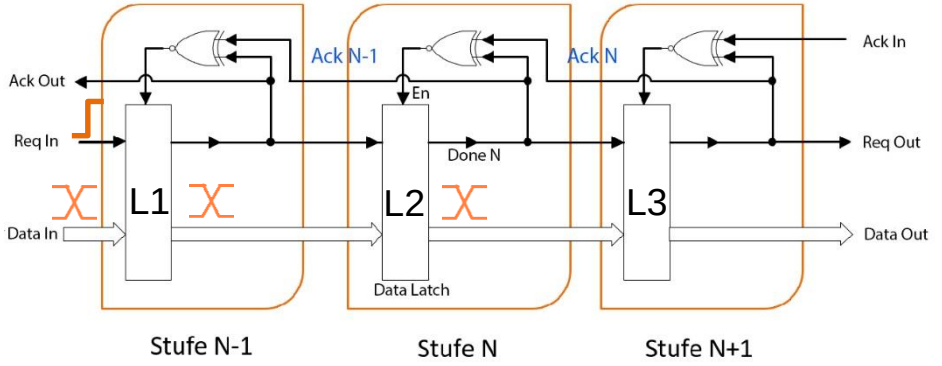
Mousetrap-Verhalten

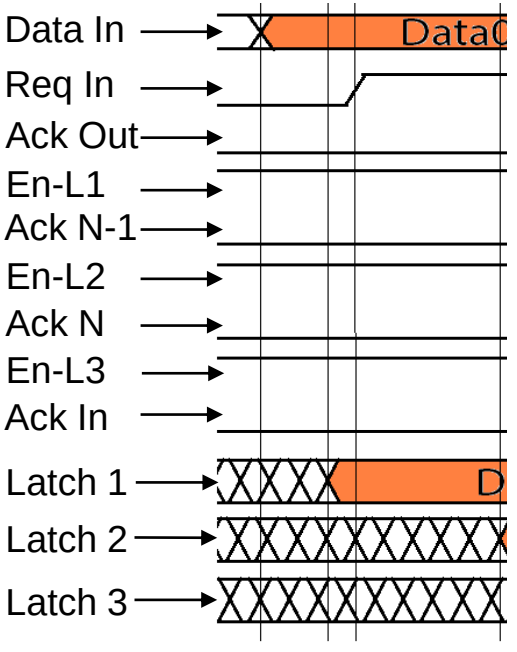




Prinzip :

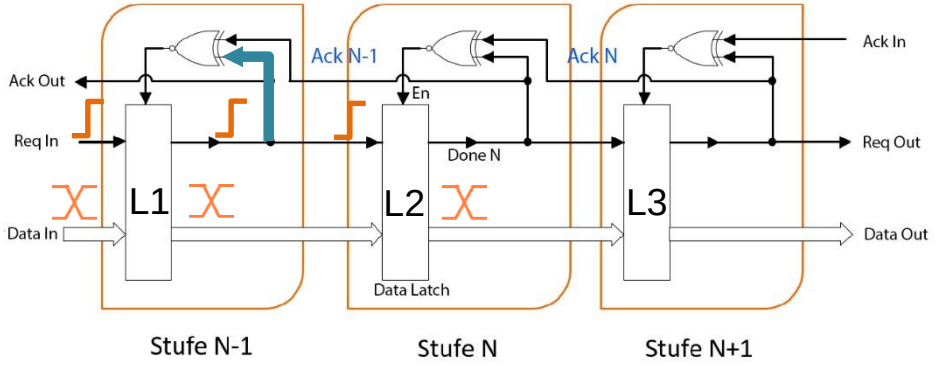
Mousetrap-Verhalten

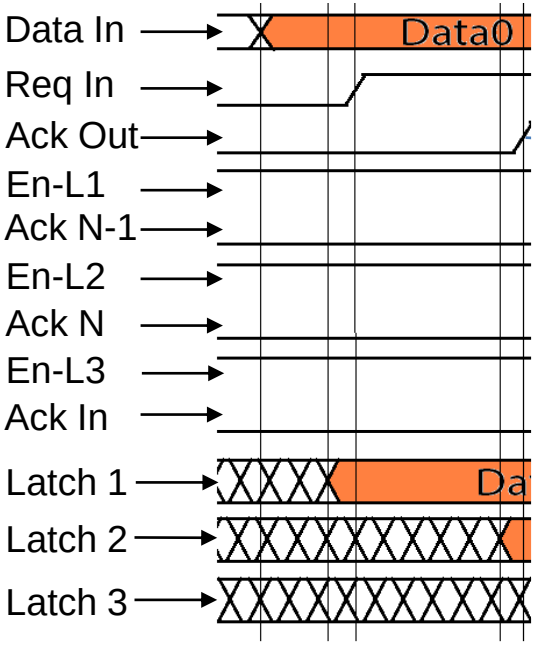




Prinzip :

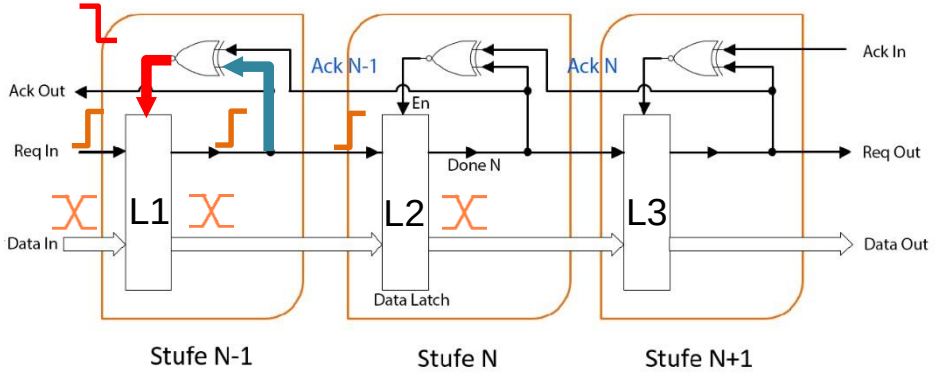
Mousetrap-Verhalten

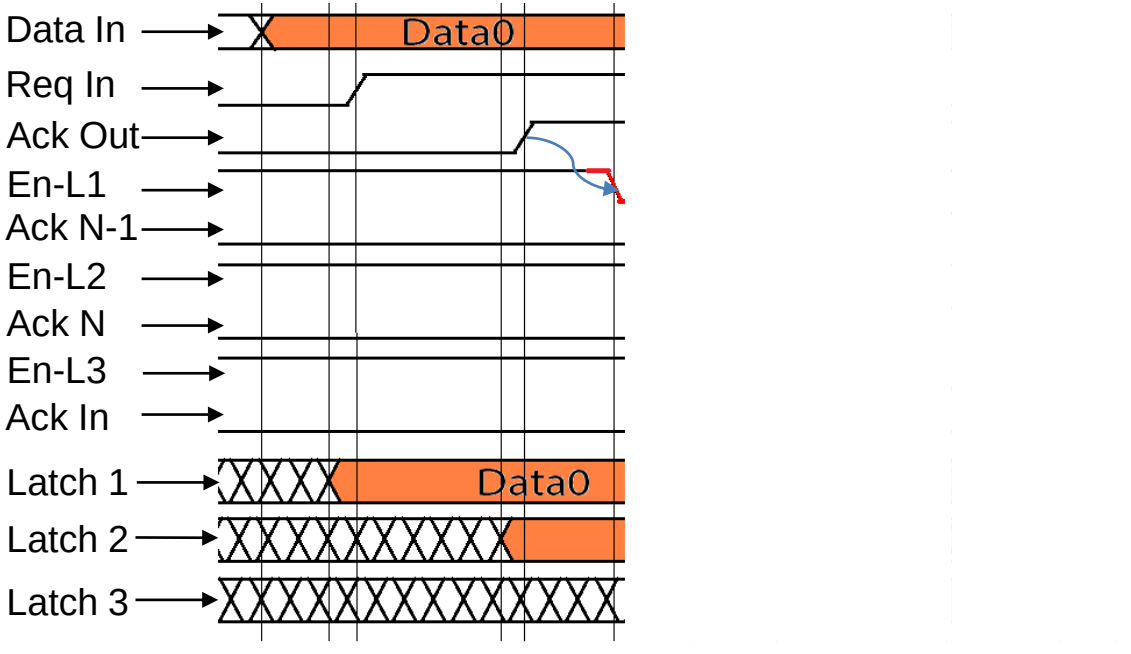




Prinzip :

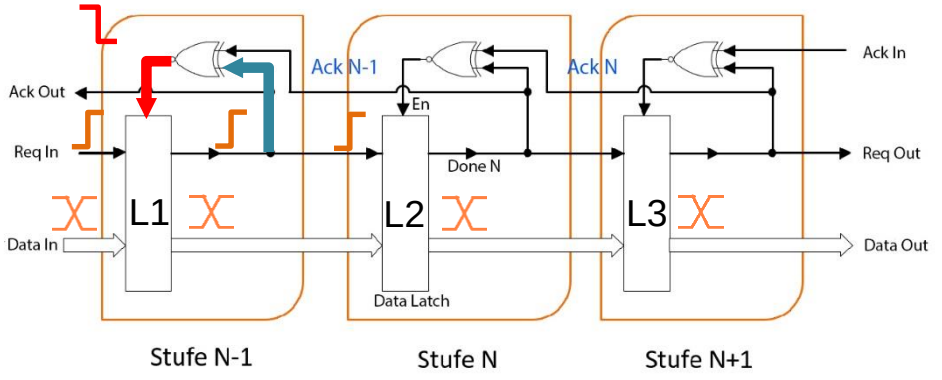
Mousetrap-Verhalten

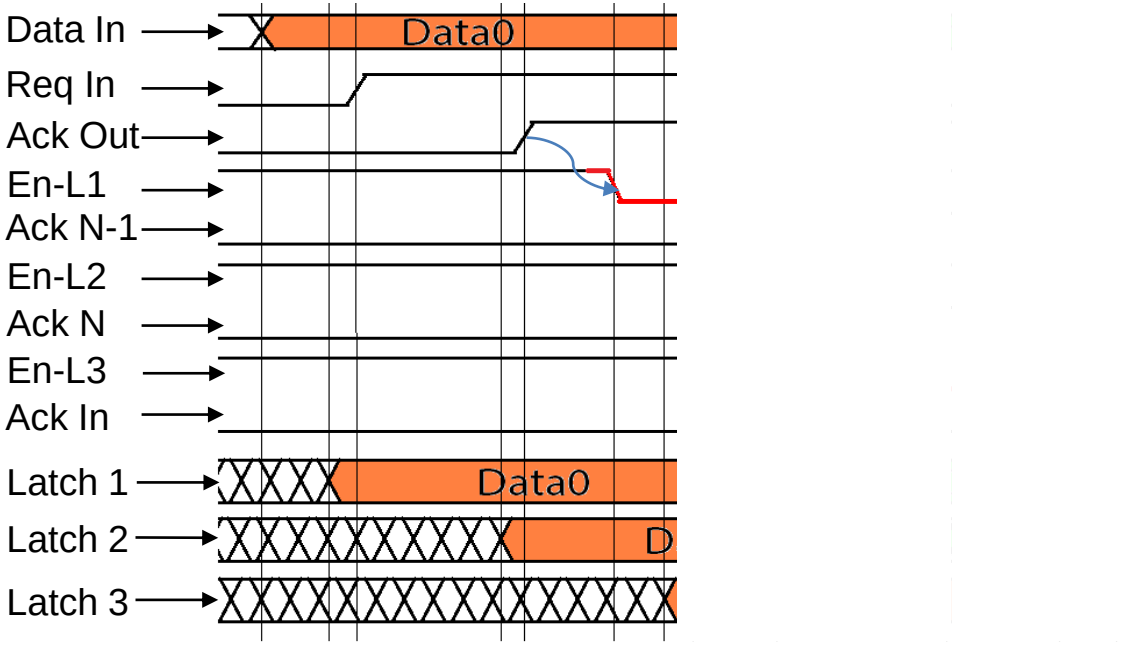




Prinzip :

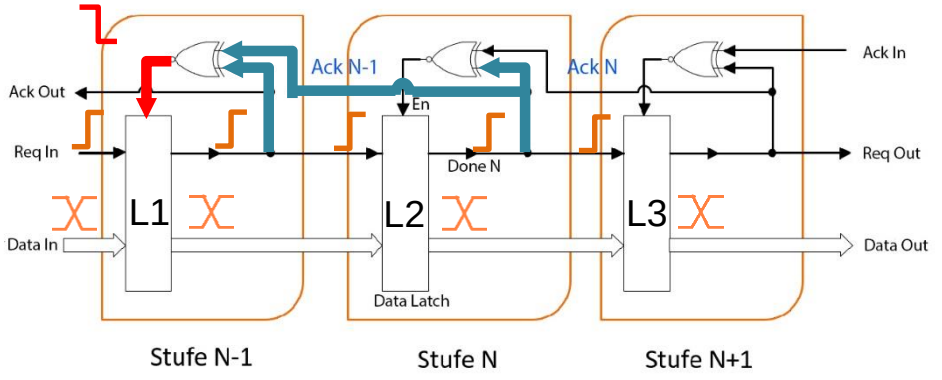
Mousetrap-Verhalten

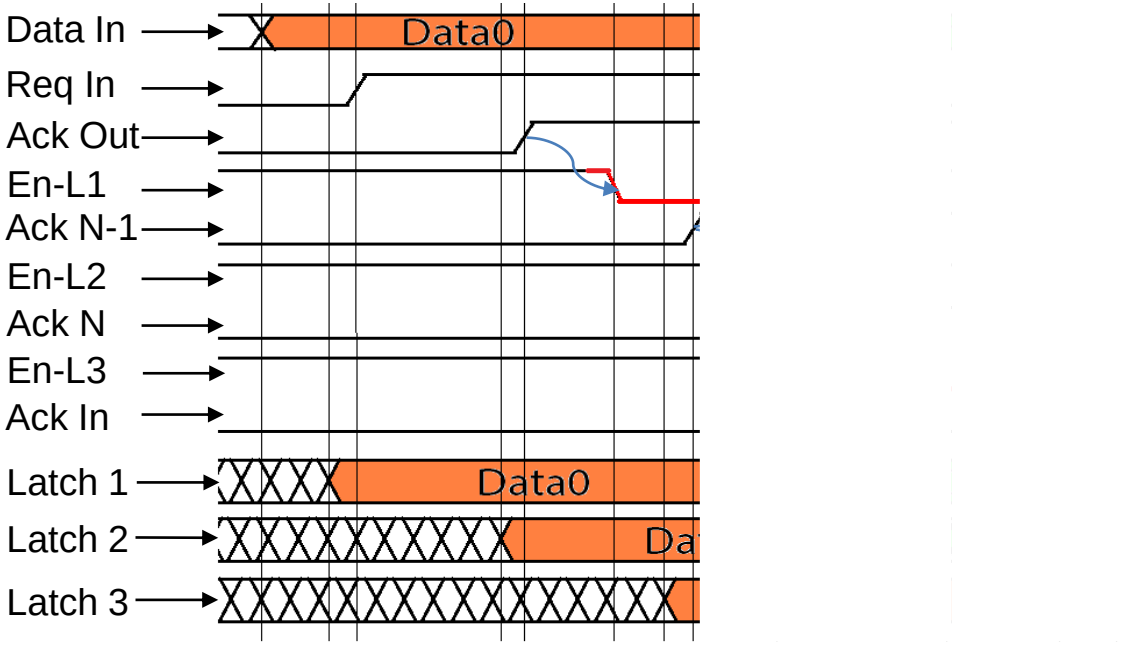




Prinzip :

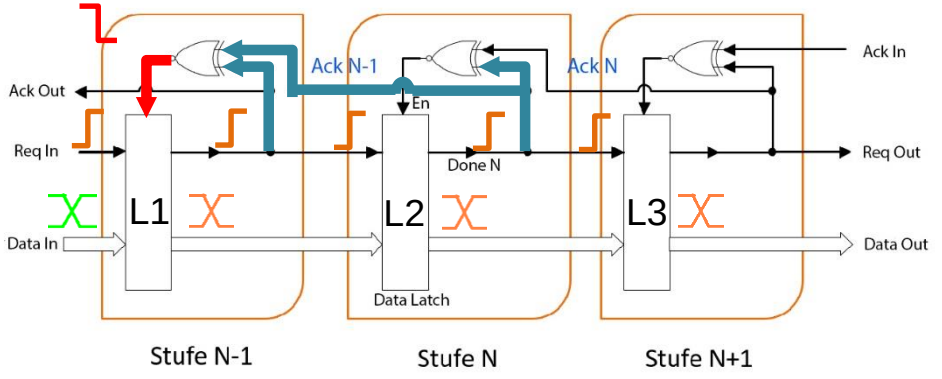
Mousetrap-Verhalten

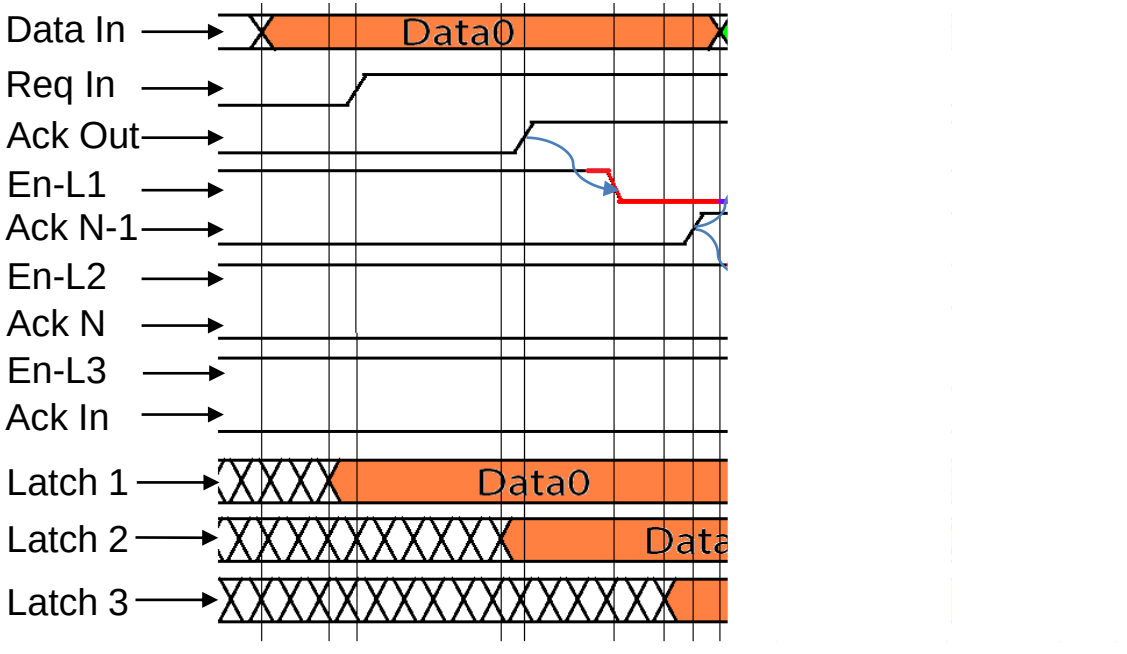




Prinzip :

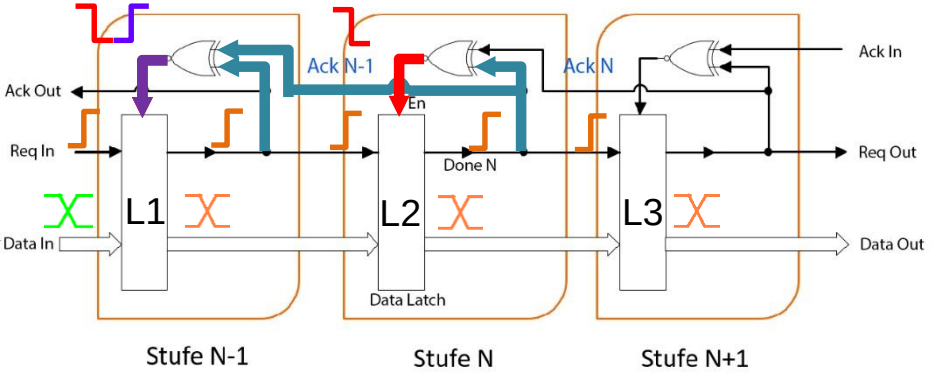
Mousetrap-Verhalten

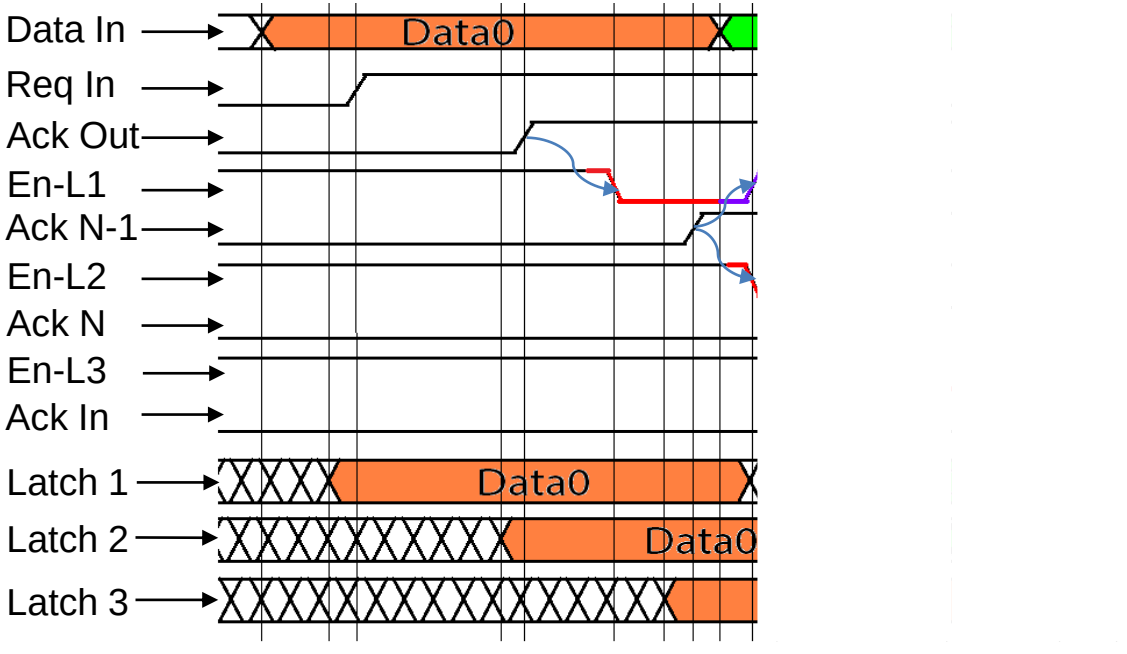




Prinzip :

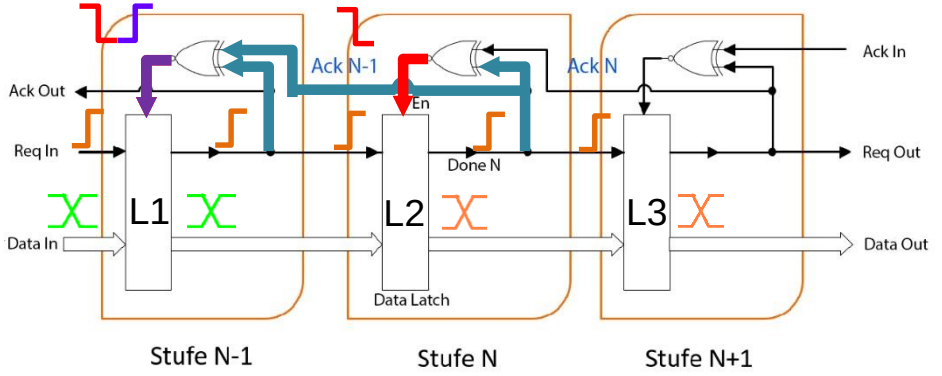
Mousetrap-Verhalten

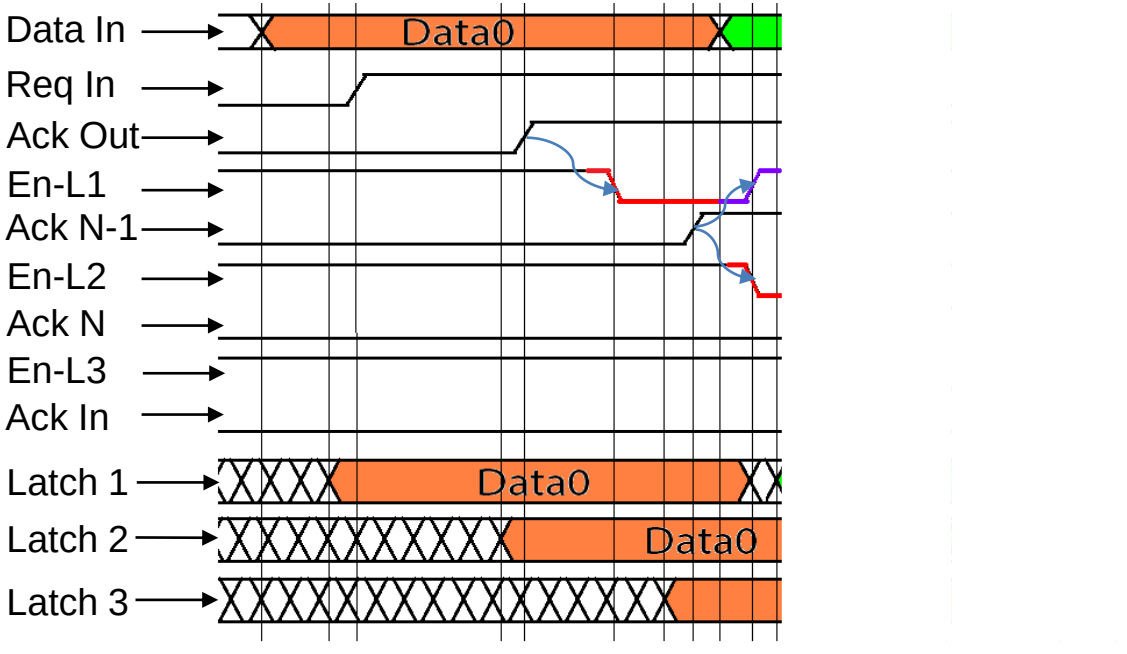




Prinzip :

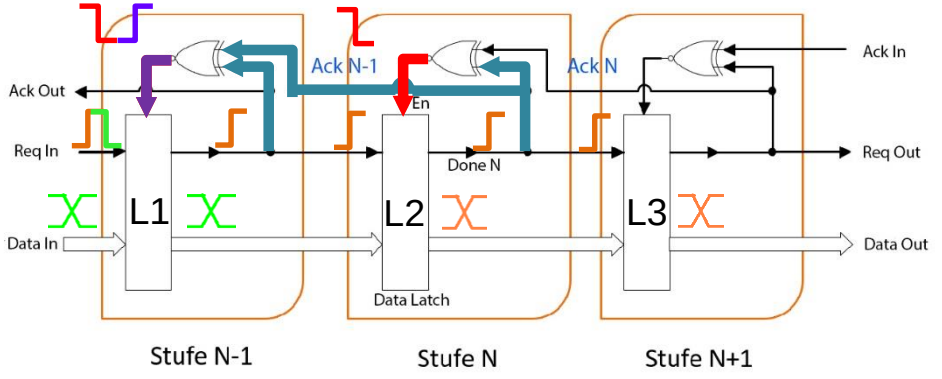
Mousetrap-Verhalten

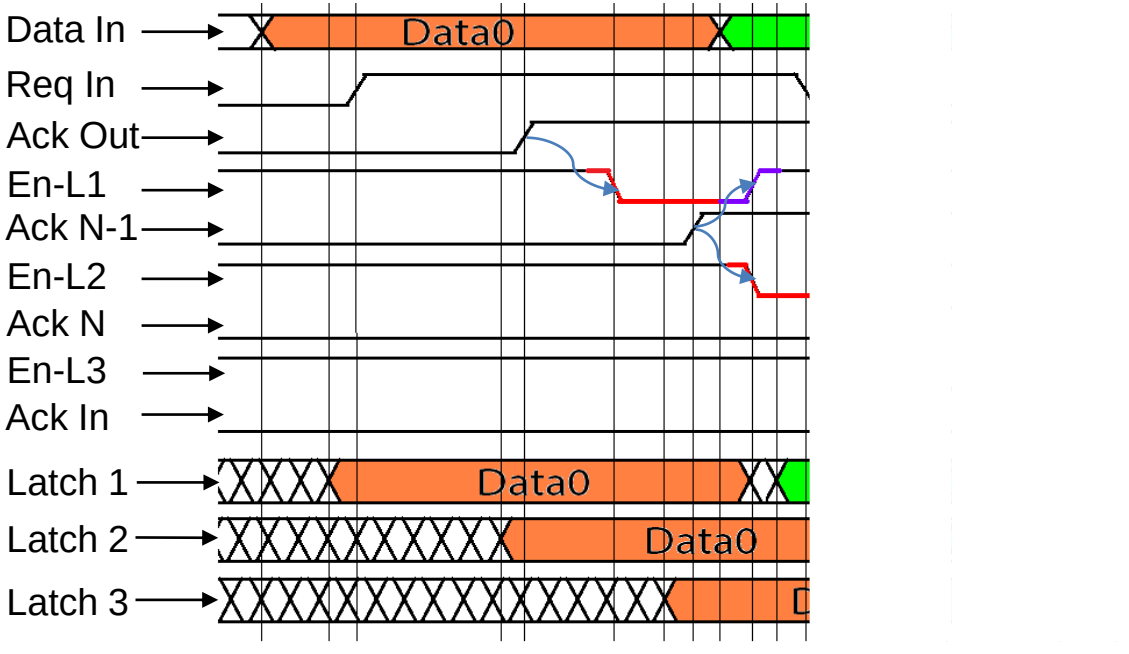




Prinzip :

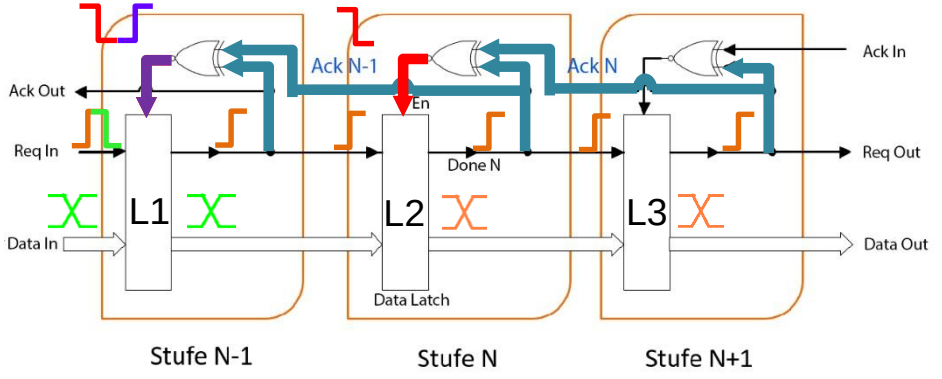
Mousetrap-Verhalten

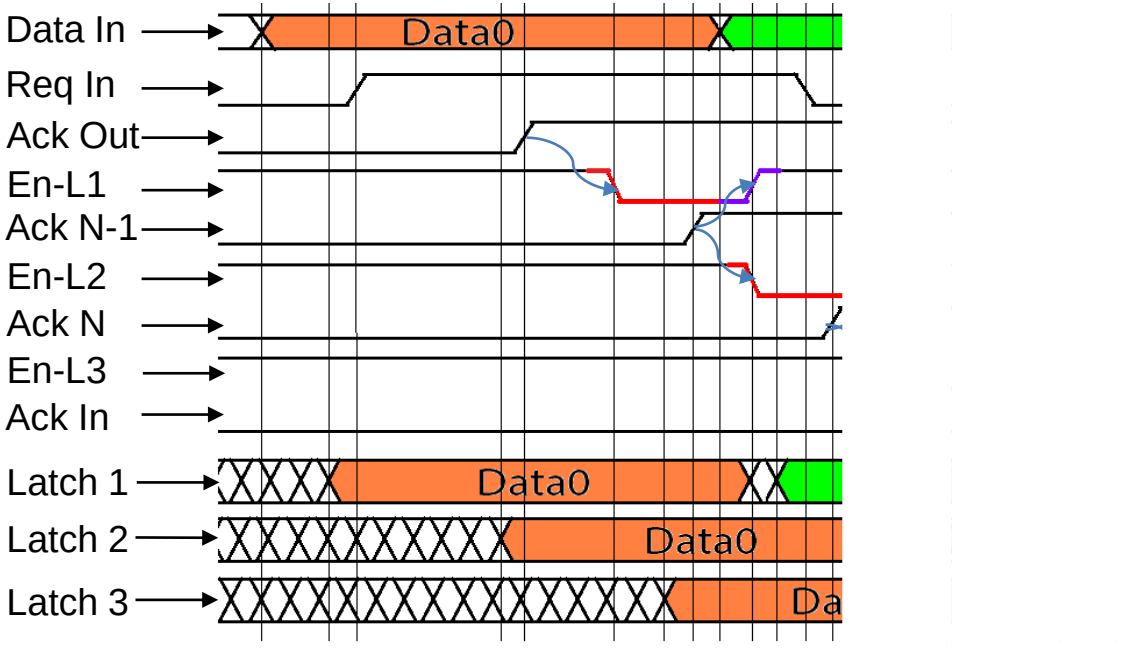




Prinzip :

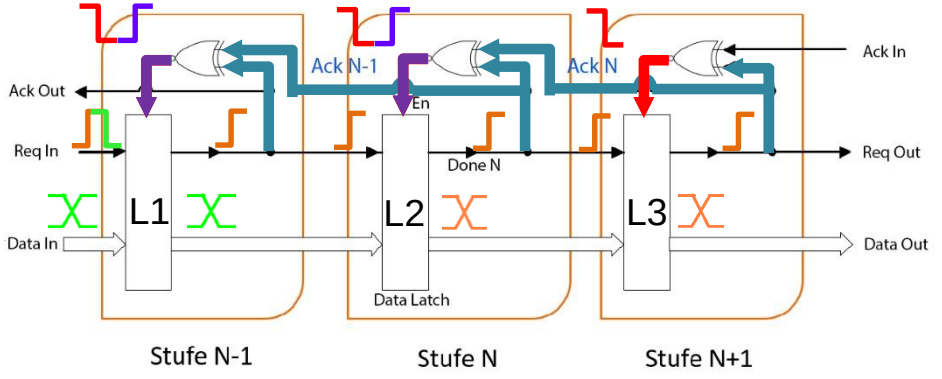
Mousetrap-Verhalten

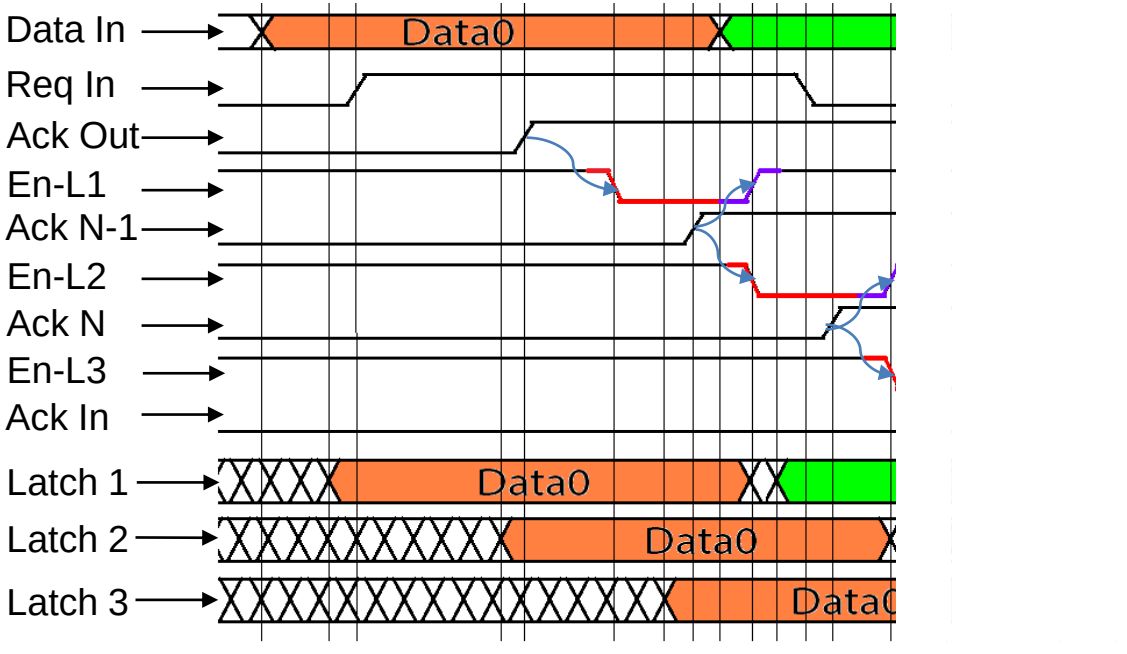




Prinzip :

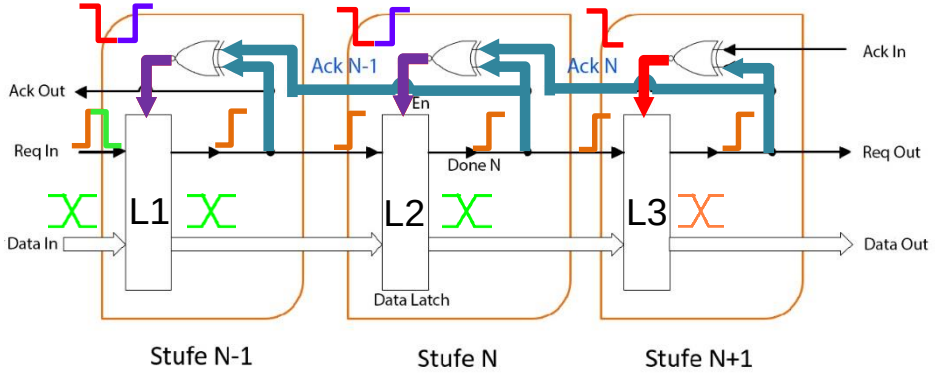
Mousetrap-Verhalten

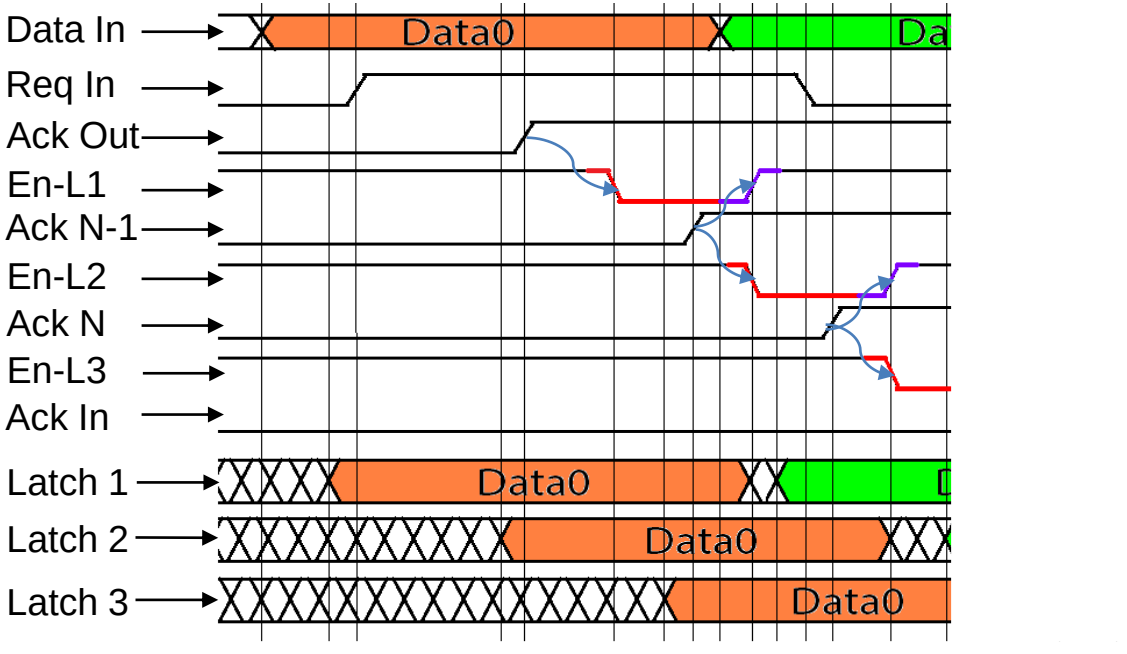




Prinzip :

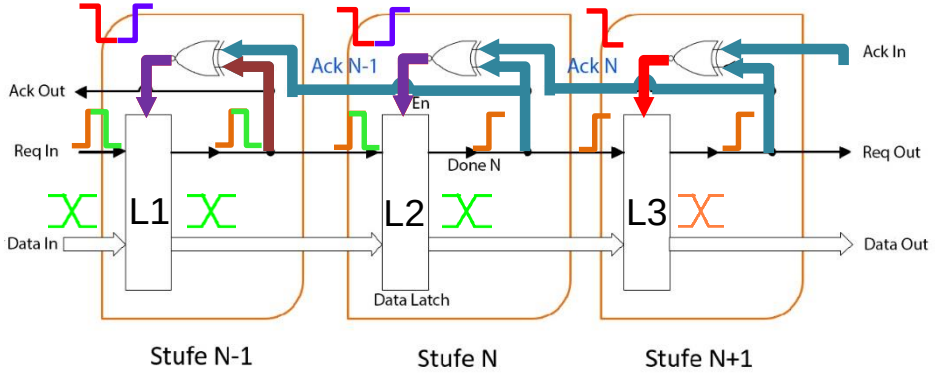
Mousetrap-Verhalten

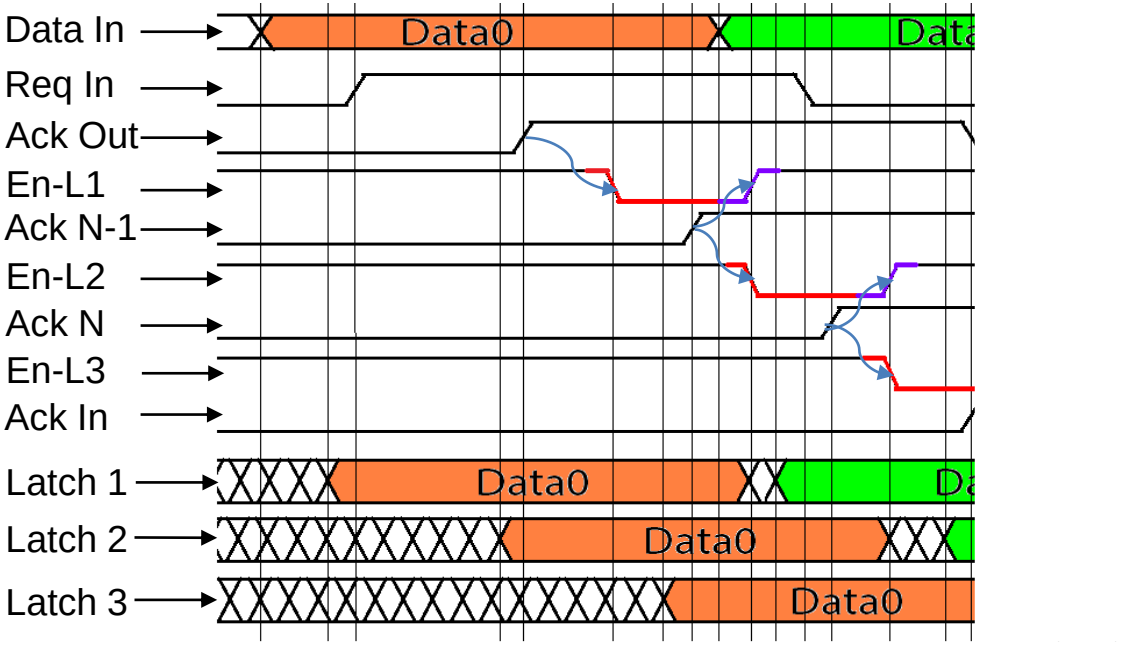




Prinzip :

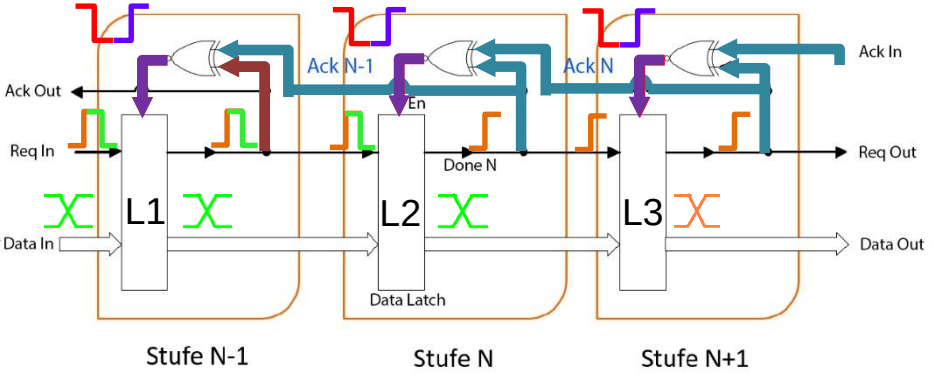
Mousetrap-Verhalten

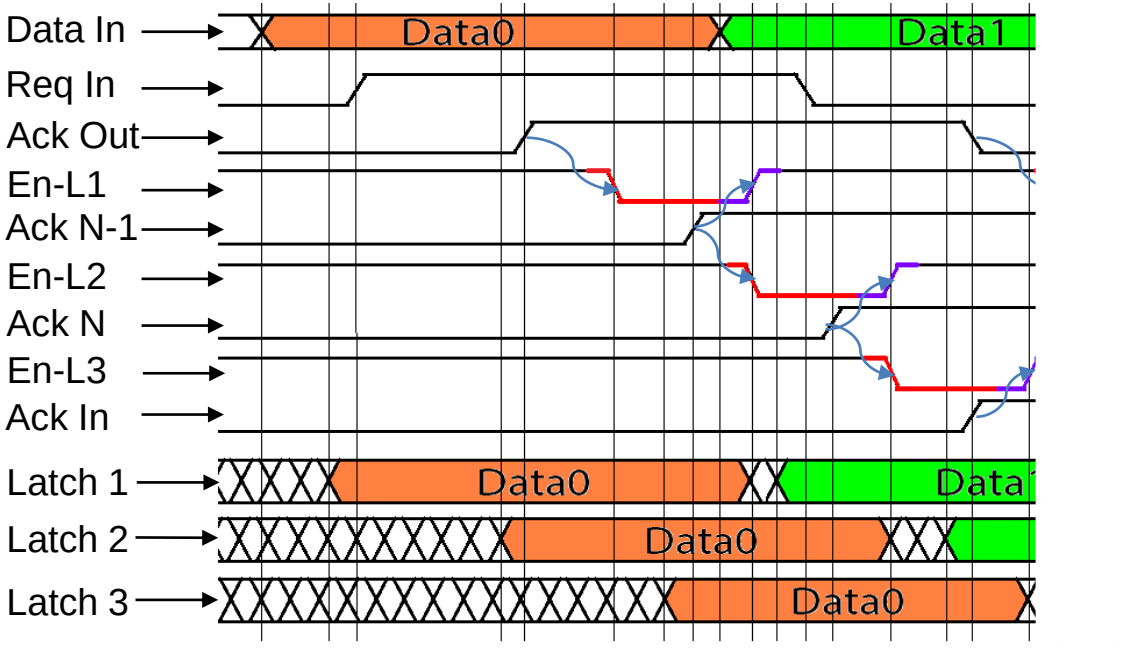




Prinzip :

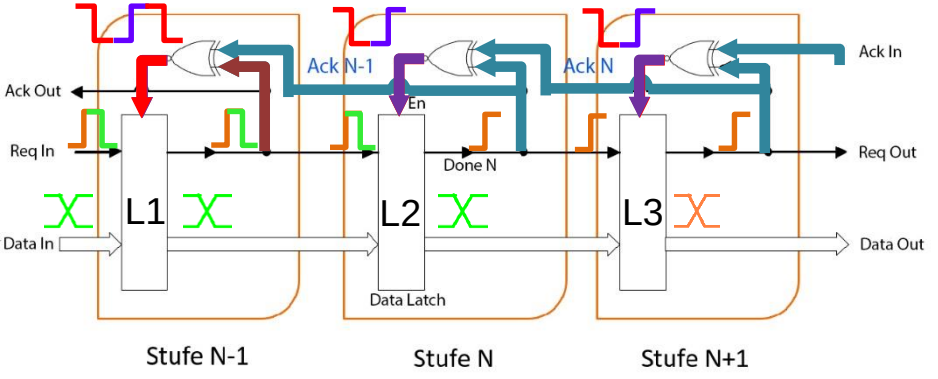
Mousetrap-Verhalten

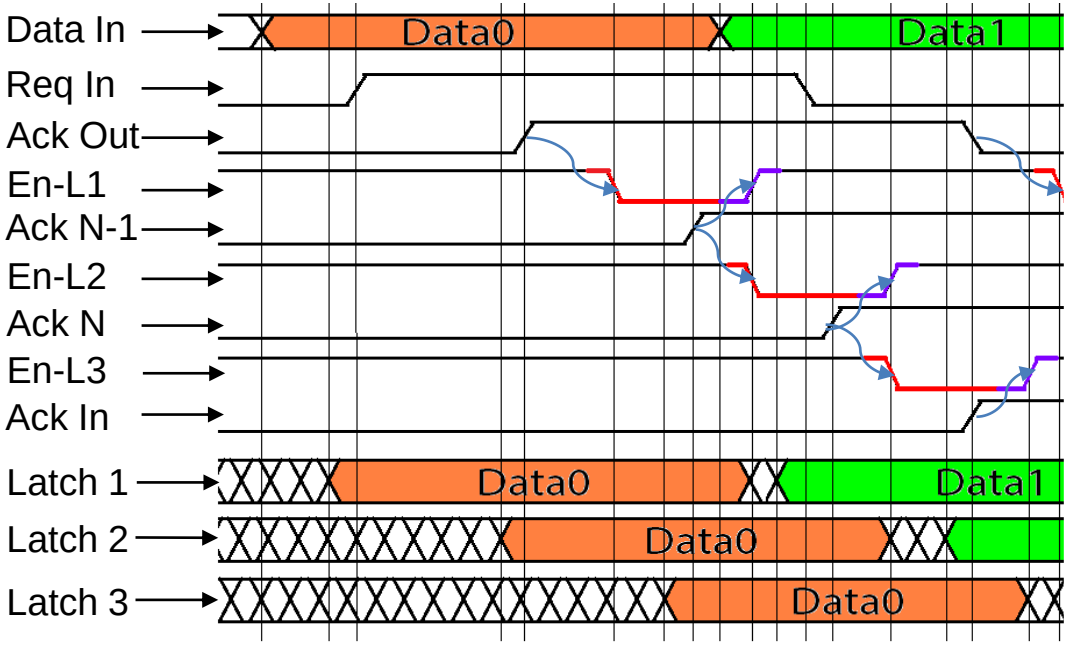




Prinzip :

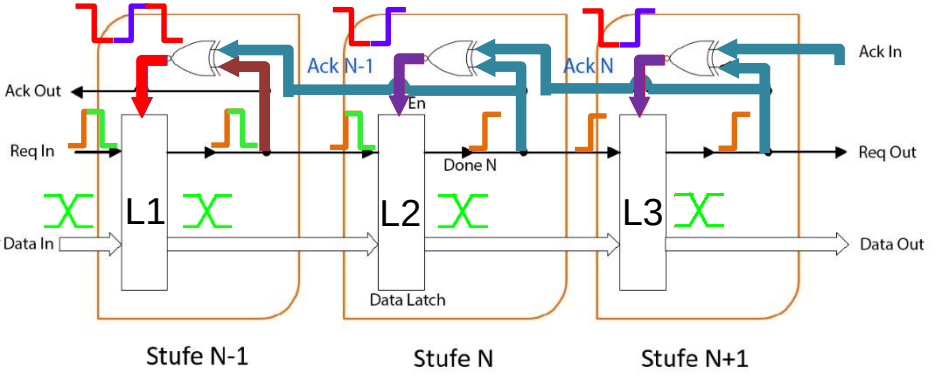
Mousetrap-Verhalten

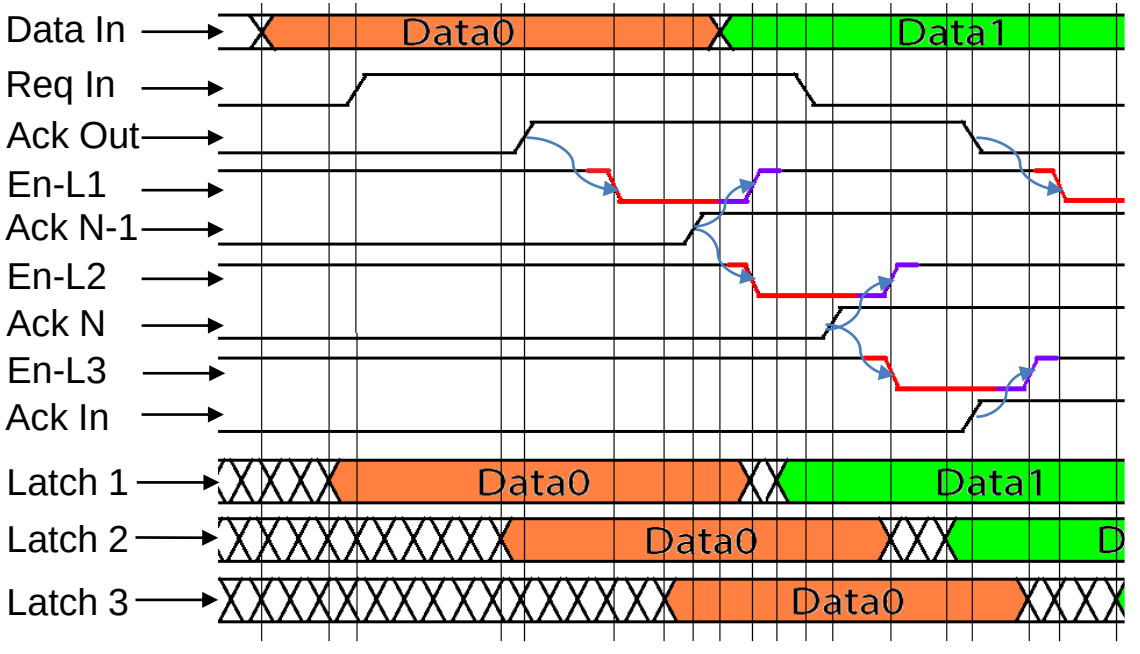




Prinzip :

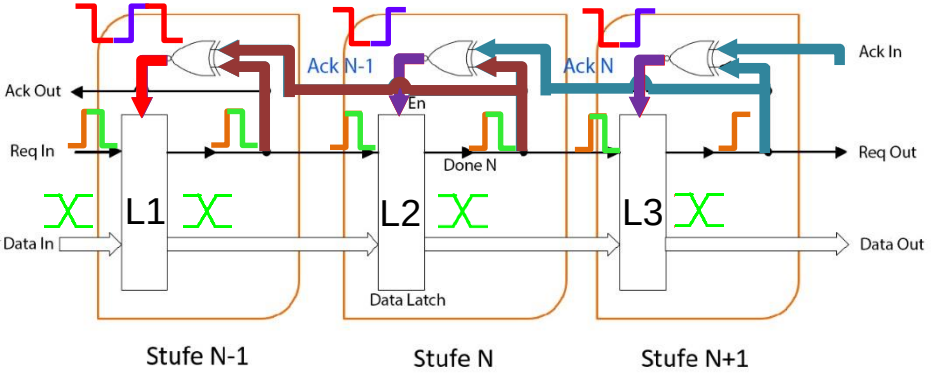
Mousetrap-Verhalten

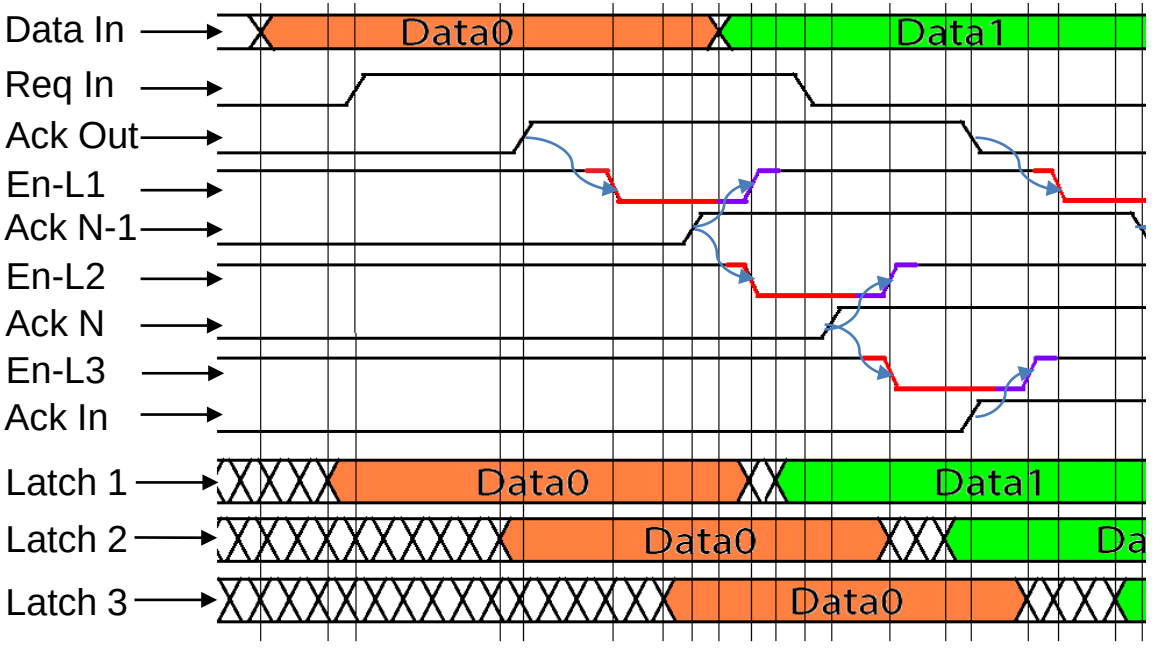




Prinzip :

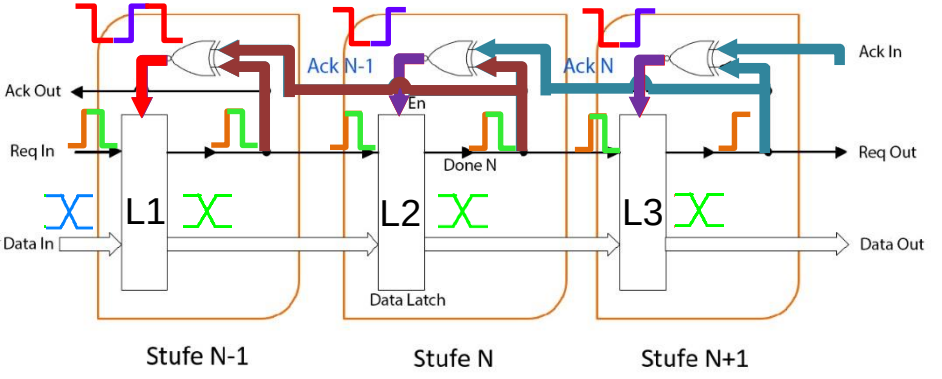
Mousetrap-Verhalten

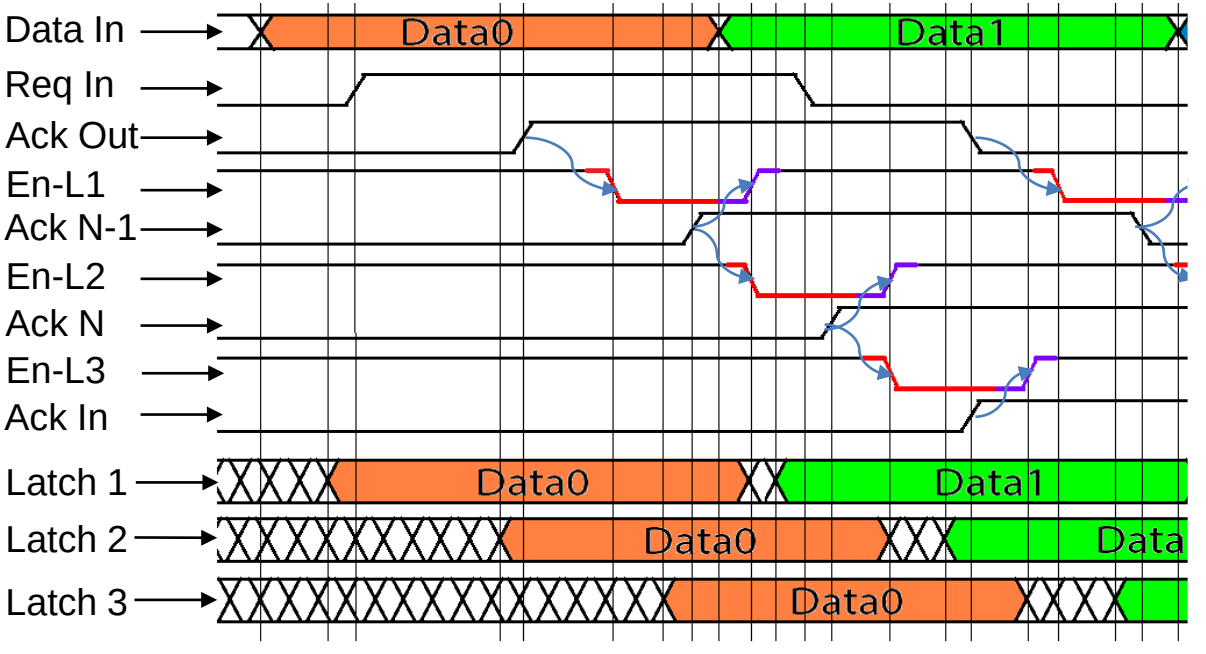




Prinzip :

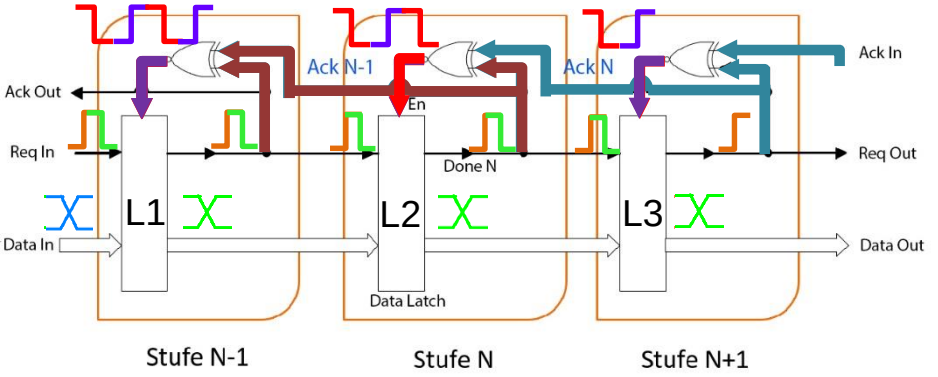
Mousetrap-Verhalten

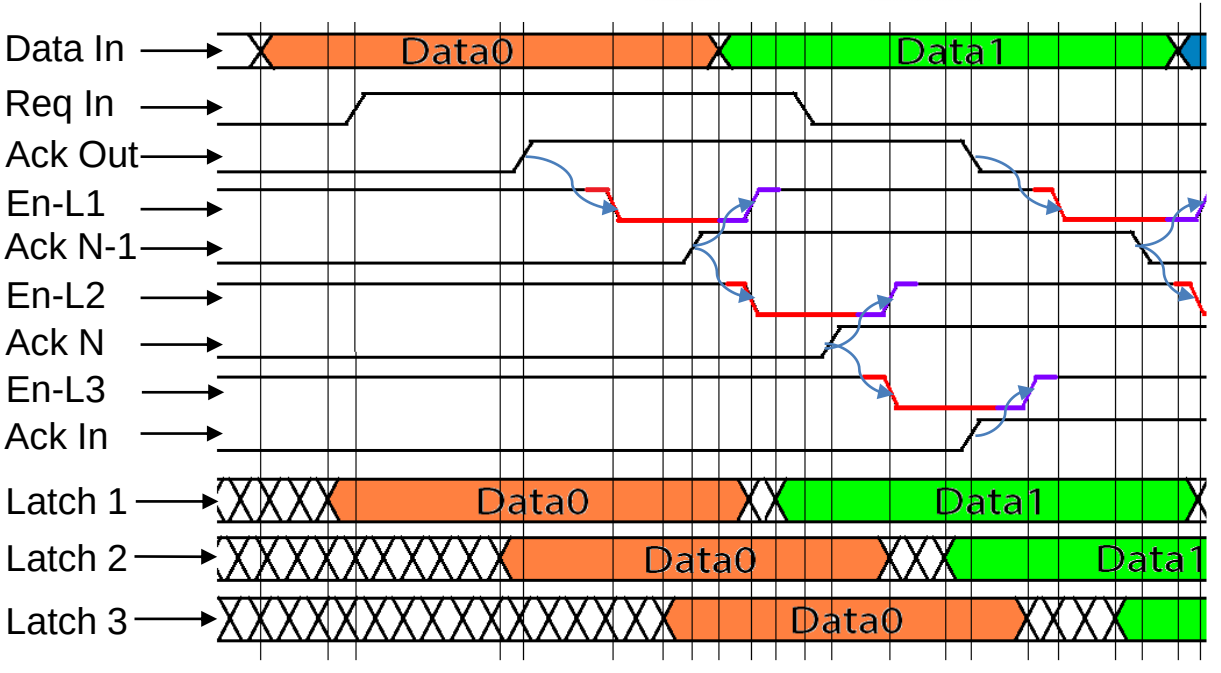




Prinzip :

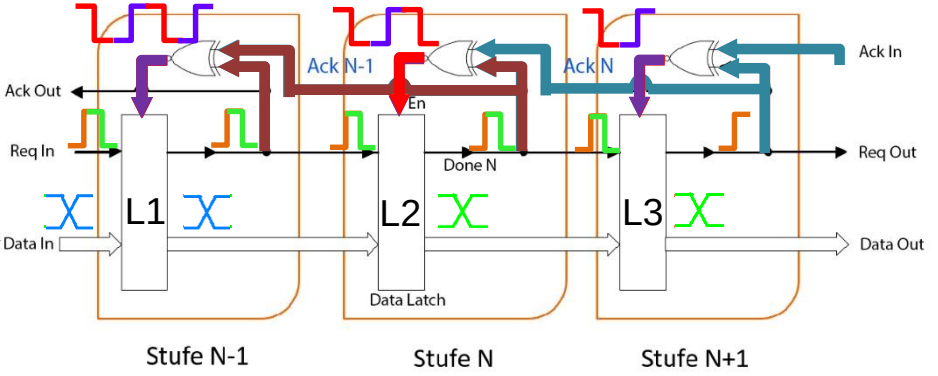
Mousetrap-Verhalten

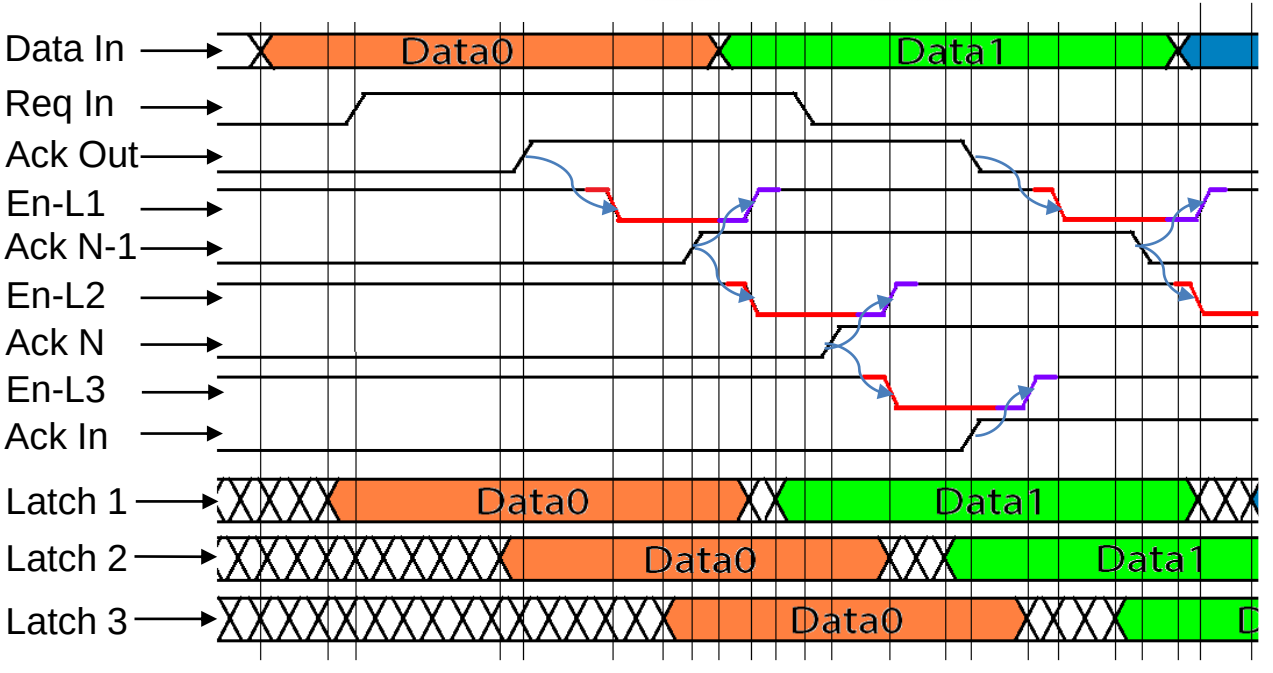




Prinzip :

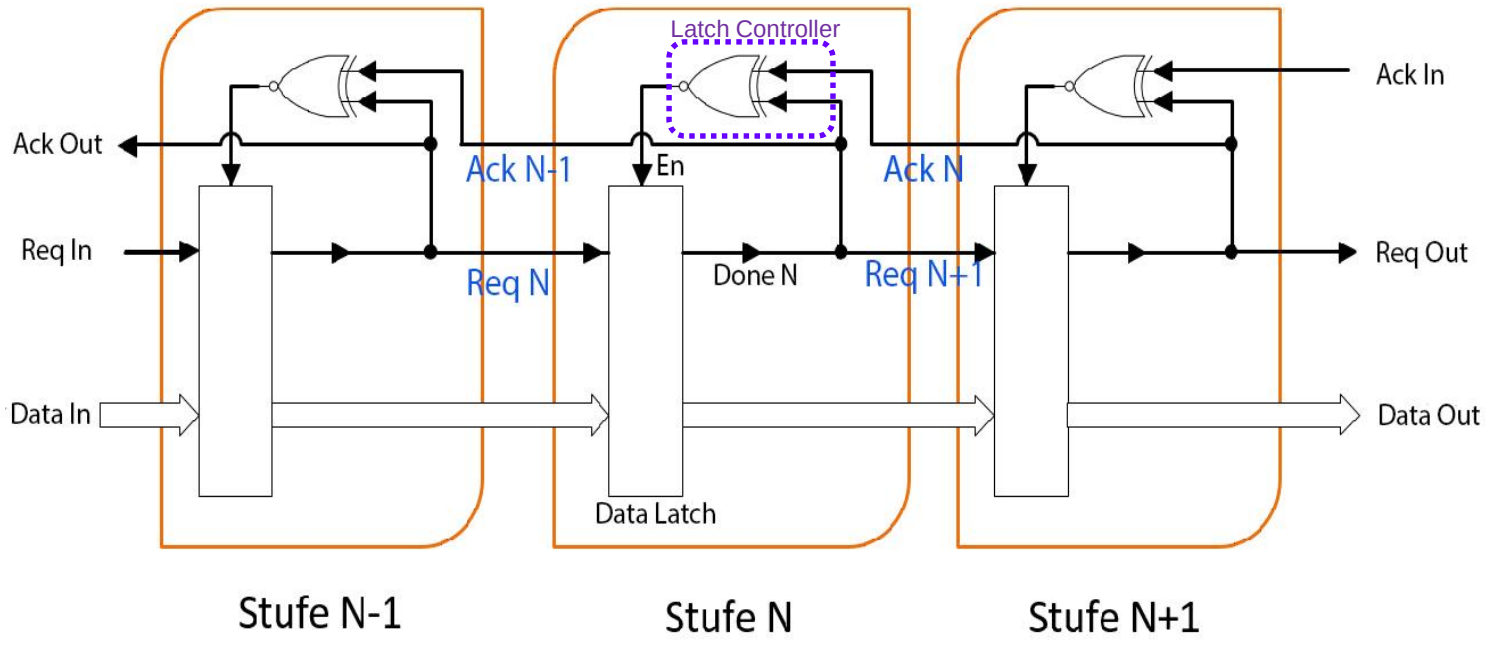
Mousetrap-Verhalten





Timing : Pipeline

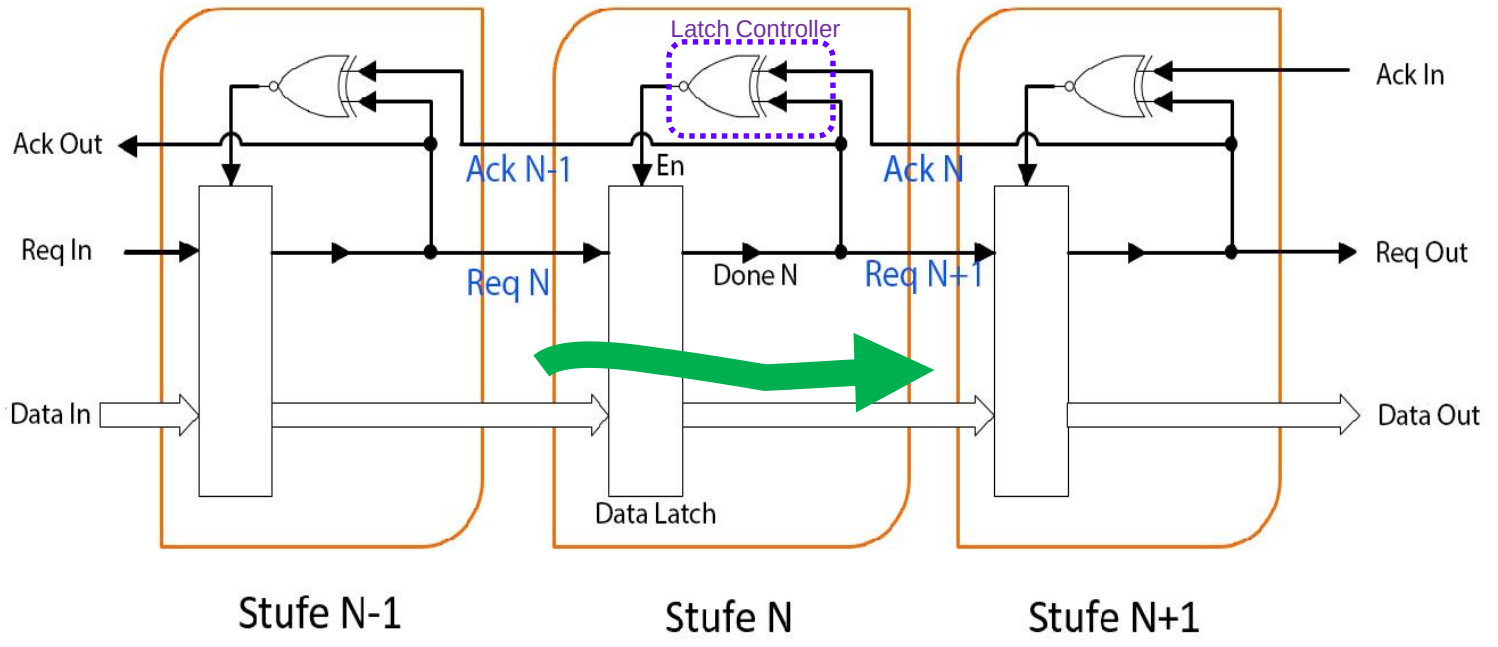
FIFO Cycle Time



Der Zykluszeit:

Timing : Pipeline

FIFO Cycle Time

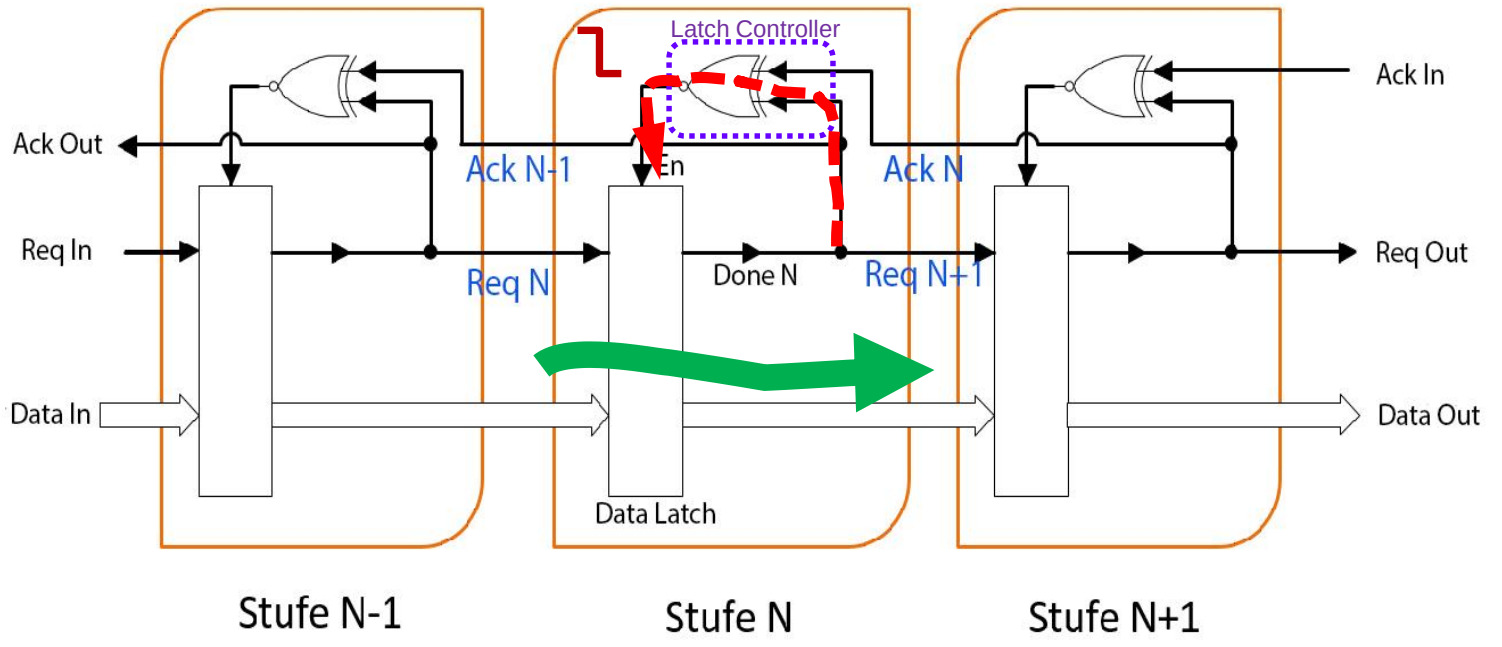


Der Zykluszeit: $T_{Latch\ N}$

Timing : Pipeline

FIFO Cycle Time

Schnelle Selbstschleife :
N selbst deaktiviert

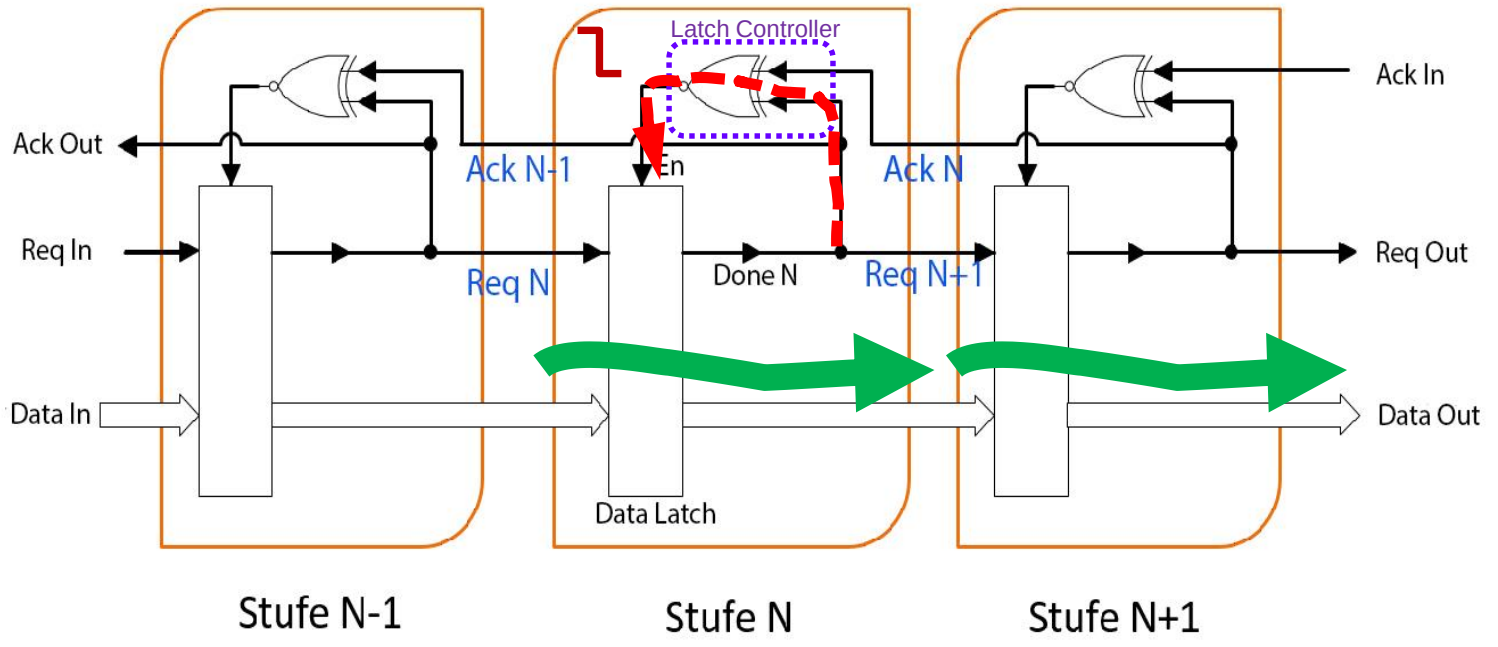


Der Zykluszeit: $T_{Latch\ N}$

Timing : Pipeline

FIFO Cycle Time

Schnelle Selbstschleife :
N selbst deaktiviert

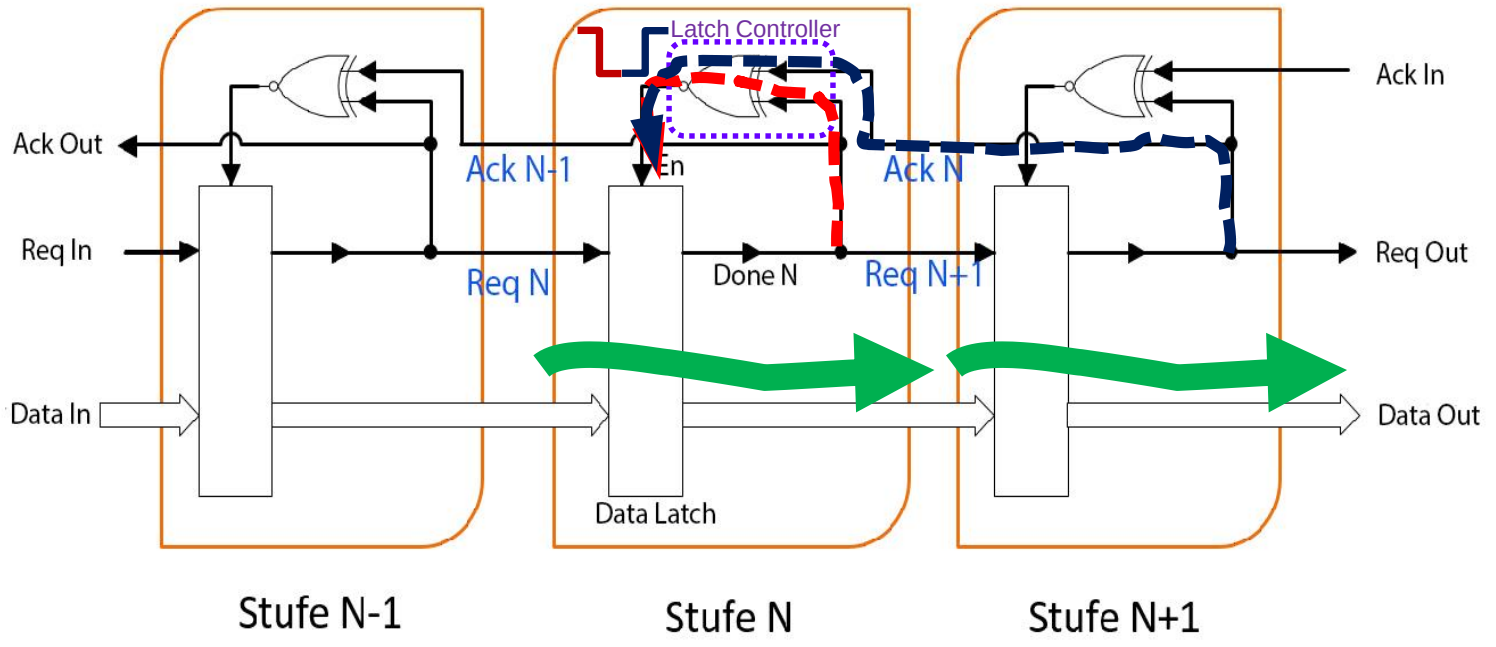


Der Zykluszeit: $T_{Latch\ N} + T_{Latch\ N+1}$

Timing : Pipeline

FIFO Cycle Time

Schnelle Selbstschleife :
N selbst deaktiviert

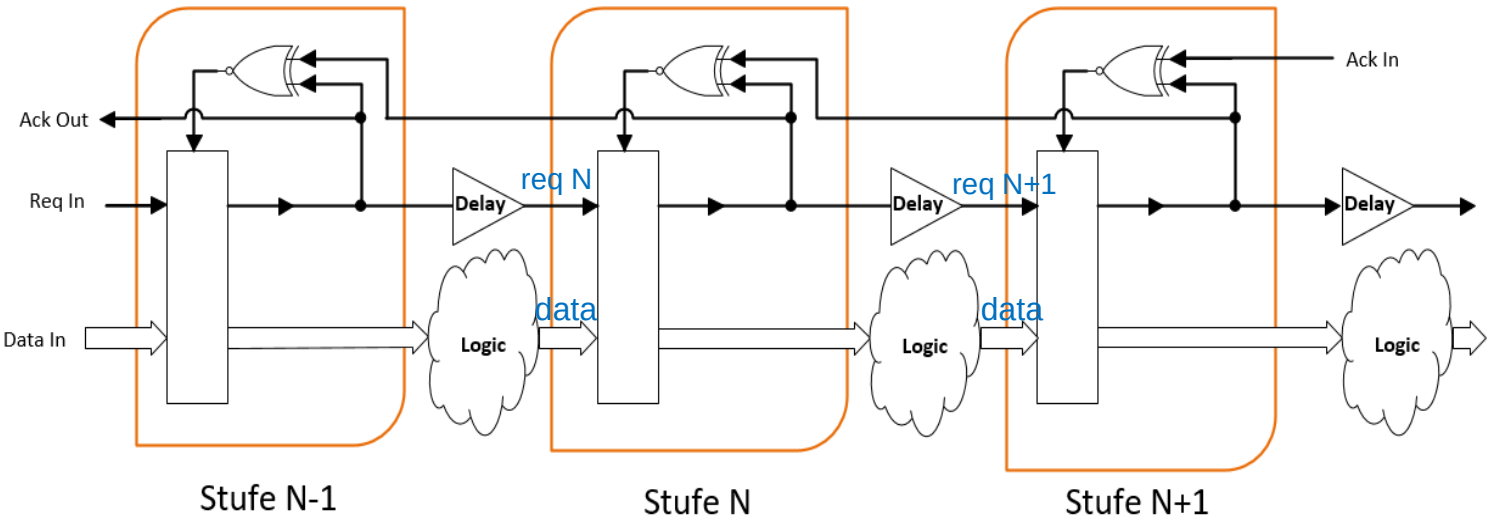


Der Zykluszeit: $T_{Latch\ N} + T_{Latch\ N+1} + T_{XNOR\uparrow}$

Timing : Pipeline with Logic

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

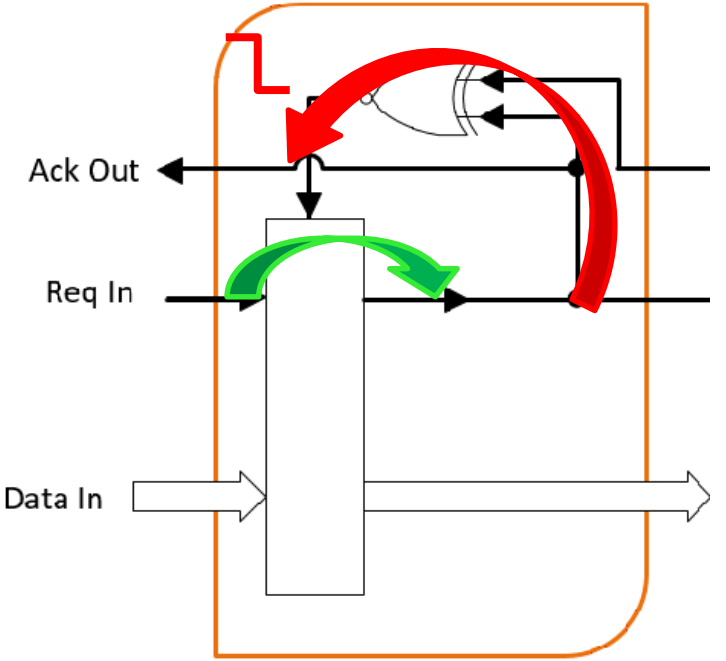
Einfache Hinzufügen : Logikblock + Passende Verzögerung



Die Zykluszeit: 2xT_{Latch} + **T_{Logic}** + T_{XNOR}↑

Timing

Setup Time

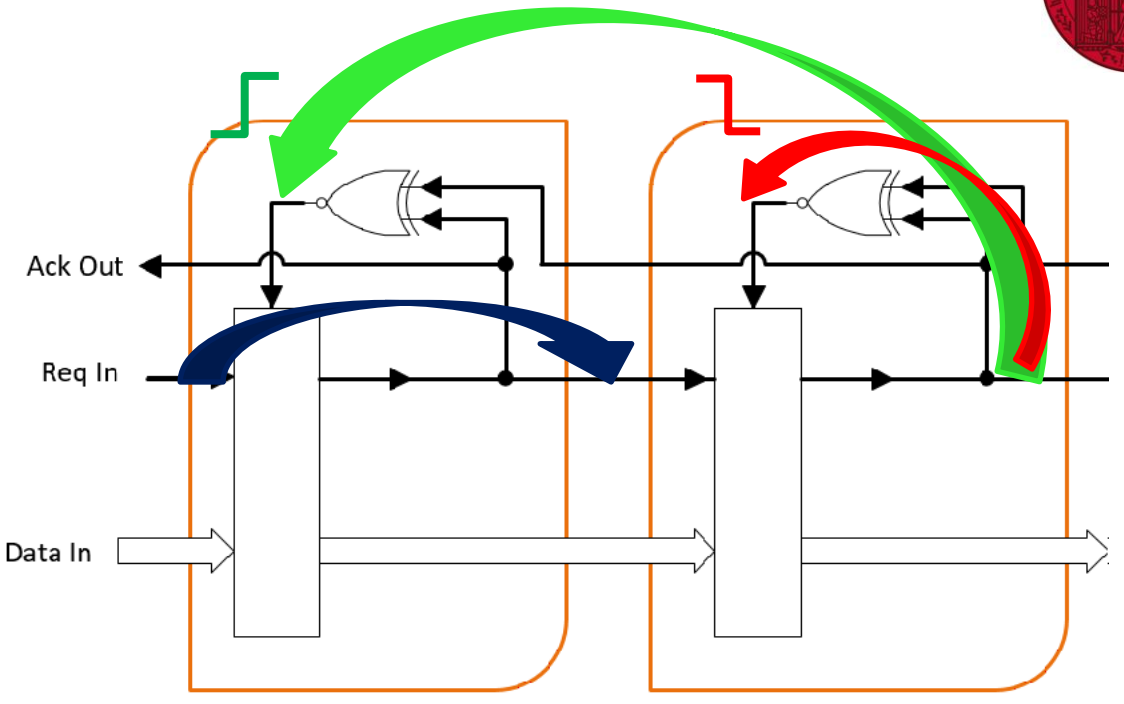


Setup Time:

$$T_{reqN \rightarrow doneN} + T_{XNOR\ N\downarrow} > T_{setup}$$

Timing

Hold Time



Hold Time:

$$T_{XNOR\ N-1\uparrow} + T_{Latch\ N-1} > T_{XNOR\ N\downarrow} + T_{hold}$$

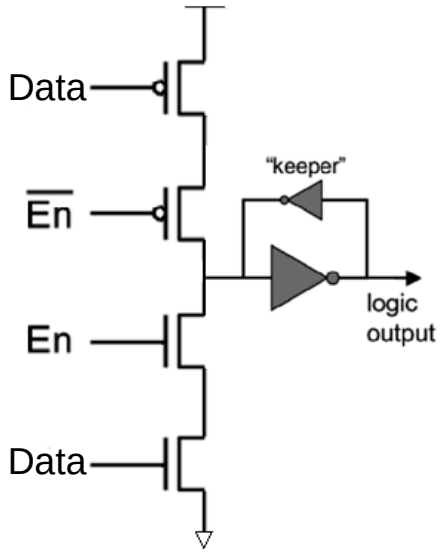
$$\text{If } T_{XNOR\ N-1\uparrow} \approx T_{XNOR\ N\downarrow} \blacktriangleright\blacktriangleright T_{Latch\ N-1} > T_{hold}$$

Einzelheiten

Clocked CMOS (C^2MOS)

Die Vorteile von C^2MOS:

- Kleinere Verzögerung
- Kleinere Fläche
- Geringerer Stromverbrauch

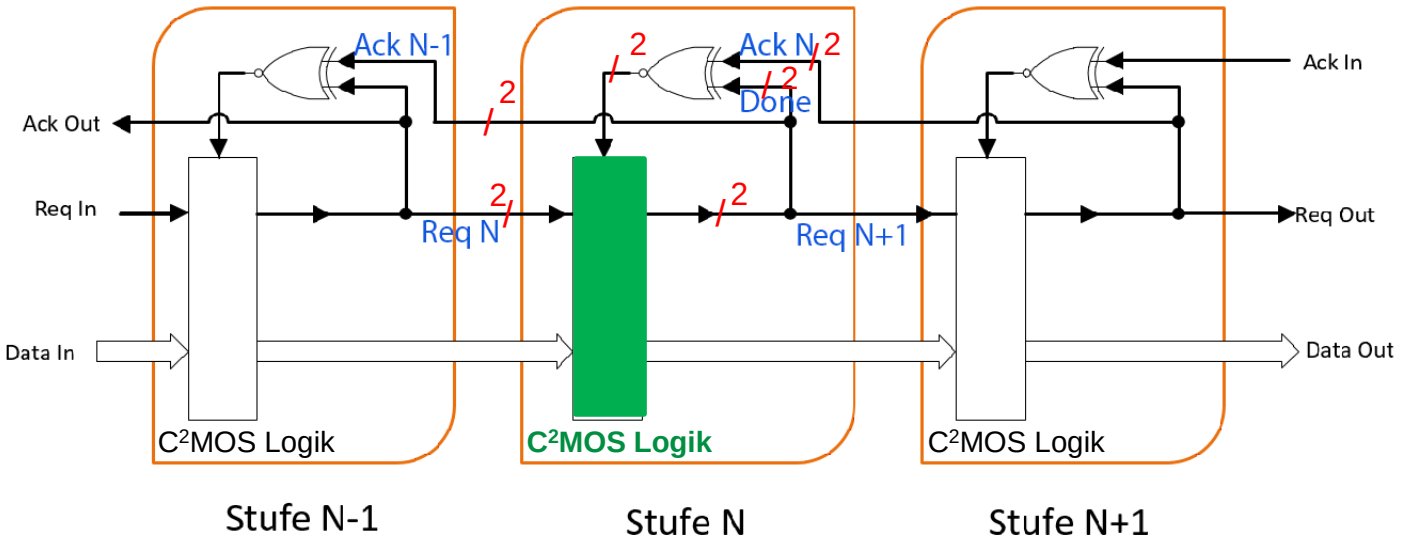


C^2MOS und Logik

Montek Singh & Steven M. Nowick 2001 New York NY10027 IEEE
Ultra-High-Speed Transition-Signaling Asynchronous Pipelines (PPT)

Einzelheiten

Clocked CMOS (C^2MOS) : Latch Kontroller



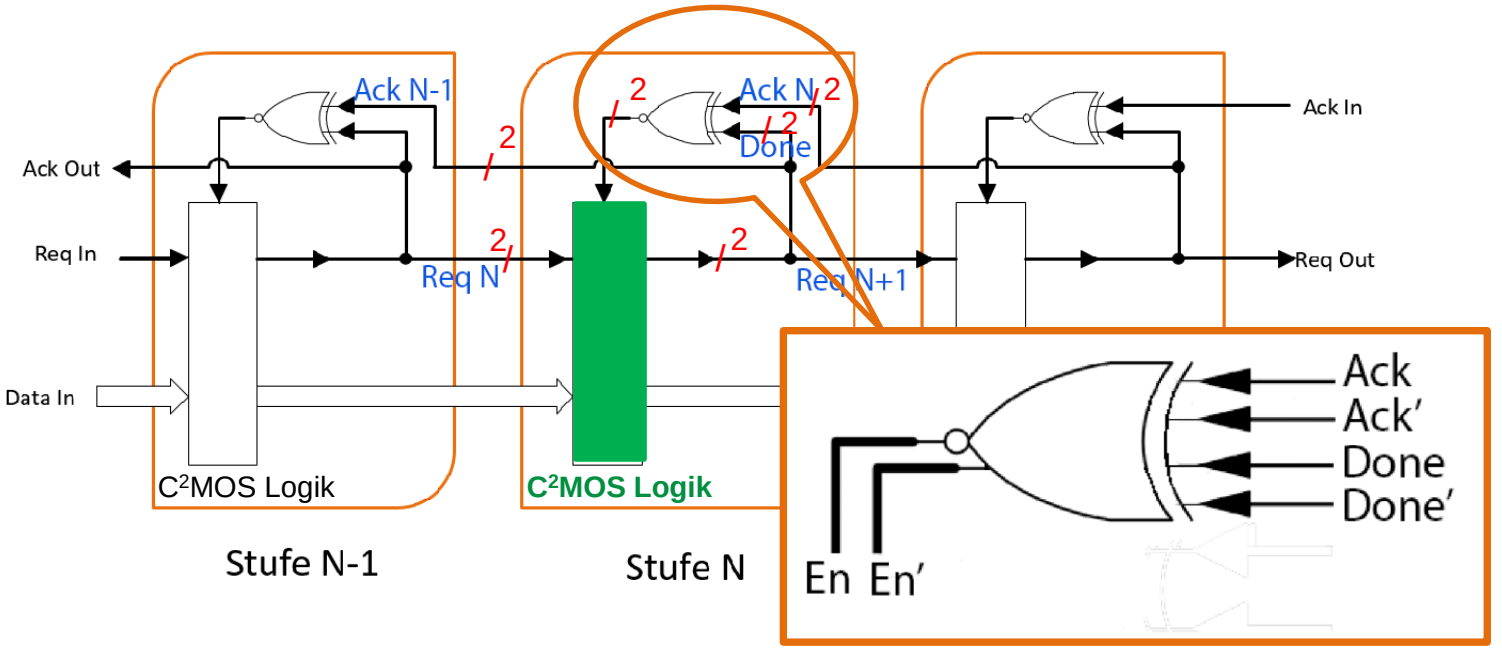
Neue Kontroller-Optimierung : "**Dual-Rail XNOR**"

- eliminiert "**Inverter Delays**" von "Cycle Time"

Die Zykluszeit: $2xT_{C^2MOS} + T_{XNOR\uparrow}$

Einzelheiten

Clocked CMOS (C^2MOS) : Latch Kontroller



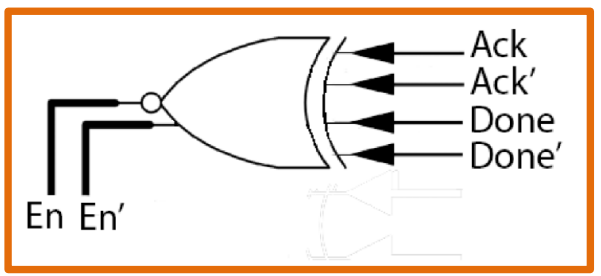
Neue Kontroller-Optimierung : "**Dual-Rail XNOR**"

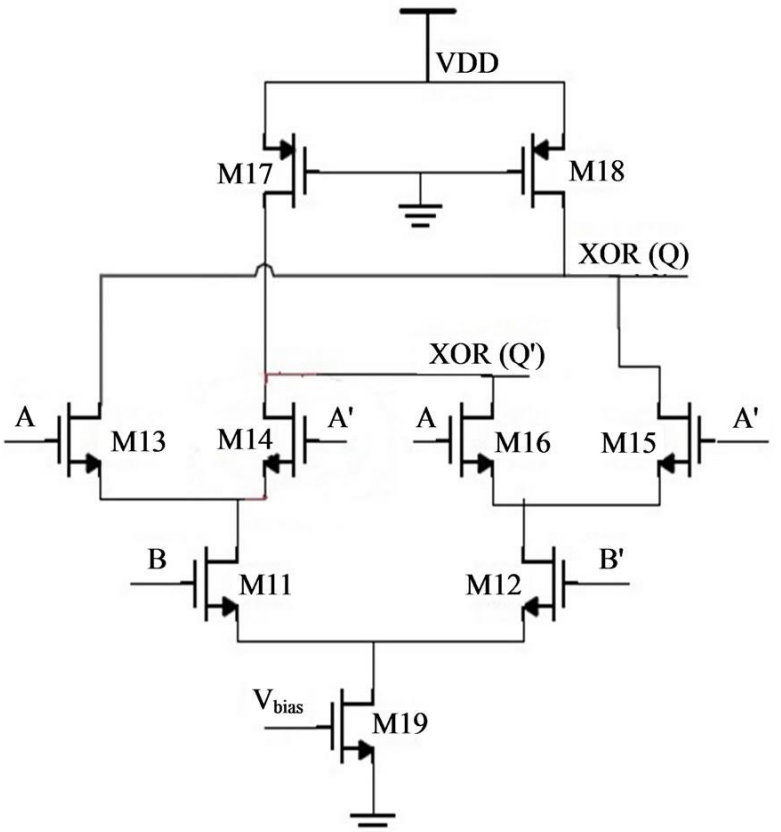
- eliminiert "**Inverter Delays**" von "Cycle Time"

Die Zykluszeit: $2xT_{C^2MOS} + T_{XNOR\uparrow}$

Einzelheiten

Clocked CMOS (C^2MOS) : Latch Kontroller-Schaltung





Current Mode Logic Testing of XOR/XNOR Circuit: A Case Study
http://dx.doi.org/10.4236/cs.2013.44049

Einzelheiten

Timing-Optimierung mit XNOR

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Der Zykluszeit: $2xT_{Latch} + T_{Logic} + T_{XNOR\uparrow}$

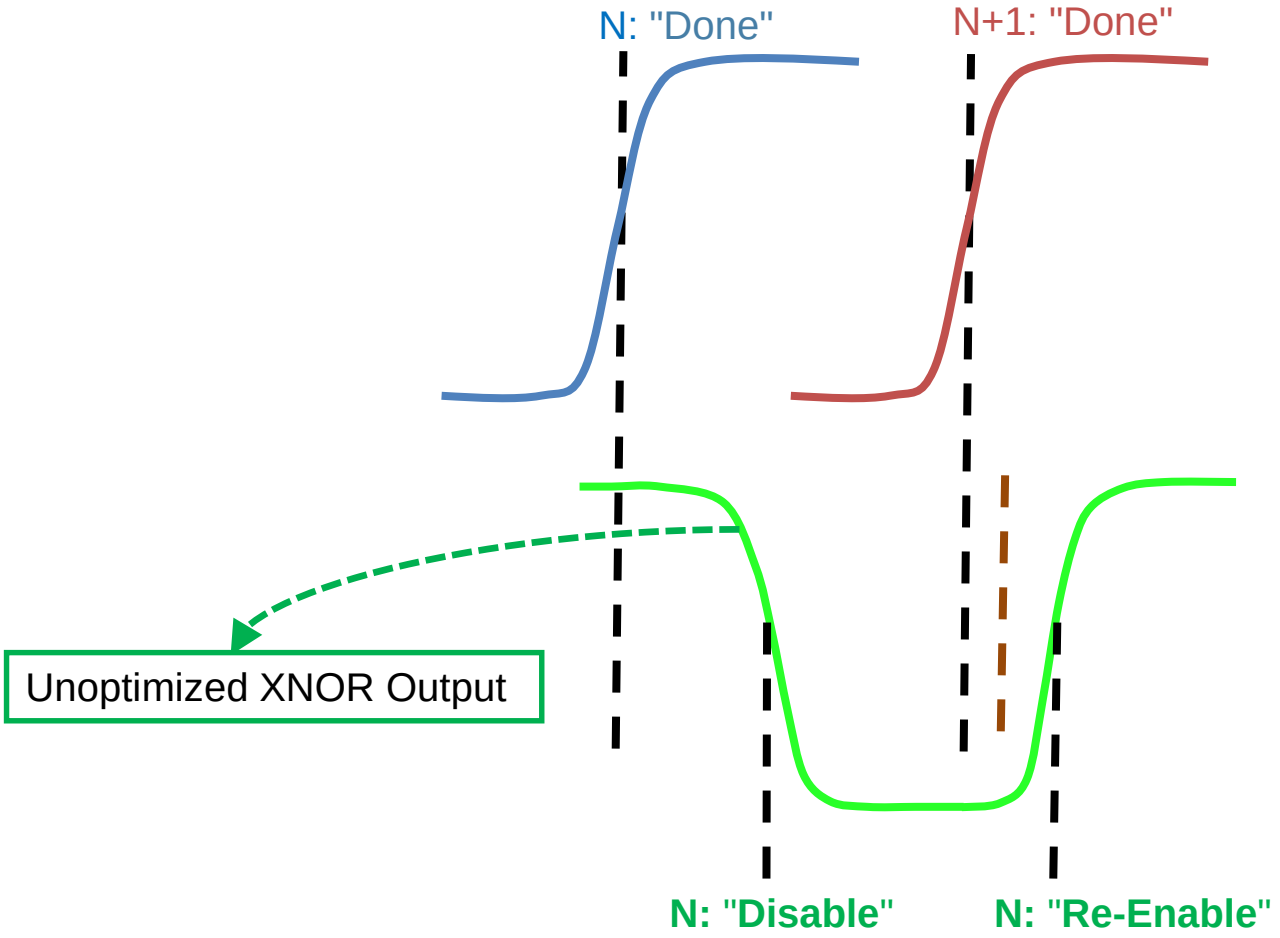
Das Ziel: früh T_{XNOR}

Die Observation:
XNOR wird zweimal duch pro Datenelement gewechselt: $T_{XNOR\downarrow}$ und $T_{XNOR\uparrow}$
- nur 2. Übergang ist entscheidend für die Leistung :$T_{XNOR\uparrow}$

Lösung: Reduzierung XNOR Ausgangsschwingung
- schnellere Anstiegszeit

Einzelheiten

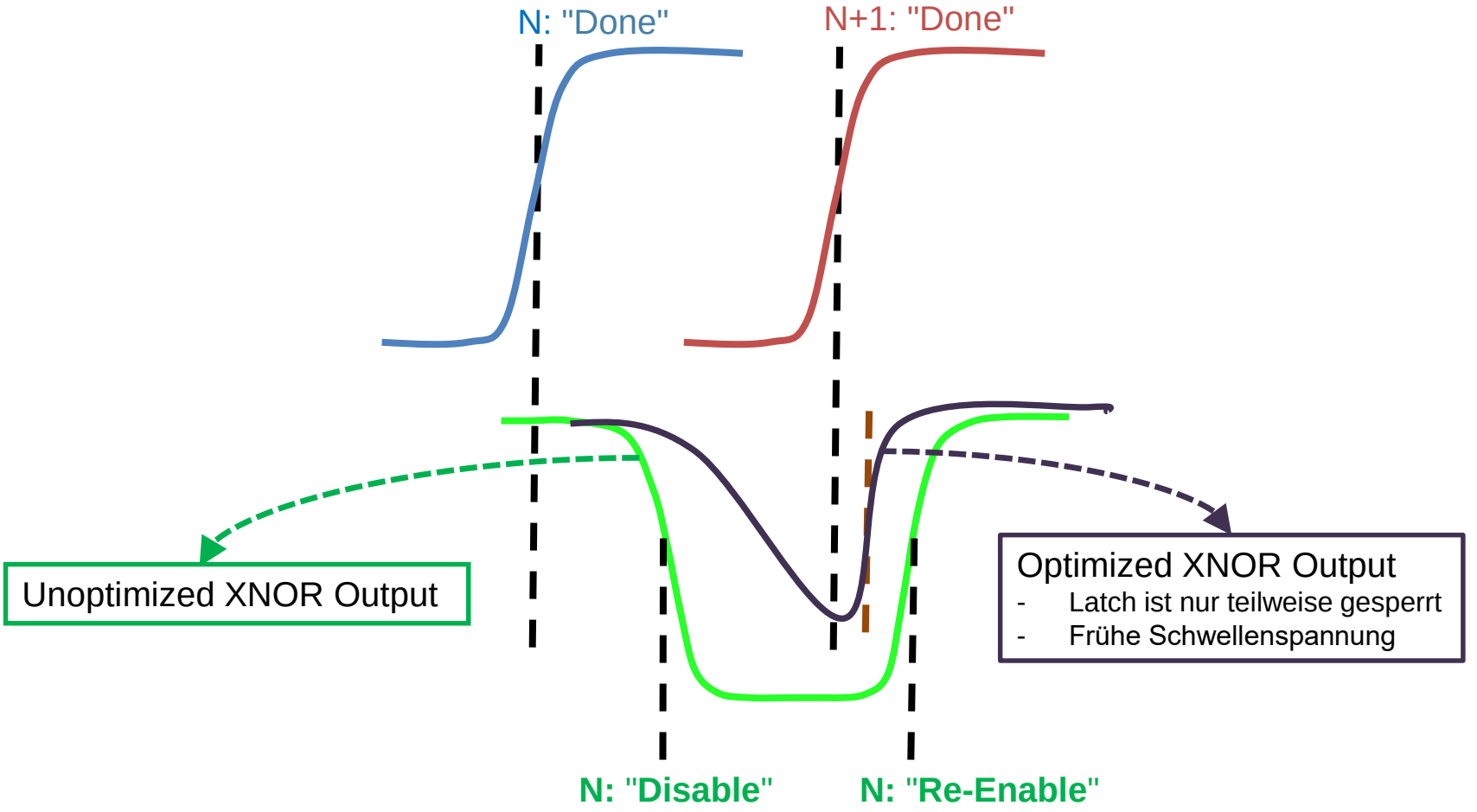
Timing-Optimierung mit XNOR



Montek Singh, Steven M. Nowick, MOUSETRAP: High-Speed Transition-Signaling Asynchronous Pipelines, p.684 -698 , VOL. 15, NO. 6, JUNE 2007

Einzelheiten

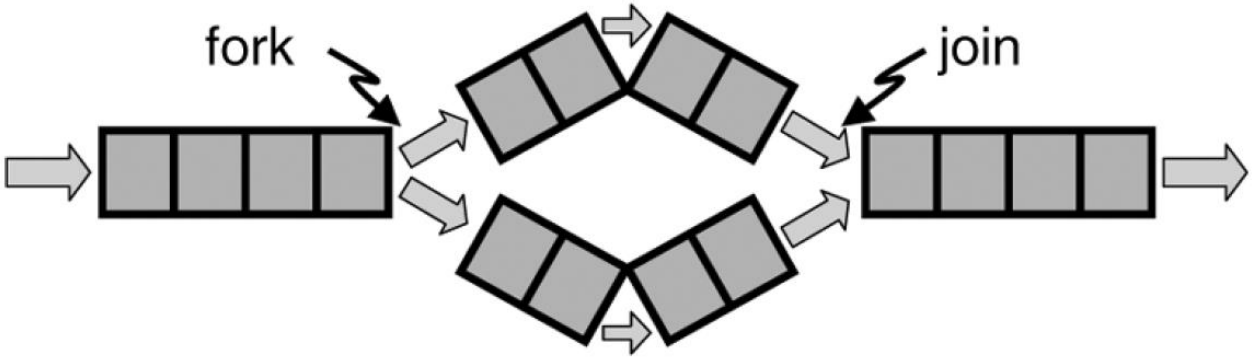
Timing-Optimierung mit XNOR



Montek Singh, Steven M. Nowick, MOUSETRAP: High-Speed Transition-Signaling Asynchronous Pipelines, p.684 -698 , VOL. 15, NO. 6, JUNE 2007

Einzelheiten

Komplexe Pipeline: Forks and Joins



Non-Linear Pipelining

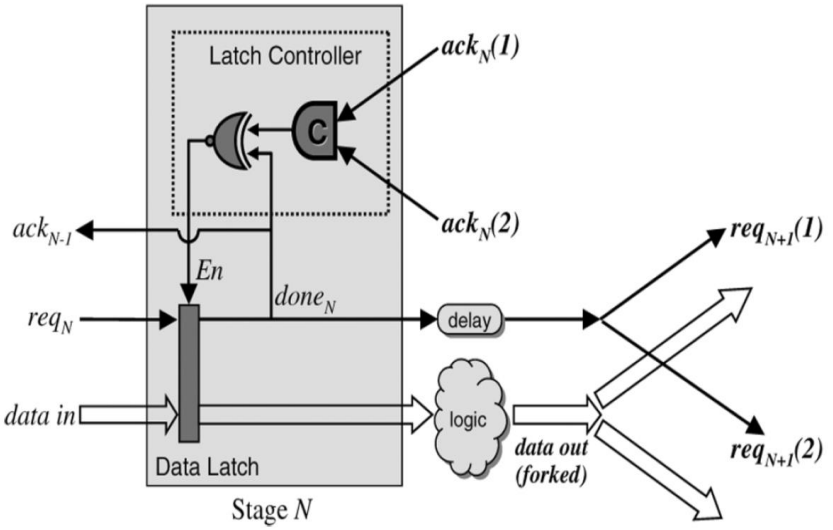
Montek Singh, Steven M. Nowick, MOUSETRAP: High-Speed Transition-Signaling Asynchronous Pipelines, p.684 -698 , VOL. 15, NO. 6, JUNE 2007

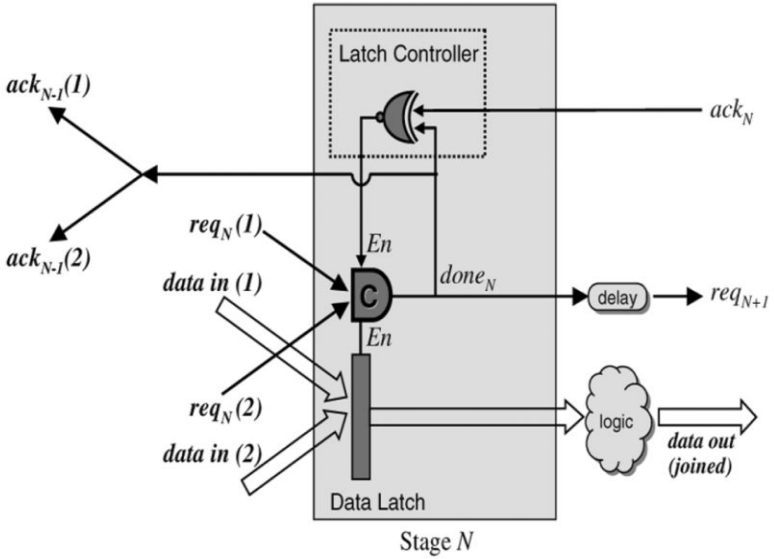
Beitrag: Einführung effiziente Schaltungsstrukturen

- **Fork:** Daten-Verteilung + Steuerung zu verschiedene Ziele
- **Join:** Daten-Zusammenführung + Steuerung aus mehreren Quellen

Einzelheiten

Komplexe Pipeline: Forks and Joins





Fork: Zusammenführung mehrerer "ack"

T_{Fork}: $2xT_{Latch} + T_{Logic} + T_C + T_{XNOR}\uparrow$

Join: Zusammenführung mehrerer "req"

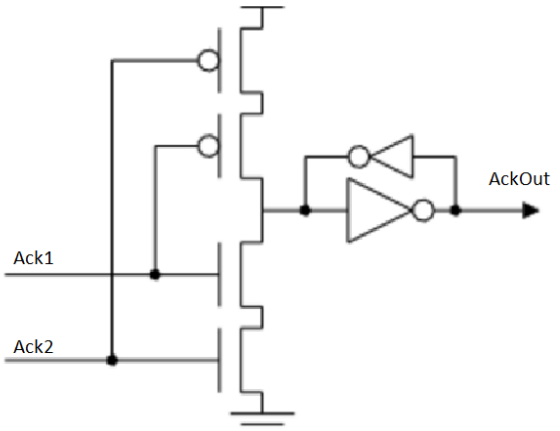
T_{Join}: $T_{aC} + T_{Logic} + T_{Latch} + T_{XNOR}\uparrow$

$$T_{Fork} > T_{Join}$$

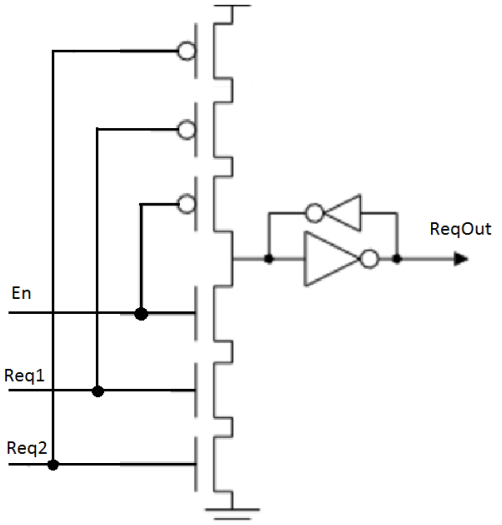
Einzelheiten

Komplexe Pipeline: Forks and Joins

A	B	Q
0	0	0
0	1	-
1	0	-
1	1	1



C-element für Ack



C-element für Req

En	A	B	Q
1	0	0	0
1	0	1	-
1	1	0	-
1	1	1	1
0	x	x	-

Fork: Zusammenführung mehrerer "ack"

T_{Fork}: $2xT_{Latch} + T_{Logic} + T_C + T_{XNOR}\uparrow$

Join: Zusammenführung mehrerer "req"

T_{Join}: $T_{aC} + T_{Logic} + T_{Latch} + T_{XNOR}\uparrow$

$$T_{Fork} > T_{Join}$$

Fazit

Speed Test

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

MOUSETRAP FIFO (250nm TSMC) Pre-Layout Simulation (10 Stufe + 16 bit breit)

Pipeline Design	Latch delay Tlatch (ps)	TXNOR↑ (ps)	TXNOR↓ (ps)	Cycle Time	(ps)	Throughput (GHz)
Mousetrap	110	65	63	2.Tlatch+TXNOR↑	285	3.51

MOUSETRAP FIFO (180nm TSMC) Post-Layout Simulation (10 Stufe + 16 bit breit)

Pipeline Design	Latch delay Tlatch (ps)	TXNOR↑ (ps)	TXNOR↓ (ps)	Cycle Time	(ps)	Throughput (GHz)
Mousetrap	188	102	115	2.Tlatch+TXNOR↑	477	2.1
Mousetrap (Opt)	179	63	131	2.Tlatch+TXNOR↑	421	2.38

Fazit

Leider kein fairer Vergleich in Literatur. Nur unterschiedliche Technologien vergleichen. Nicht überzeugend. Sie zeigen trotzdem die Aussagen der Papers.

Fazit

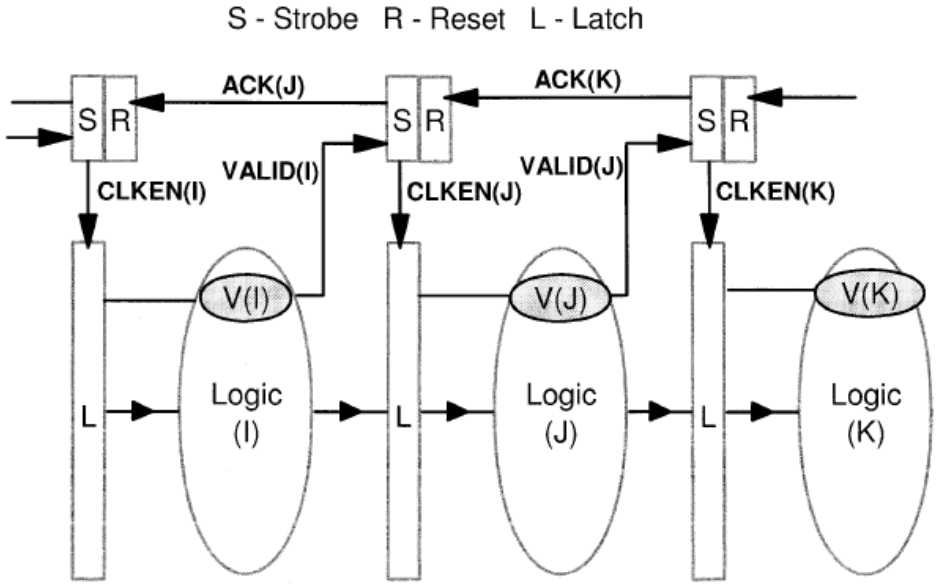
Geschwindigkeitsvergleich mit IPCMOS

IPCMOS (asynchronous interlocked pipelined CMOS)

- 3.3~4.5GHz
- IBM 180nm 1.5V (silicon-on-insulator)
- Post-layout simulation

MOUSETRAP

- 2.1~2.38GHz
- TSMC 180nm
- Post-layout simulation



Interlocked **P**ipelined **CMOS**

S. E. Schuster, *P.W.Cook;* Low-Power Synchronous-to-Asynchronousto- Synchronous Interlocked Pipelined CMOS Circuits Operating at 3.3–4.5 GHz ; VOL. 38, NO. 4, APRIL 2003

Fazit

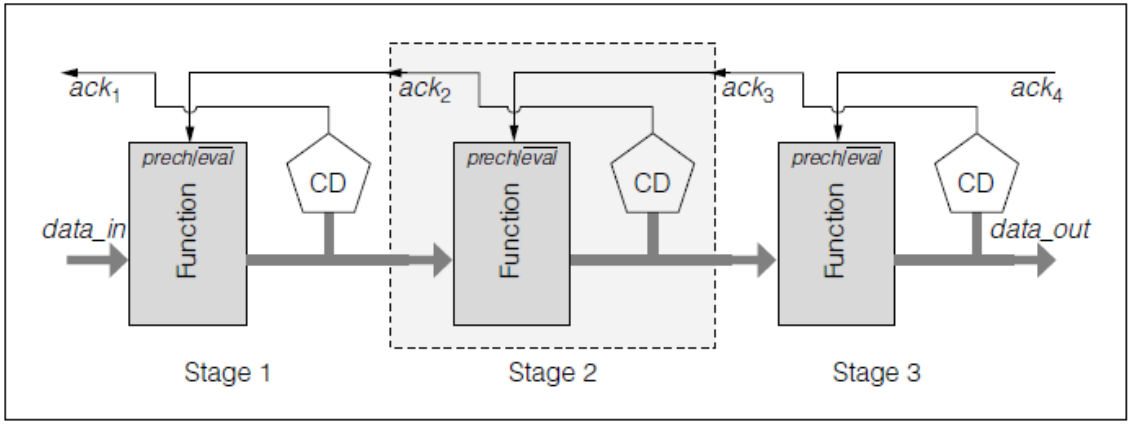
Mousetrap-Energieverbrauch mit PS0

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Power und Energieverbrauch von 10 Stufe FIFO

Design	Throughput (GHz)	Power (mW)	Energy/item/stage (pJ)
MOUSETRAP	2.1	30.8	1.49
PS0	0.51	26.3	5.20

PS0 : Dual-Rail dynamische asynchrone Pipeline (1990)



S. M. Nowick, M. Singh: High-Performance Asynchronous Pipelines: An Overview, Design & Test of Computers, IEEE vol.28, iss: 5 , p.8-22, Oct. 2011

Fazit

Mousetrap-Energieverbrauch mit PS0

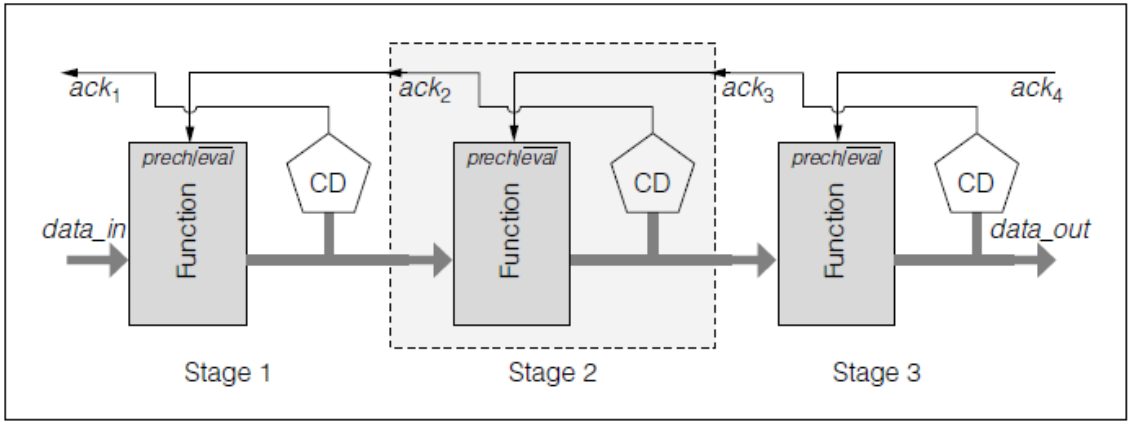
UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Power und Energieverbrauch von 10 Stufe FIFO

Design	Throughput (GHz)	Power (mW)	Energy/item/stage (pJ)
MOUSETRAP	2.1	30.8	1.49
PS0	0.51	26.3	5.20

%71 Energieeffizient

PS0 : Dual-Rail dynamische asynchrone Pipeline (1990)



S. M. Nowick, M. Singh: High-Performance Asynchronous Pipelines: An Overview, Design & Test of Computers, IEEE vol.28, iss: 5 , p.8-22, Oct. 2011