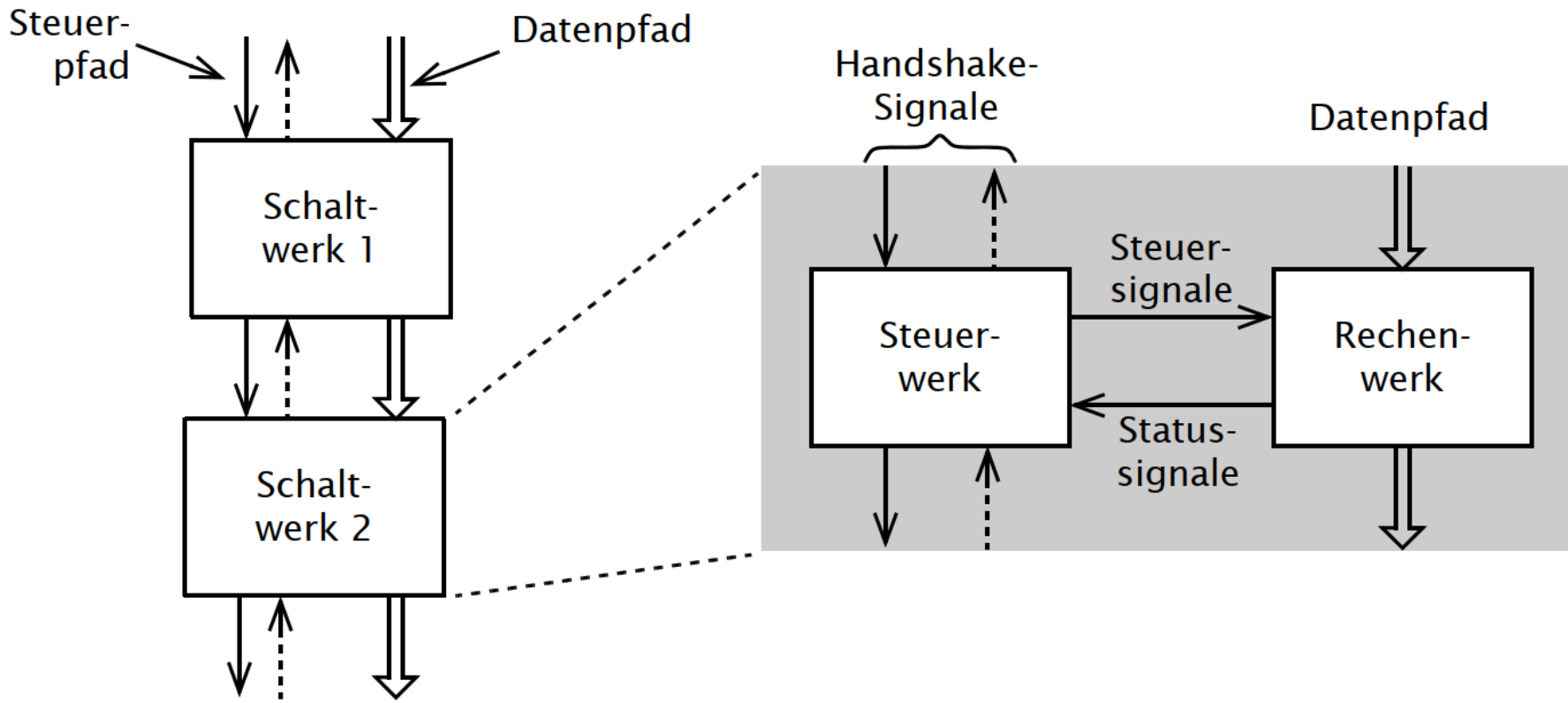
Synchronisation mit Handshaking

- Schaltwerksentwurf mit Synchronisation
  - Die Komplexität moderner digitaler Systemen lässt sich heute nur durch den Einsatz der Modularisierung und der Hierarchiebildung im Entwurf beherrschen.
  - Diese Lösungsansatz impliziert einen Entwurf digitaler Systeme auf der Grundlage von kooperierenden Schaltwerken.
  - Der Kooperation zwischen Schaltwerken liegt ein Regelwerk zugrunde, nach dem der Informationsaustausch nur mit Hilfe von Synchronisations-signalen (sog. Handshake-Signalen) stattfinden kann.

- Die Notwendigkeit für Handshaking resultiert daraus, dass
  - Schaltwerke mit unterschiedlichen Taktdomänen arbeiten,
  - Schaltwerke unterschiedlich schnell arbeiten (Producer/Consumer),
  - Schaltwerke nicht in jedem Takt, sondern zu sporadischen Zeitpunkten Daten liefern.

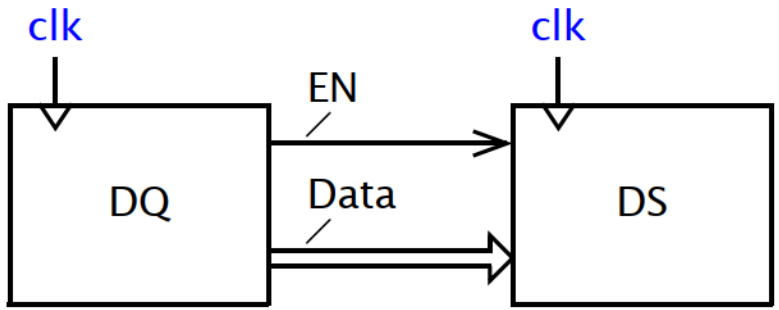
Synchronisation mit Handshaking

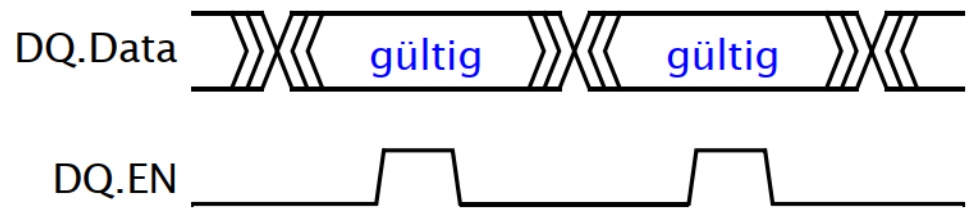
- Prinzipieller Systemaufbau mit kooperierenden (verketteten) Schalt-werken



Synchronisation mit Handshaking

- Ein-Signal-Handshaking
  - An diesem Synchronisationsprinzip ist nur ein Hadshake-Signal beteiligt, allerdings mit einem gemeinsamen Takt.
  - Die Datenquelle gibt ein Datum auf dem Datenbus aus und nach einer optionalen Verzögerung zeigt sie dessen Gültigkeit mit dem Aktivieren eines Valid-Signals (z.B. Enable/Ready/Done) an.
  - Dieses Verfahren funktioniert nur dann fehlerfrei, wenn die Datensenke ankommende Daten schneller verarbeiten bzw. weitergeben kann, als die Datenquelle sie liefert.
  - Häufig eingesetztes Verfahren in fließbandorganisierten Systemen.





Synchronisation mit Handshaking

- Zwei-Signal-Handshaking:
  - (vollverzahntes Handshaking, Vierflanken-Handshaking)
  - Datenquelle (DQ, Sender) ---> Datensenke (DS, Empfänger)
  - Der Datentransfer erfolgt unidirektional von der Datenquelle zur
  - Datensenke
  - In der Abhängigkeit davon, wer den Datentransfer initiiert, unterscheidet man zwischen zwei Fällen, und zwar spricht von einem DQ-initiierten oder dem DS-initiierten Transfer.
  - In beiden Fällen dürfen sich die an der Synchronisation beteiligten Signalflanke nicht überholen!

  - Hinweis: Die Kreativität der Entwickler kennt keine Grenzen, weshalb man oft in der technischen Literatur immer wieder unterschiedliche Namen für die Handshake-Signale findet, die aber in Wirklichkeit das gleiche machen.

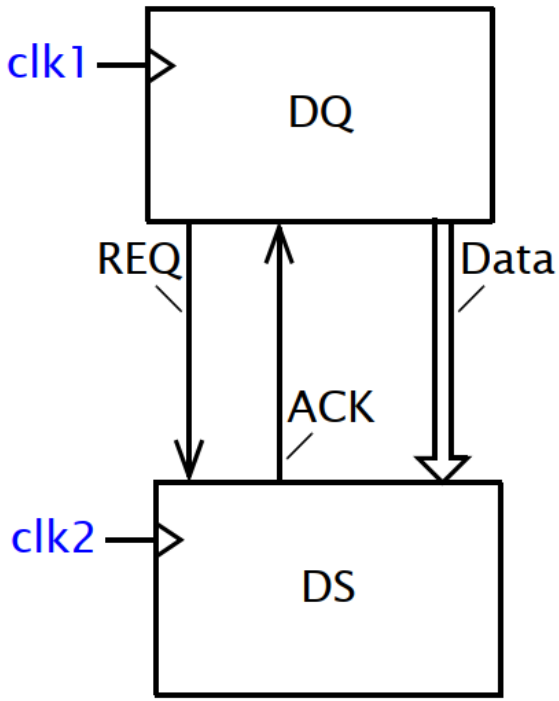
Synchronisation mit Handshaking

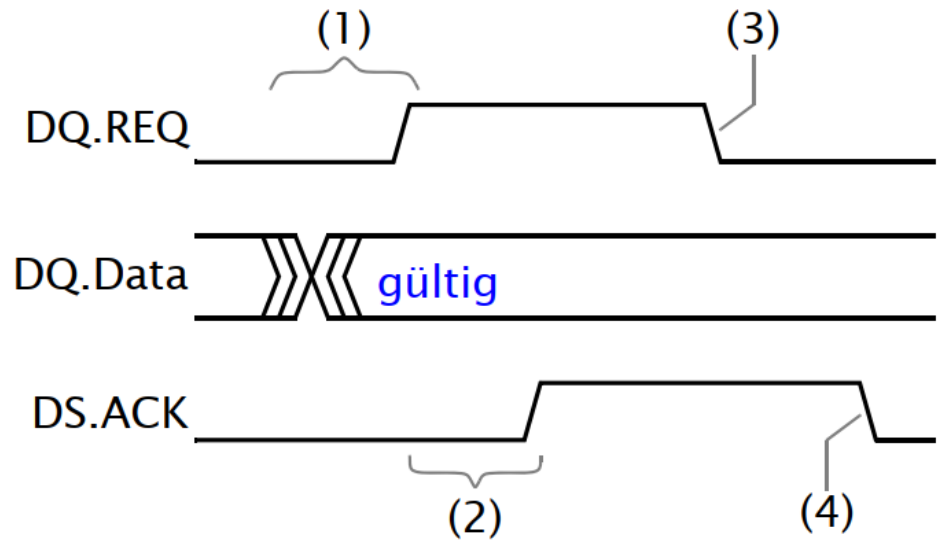
- Der DQ-initiierte Transfer:
  - Am DQ-initiierten Transfer sind zwei Handshake-Signale beteiligt:
    - ausgehend von der Datenquelle REQ (request)
    - ausgehend von der Datensenke ACK (acknowledge)

- Der Ablauf:
  1. Die Datenquelle gibt ein Datum auf dem Datenbus aus und nach einer optionalen Verzögerung (wegen Signallaufzeiten auf dem Bus oder undefinierten Signalwerten) zeigt sie dessen Gültigkeit mit dem Aktivieren (z.B. mit der steigenden Flanke) des Signals REQ an.
  2. Die Datensenke übernimmt das Datum und zeigt mit dem Aktivieren (z.B. mit der steigenden Flanke) des Signals ACK an, dass sie für weiteren Empfang nicht bereit ist.
  3. Darauf hin deaktiviert die Datenquelle das Signal REQ.
  4. Sobald die Datensenke wieder empfangsbereit ist, zeigt sie ihre Bereitschaft durch das Deaktivieren des Signals ACK an.

Synchronisation mit Handshaking

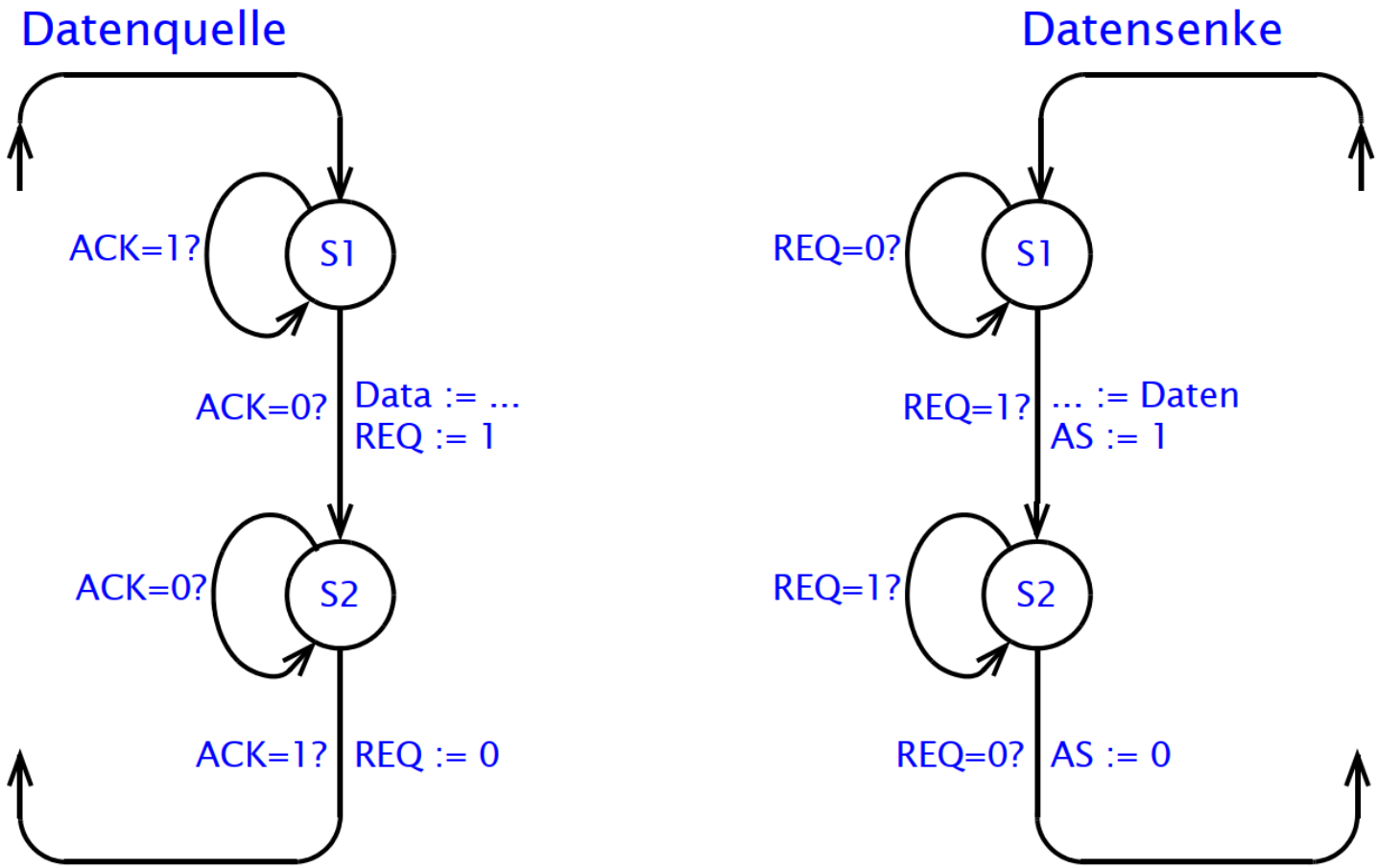
- Der DQ-initiierte Transfer:





Synchronisation mit Handshaking

- Der DQ-initiierte Transfer:



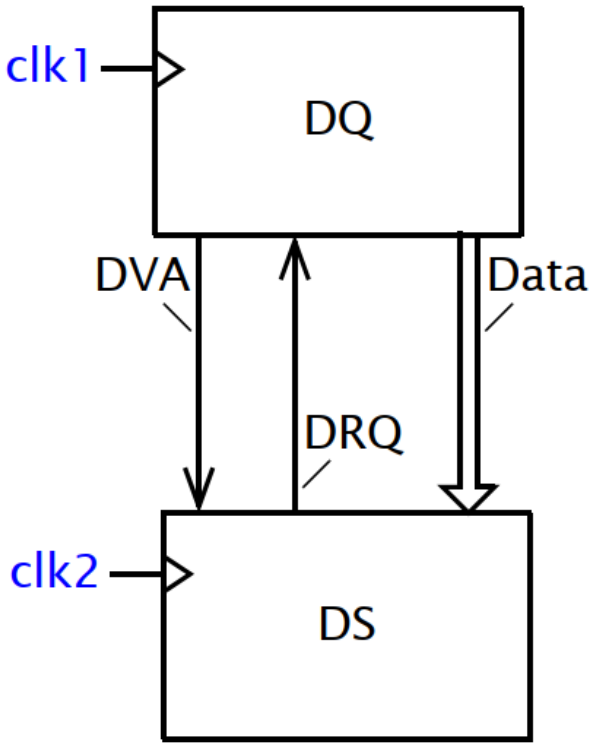
Synchronisation mit Handshaking

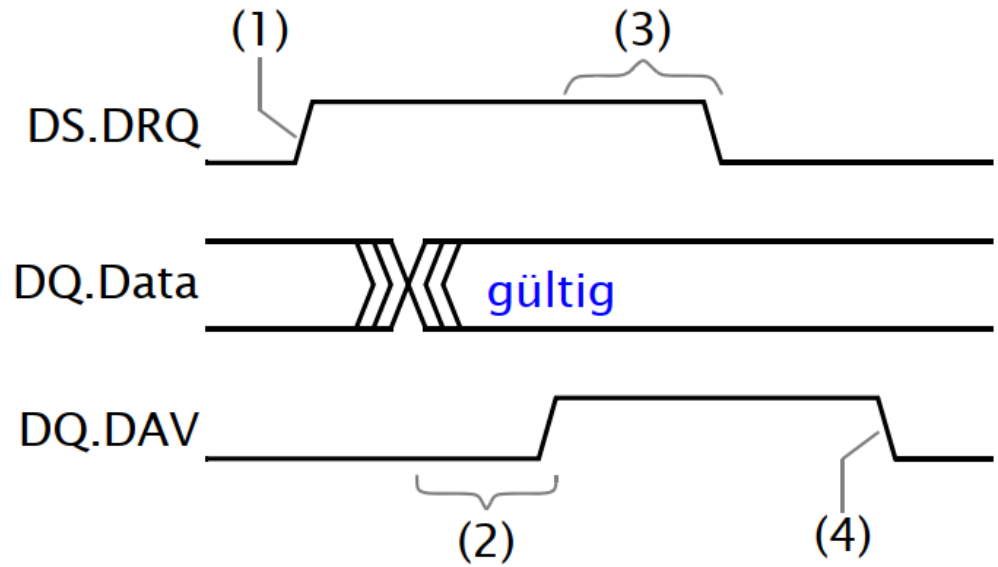
- Der DS-initiierte Transfer:
  - Am DS-initiierten Transfer sind zwei Handshake-Signale beteiligt:
    - ausgehend von der Datensenke DRQ (data request) für Datenanforderung
    - ausgehend von der Datenquelle DAV (data valid) für Datenbestätigung.

- Der Ablauf:
  1. Die Datensenke fordert mit dem Aktivieren (z.B. mit der steigenden Flanke) des Signals DRQ ein neues Datum an.
  2. Daraufhin gibt die Datenquelle ein Datum aus und zeigt dessen Gültigkeit mit dem Aktivieren (z.B. mit der steigenden Flanke) des Signals DAV an.
  3. Die Datensenke übernimmt das Datum und zeigt ihrerseits diesen Zustand mit dem Deaktivieren des Signals DRQ an.
  4. Daraufhin deaktiviert auch die Datenquelle ihrerseits das Signal DAV, womit sie anzeigt, dass momentan keine gültige Daten vorliegen.

Synchronisation mit Handshaking

- Der DS-initiierte Transfer:



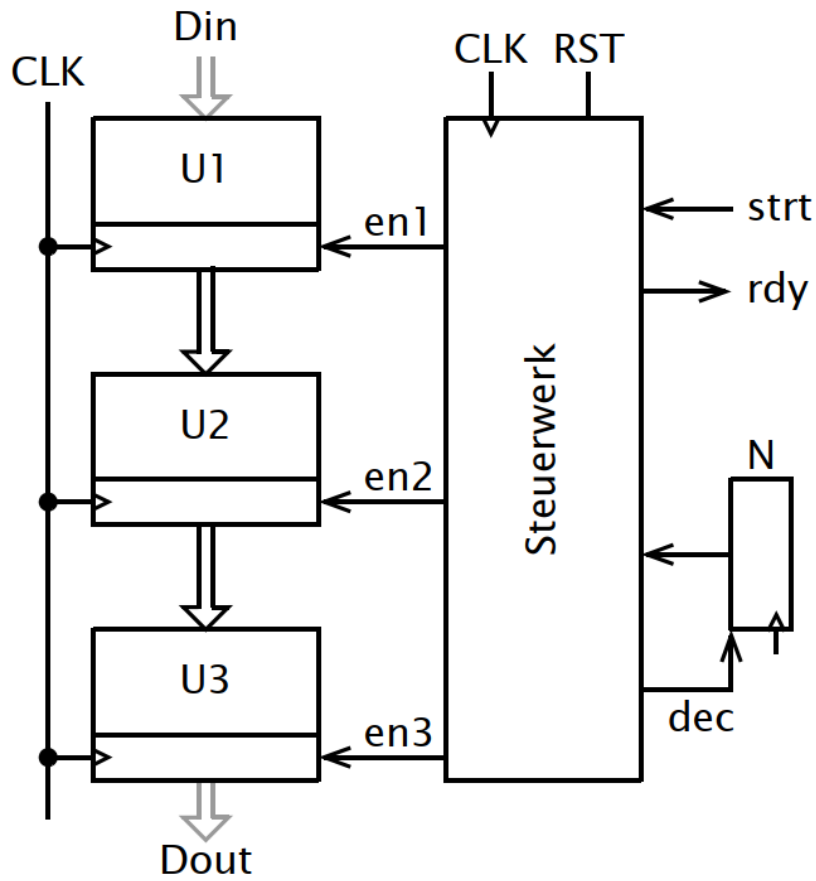


Synchronisation mit Handshaking

- Design-Regeln:
  - Schaltwerke/Steuerwerke werden in VHDL mit Hilfe von Prozessen mit CASE- und IF-Anweisungen modelliert.
  - Kooperierende Schaltwerke/Steuerwerke kommunizieren mit Hilfe von Handshake-Signalen miteinander.
  - In komplexen digitalen Systemen ist eine sorgfältige Planung von Handshake-Mechanismen erforderlich. Dadurch lassen sich oft überflüssige FIFO-Strukturen vermeiden.
  - Wenn keine zuverlässigen Aussagen über das Laufzeitverhalten kooperierender Schaltwerke gemacht werden können, oder wenn der Einsatz solcher Schaltwerke in unterschiedlichen Umgebungen vorgesehen ist, ist es empfehlenswert, das Vierflanken-Handshaking einzusetzen.
  - FIFO-Strukturen sind nur dann notwendig, wenn ankommende Daten nicht sofort, sondern erst nachdem z.B. ein Datenpaket vollständig empfangen ist, weiter verarbeitet werden können.

Synchronisation mit Pipelining

- Rechenwerke mit fließbandorganisierten Funktionseinheiten



	U1	U2	U3
T_0	X		
T_1	X	X	
T_2	X	X	X
. . .	X	X	X
T_N	X	X	X
T_{N+1}		X	X
T_{N+2}			X

Synchronisation mit Pipelining

- Rechenwerke mit fließbandorganisierten Funktionseinheiten

N=1

	U1	U2	U3
S0			
S1	X		
S6		X	
S5			X
S0			
S0			

N=2

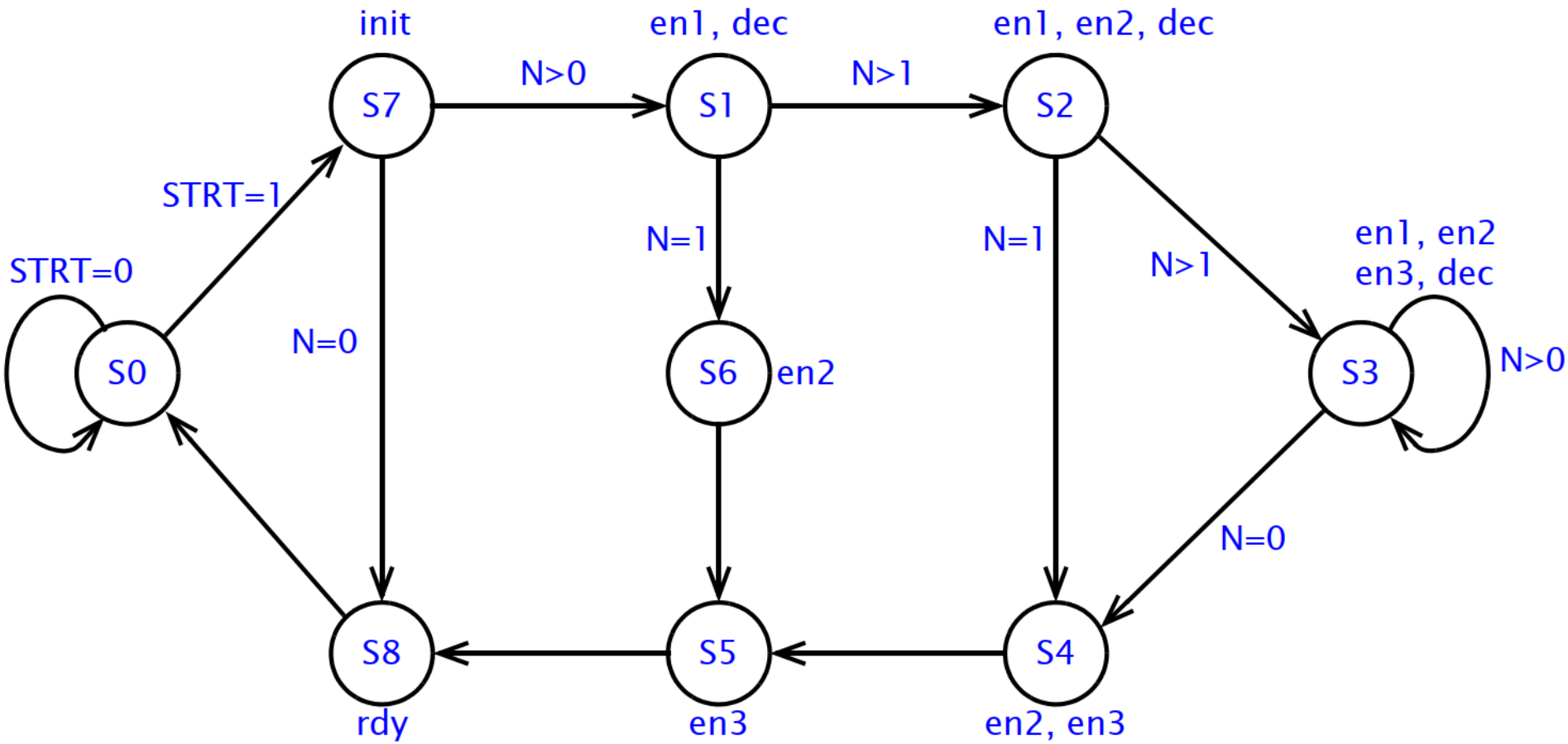
	U1	U2	U3
S0			
S1	X		
S2	X	X	
S4		X	X
S5			X
S0			

N≥3

	U1	U2	U3
S0			
S1	X		
S2	X	X	
S3	X	X	X
S4		X	X
S5			X

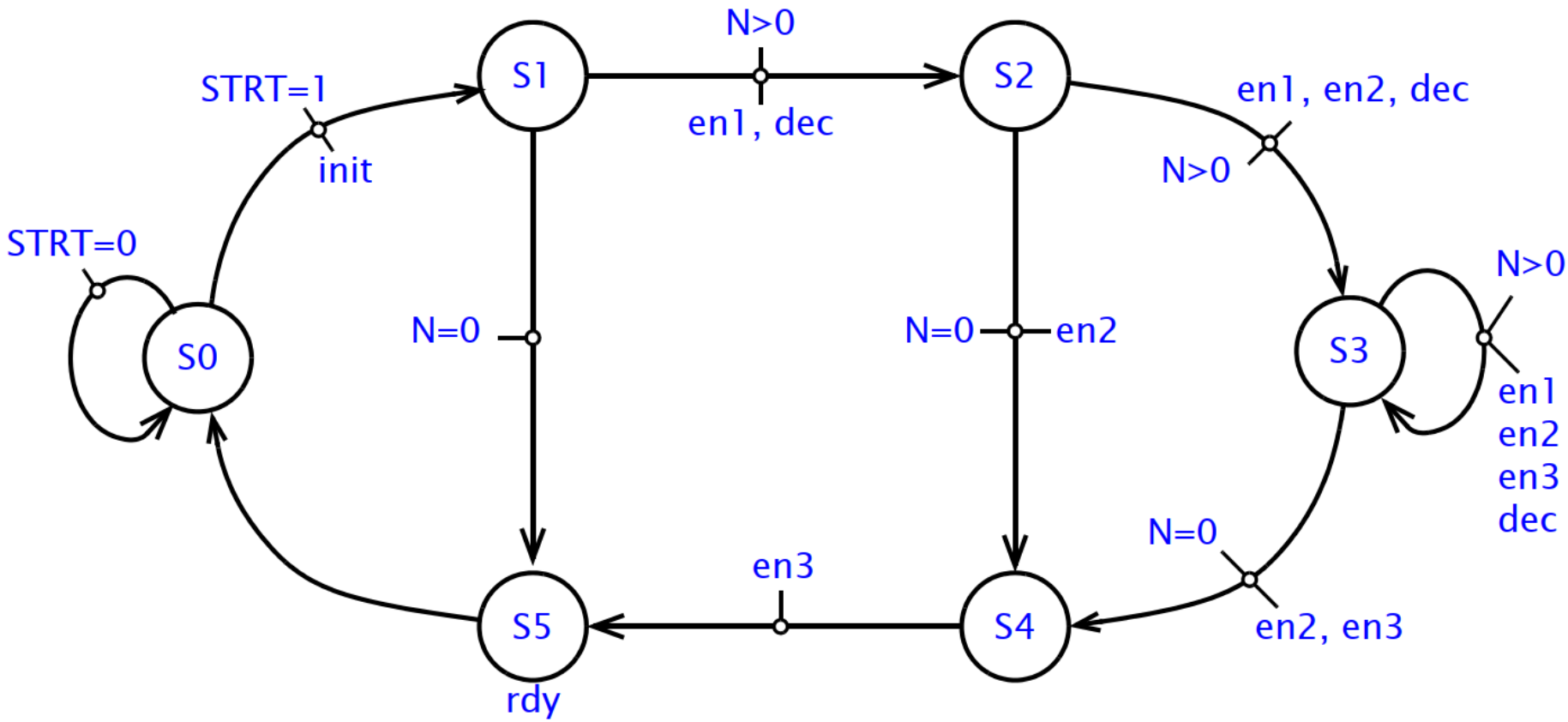
Synchronisation mit Pipelining

- Zustandsgraph eines Moore-Steuerwerks für ein Rechenwerk mit fließbandorganisierten Funktionseinheiten



Synchronisation mit Pipelining

- Zustandsgraph eines Mealy-Steuerwerks für ein Rechenwerk mit fließbandorganisierten Funktionseinheiten



Synchronisation mit Pipelining

- Steuertabelle für das Mealy-Steuerwerk für ein Rechenwerk mit fließbandorganisierten Funktionseinheiten

S	strt	N	S^+	init	dec	en1	en2	en3	rdy
S0	0	-	S0	0	0	0	0	0	0
S0	1	-	S1	1	0	0	0	0	0
S1	-	=0	S5	0	0	0	0	0	0
S1	-	>0	S2	0	1	1	0	0	0
S2	-	=0	S4	0	0	0	1	0	0
S2	-	>0	S3	0	1	1	1	0	0
S3	-	=0	S4	0	0	0	1	1	0
S3	-	>0	S3	0	1	1	1	1	0
S4	-	-	S5	0	0	0	0	1	0
S5	-	-	S0	0	0	0	0	0	1