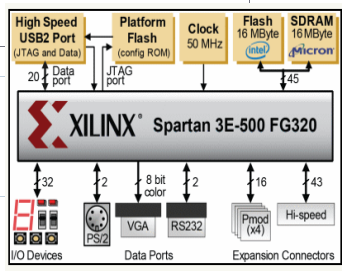


**Computer Engineering**  
**WS 2012**

**Digitale Systeme**  
**μC - FPGA**

**Prof. Dr. B. Schwarz**





## Synchronisation



### Einleitung: μC – FPGA Kommunikation

#### Asynchrone Eingangssignale

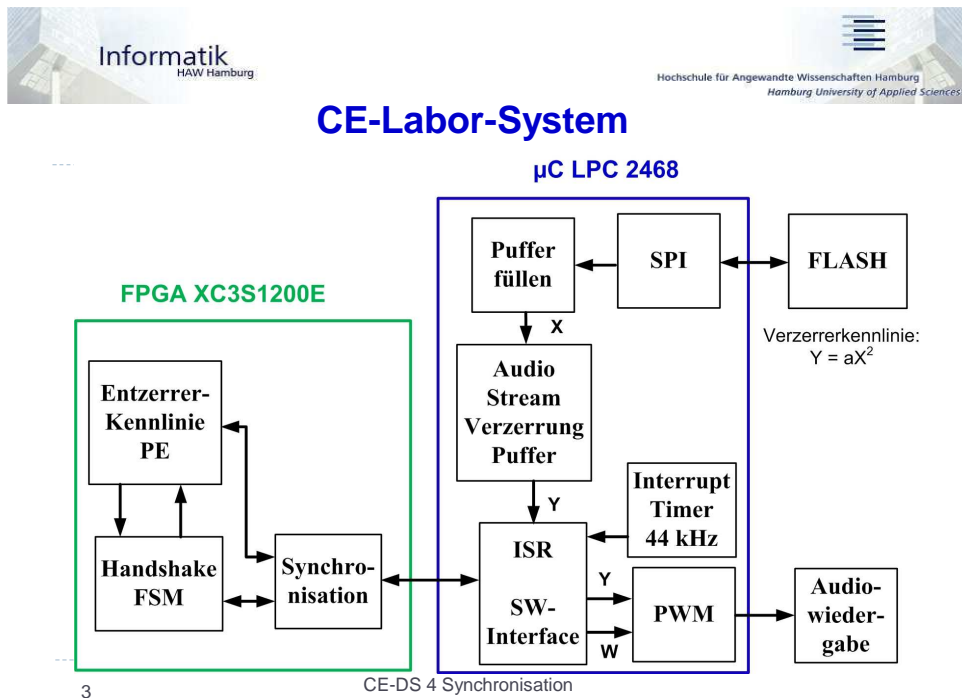
#### MTBF bei Flipflops mit metastabilen Zuständen

#### Synchronisationsschaltung für lange Impulse

#### Synchronisationsschaltung für kurze Impulse

### Kommunikation zwischen asynchronen Clock-Bereichen

#### Vier-Phasen Handshake



### FPGA-intern:

- Synchrone Systeme mit einer Clock als Referenzsignal.
- Taktflankenereignis bestimmt die Datenaufnahme und die Ausgangsaktualisierung.
- Bedingung:  $T_{CLK} > T_{PD} + T_{LOGIK} + T_{SU}$

### Externe Eingänge:

- Pegeländerungen treten unabhängig von der FPGA-Clock auf.
- µC-Clock ist kein ganzes Vielfaches der FPGA-Clock.
- Phasenlage der Clocks ist unbestimmt.
- Variable Taktanzahl pro C-Anweisung.



## Asynchrone Kommunikation

**Ereignisgesteuerte Effekte** an den FPGA-Eingängen erfordern **Ansatz ohne** Abhängigkeit von einem gemeinsamen oder übertragenen Clock-Signal.

- **Sender-Empfänger-Kommunikation** mit einem Protokoll, das Signalisierungskonventionen nutzt.
- Jede Komponente arbeitet mit der **eigenen Taktrate**.
- Nur für Interaktionen findet eine Kommunikation mit **synchronisierten Abläufen** statt.

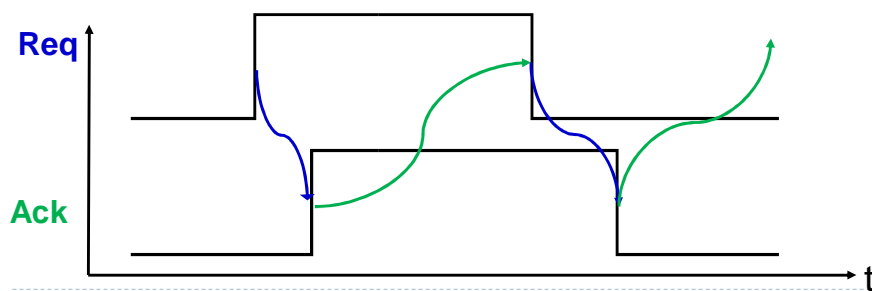
5

CE-DS 4 Synchronisation



## Handshake-Kommunikation

- **Locally clocked – globally delay-insensitive**
- **Requester/Client/Master** – **Provider/Server/Slave**
- **Vier-Phasen-Handshake**

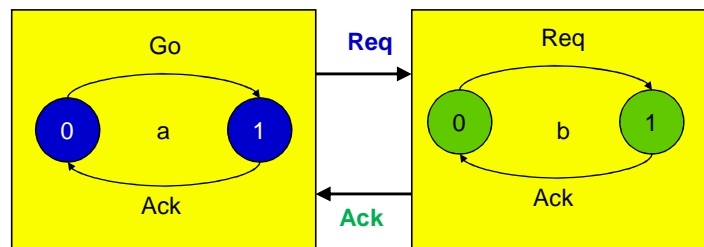


6

CE-DS 4 Synchronisation

## Wechselseitige Abstimmung

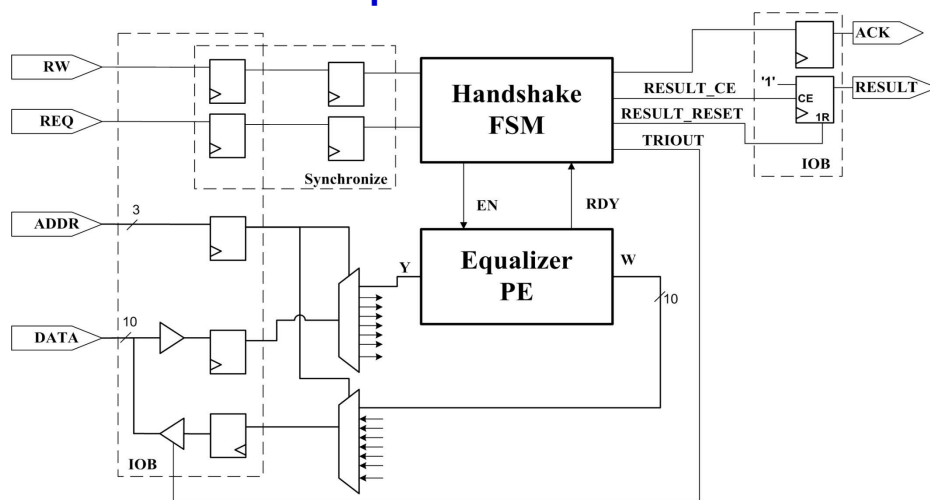
- Beide Seiten betreiben die Abstimmung mit einem Automaten.
- Zustandstransitionen zeigen den Fortschritt der abgestimmten Kommunikation an.



7

CE-DS 4 Synchronisation

## Asynchrone Kommunikation μC - FPGA



8

CE-DS 4 Synchronisation

## FPGA-Schnittstellen

- Eingangssignale werden durch **D-FFs** auf den FPGA-Takt **synchronisiert**:  
Pegel ändern sich gleichzeitig.  
Weniger, kürzere Hazards in der Eingangslogik.
- Ausgangsregister liefern eine parallele Pegelaktualisierung zu den GPIOs des  $\mu\text{C}$ .
- D-FFs in den **Input-Output-Blocks (IOBs)** stehen für beide Richtungen zur Verfügung.
- **Handshake-Signale (REQ, RW)**, die die getaktete Zustandssequenz in der FSM beeinflussen, sind einer speziellen Synchronisation zu unterziehen.

9

CE-DS 4 Synchronisation

## Asynchrone Eingangssignale

- Asynchron sind alle Eingangssignale eines synchronen RTL-Entwurfs, die nicht mit dem Systemtakt synchronisiert sind, deren Pegeländerung also irgendwann während des Taktzyklus erfolgen kann, also auch **während des Entscheidungsintervalls**  
 $t_E = t_s + t_h$  der abtastenden Clock.

Dazu gehören z.B.:

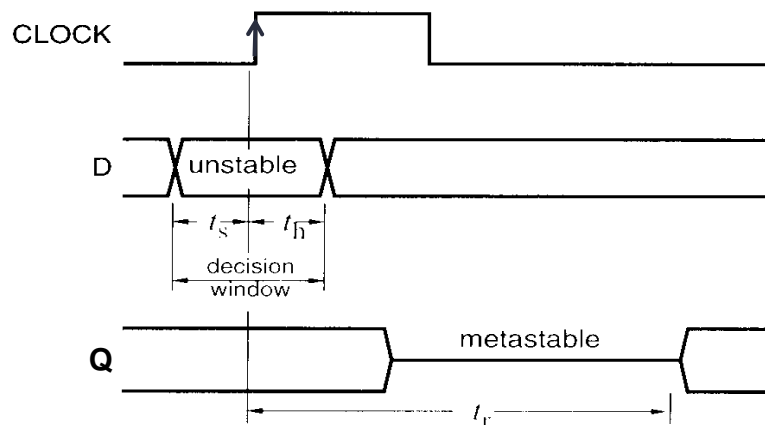
- Anforderungen externer Geräte: Tastatureingaben, Interrupts, serielle Schnittstellen, ...
- Der Datenaustausch zwischen zwei jeweils synchronen Systemen, die jedoch mit unterschiedlicher Frequenz betrieben werden: Rechnerkopplungen, USB-Interfaces zum FPGA-Prozessor.

10

CE-DS 4 Synchronisation



## Metastabiler Zustand des D-FF Ausgangs



11

CE-DS 4 Synchronisation



## Metastabiler Zustand

- Der metastabile Zustand liegt im Bereich zwischen dem High- und Low-Pegel bei „midsupply“.
- Es ist nicht vorhersehbar, welcher Logikpegel sich im Anschluss an die **Auflösungszeit  $t_r$**  (resolution time) am D-FF Ausgang einstellt.
- Alle asynchronen Eingangssignale, die auf Flipflop-, Zähler- FSM- oder Schieberegister-Eingänge geführt werden, sind speziell zu synchronisieren.
- Dies soll sicher stellen, dass sich deren Eingangssignale **nicht während des Entscheidungsintervalls  $t_E = t_s + t_h$  verändert**. Andernfalls können die Register- bzw. Zählerausgänge in den metastabilen Zustand gehen, der dann über das gesamte System verteilt wird.

12

CE-DS 4 Synchronisation

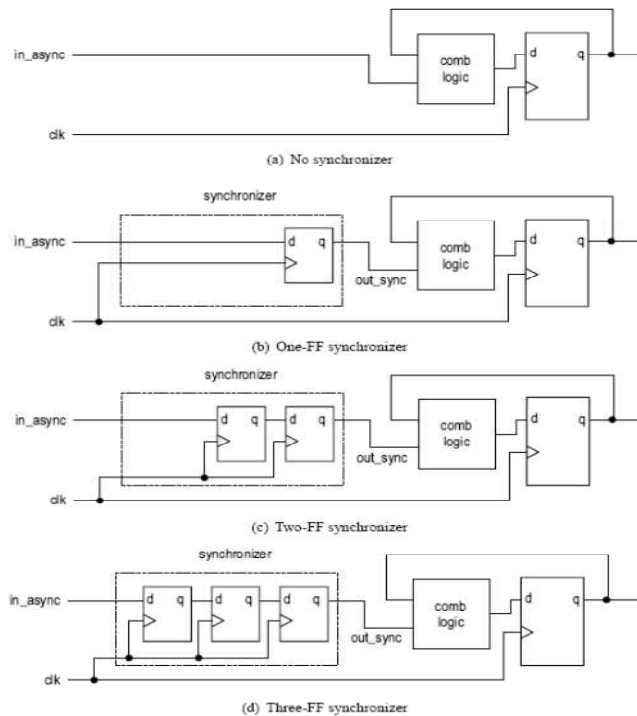


## Synchronisierer

Unterschiedliche  
lange Zeitfenster  
für die  
Auflösungszeit  $t_r$   
verfügbar.

aus P. P. Chu:  
RTL Hardware Design Using VHDL, Wiley 2006

13

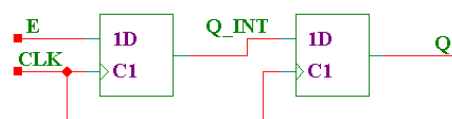


Hochschule für Angewandte Wissenschaften Hamburg  
Hamburg University of Applied Sciences

## Synchronisationsschaltung für lange Impulse

- An einem vorgeschalteten Synchronisations-FF selbst können die Setup- und Hold-Zeiten verletzt werden.

### Synchronisations Flipflop



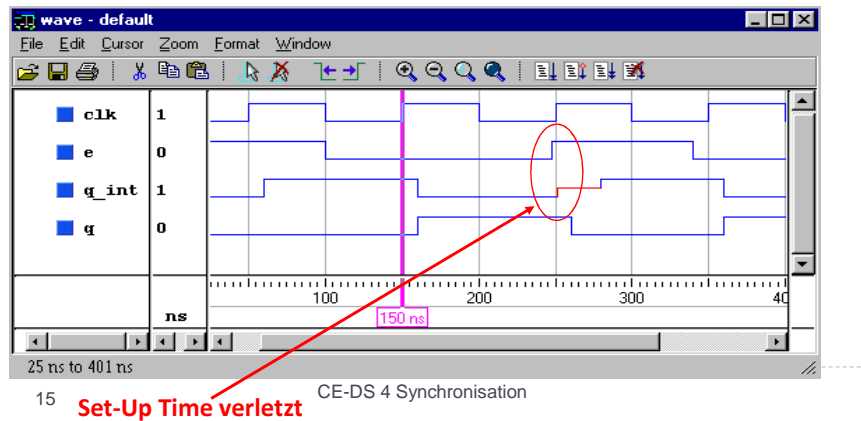
- Sofern das 1. Synchronisations-D-FF in den metastabilen Zustand übergegangen ist, kann dieser Zustand sich jedoch im Laufe der Taktperiode auflösen ( $t_r < T_{CLK}$ ) und damit das 2. D-FF einen korrekten Logikpegel Q\_INT übernehmen.
- Damit wäre die Wahrscheinlichkeit reduziert, dass angeschlossene FSMs und Datenpfade mit undefinierten Pegeln angesteuert werden.

14

CE-DS 4 Synchronisation

## Auflösungszeit

- Das Signal E wird durch die Synchronisation um bis zu zwei Takte verzögert!
- Die Wahrscheinlichkeit für das Auftreten eines Fehlers wird durch eine mittlere Zeit zwischen auftretenden Fehlern **MTBF (mean time between failure)** beschrieben.



## Abschätzung der Fehlerhäufigkeit von D-FFs

- Synchronisationsfehler treten auf, wenn der metastabile Zustand länger andauert, als die vom Hersteller angegeben Auflösungszeit  $t_r$ .
- Die mittlere Zeit **MTBF** zwischen zwei Synchronisationsfehlern ist abhängig von:
  - $f$ : Taktfrequenz mit der die Flipflops betrieben werden.
  - $a$ : Frequenz mit der sich das asynchrone Signal ändert.
  - $T_0$ : D-FF Konstante gibt das Zeitfenster an, in dem Pegelwechsel einen Fehler erzeugen.
  - $\tau$ : D-FF Konstante des Übergangsverhaltens von metastabilen Zuständen



## MTBF

$$MTBF(tr) = \frac{\exp(tr / \tau)}{T0 \cdot f \cdot a}$$

Familie	$\tau$ /ns	T0/s	tr/ns	$t_{SU}$ /ns
74LS74	1.5	$4.0 \cdot 10^{-1}$	77.71	20
74HCxx	1.82	$1.5 \cdot 10^{-6}$	71.55	25
XC95108-20	0.17	$9.6 \cdot 10^{-18}$	2.3	10

Vgl. J. F. Wakerly: Digital Design. Principles and Practices. Pearson 4th ed 2006

17

CE-DS 4 Synchronisation

## Beispiel: MTBF

- Das Interruptsignal eines Mikroprozessor, der mit  $f = 10$  MHz Taktfrequenz arbeitet wird mit zwei 74LS74 D-FFs synchronisiert.
- Der asynchrone Interrupt tritt mit einer Rate von  $10^5$  1/s auf.
- Die verfügbare Zeit zum Abklingen des metastabilen Zustands ist  $tr = 1/f - t_{SU} = 80$ ns.
- Fehler treten auf mit  $MTBF(80 \text{ ns}) = 3.6 \cdot 10^{11}$  s, sodass fehlerhafte Übergänge etwa alle 114 Jahrhunderte auftreten!
- Falls der Prozessor jedoch mit 16MHz getaktet werden soll, ergibt sich:  
 $tr = 42.5 \text{ ns} \rightarrow MTBF(42.5 \text{ ns}) = 3.1 \text{ s}$ . **Nich takzeptabel!**
- Berechnen Sie die MTBF für den Baustein XC95108-20 jeweils bei  $f = 16$ MHz und  $f = 50$ MHz!

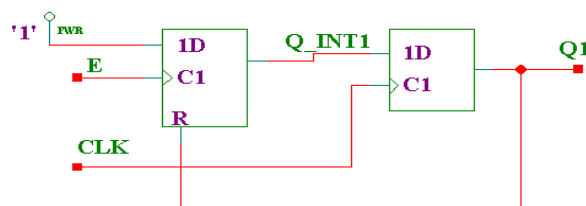
18

CE-DS 4 Synchronisation



## Synchronisationsschaltung für kurze Impulse

- Das asynchrone Eingangssignal E liegt am Takteingang des ersten Synchronisations-Flipflops.
- Das erste D-FF wird durch das zweite Flipflop asynchron zurückgesetzt.
- Unabhängig von der Impulsdauer des Signals E erzeugt die Schaltung immer einen **Puls Q1 für die Länge eines Taktes ( $T_{CLK}$ )!**
- Der Ausgang Q1 ist nicht frei von metastabilen Zuständen, ggf. muss die Schaltung um ein weiteres Flipflop ergänzt werden.

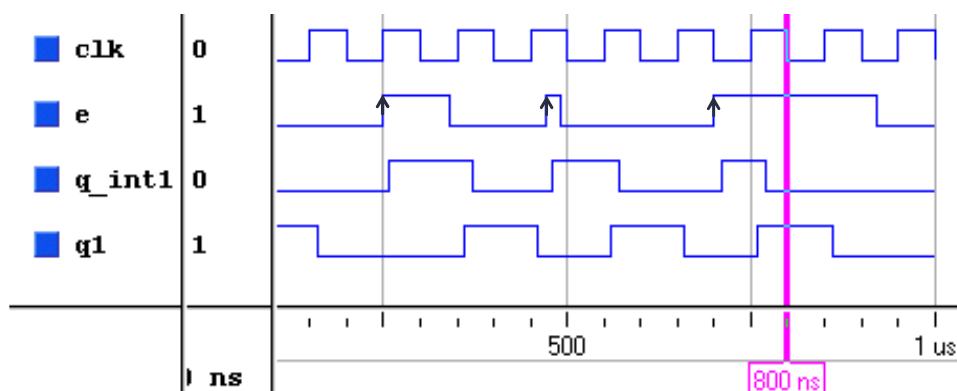


19

CE-DS 4 Synchronisation

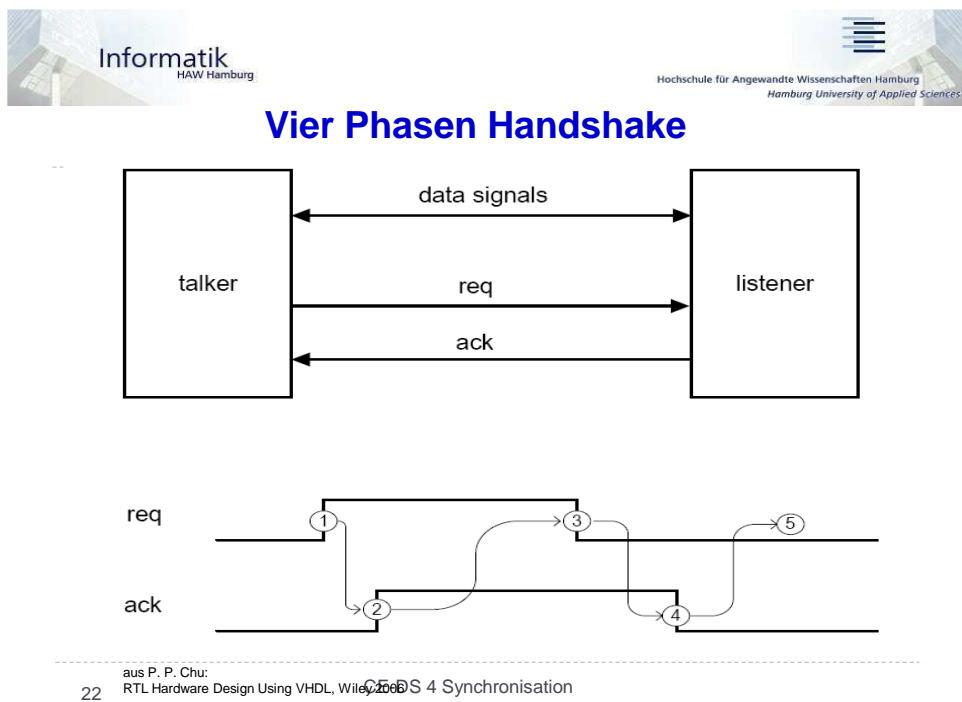
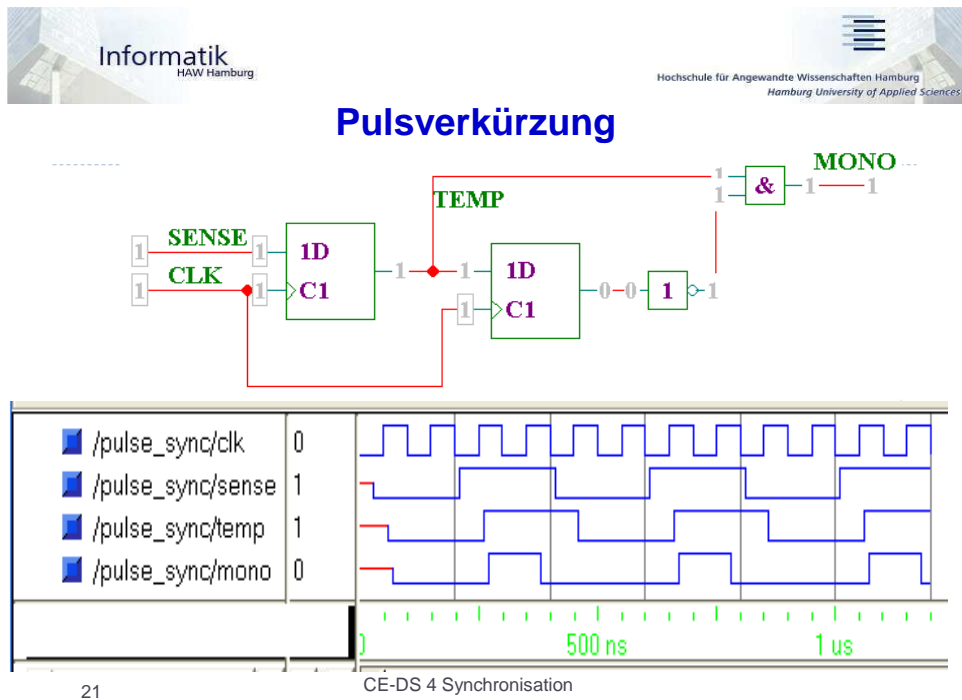


## Pulsverlängerung

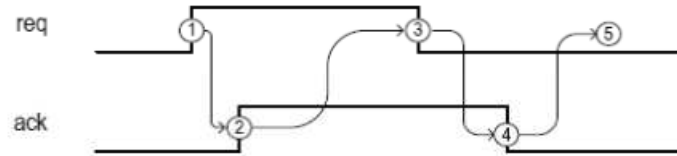


20

CE-DS 4 Synchronisation



## Aufforderungs- und Bestätigungssequenz

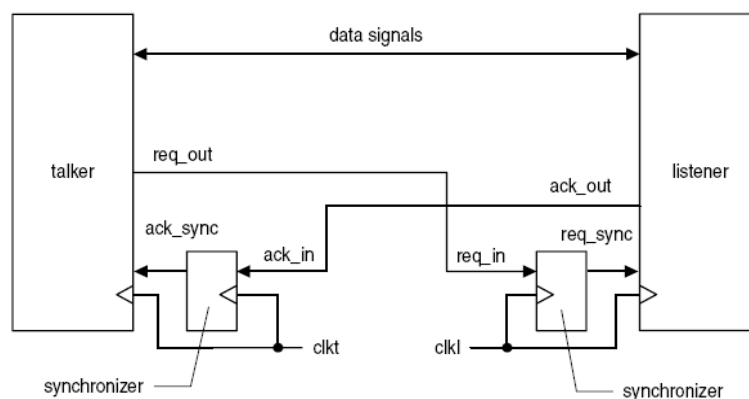


- Phase 1: Talker aktiviert      req = 1      (A)
- Phase 2: Listener aktiviert      ack = 1      (B, A)
- Phase 3: Talker deaktiviert      req = 0      (B, A)
- Phase 4: Listener deaktiviert      ack = 0      (B)
- Talker kann neuen Request starten

23

CE-DS 4 Synchronisation

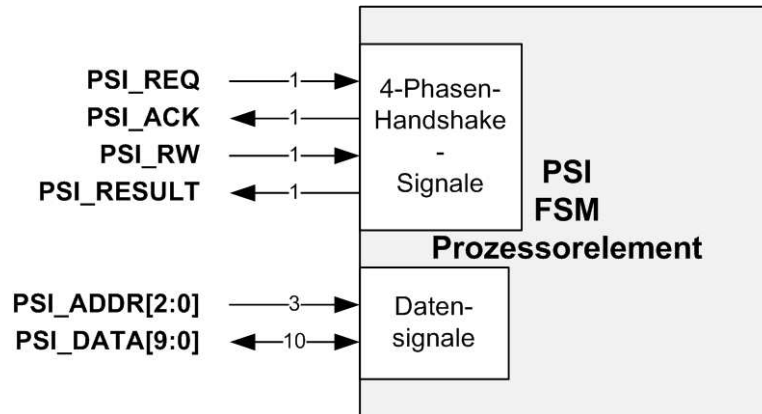
## Synchronisation erforderlich für Talker und Listener in unterschiedlichen Clock Domains



aus P. P. Chu:  
RTL Hardware Design Using VHDL, Wiley 2006  
24

CE-DS 4 Synchronisation

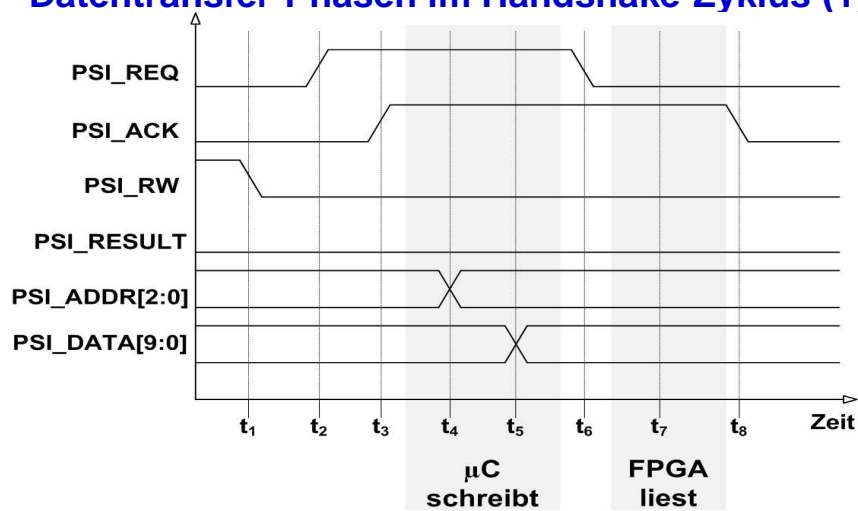
## μC – FPGA Interface



25

CE-DS 4 Synchronisation

## Datentransfer-Phasen im Handshake-Zyklus (1)

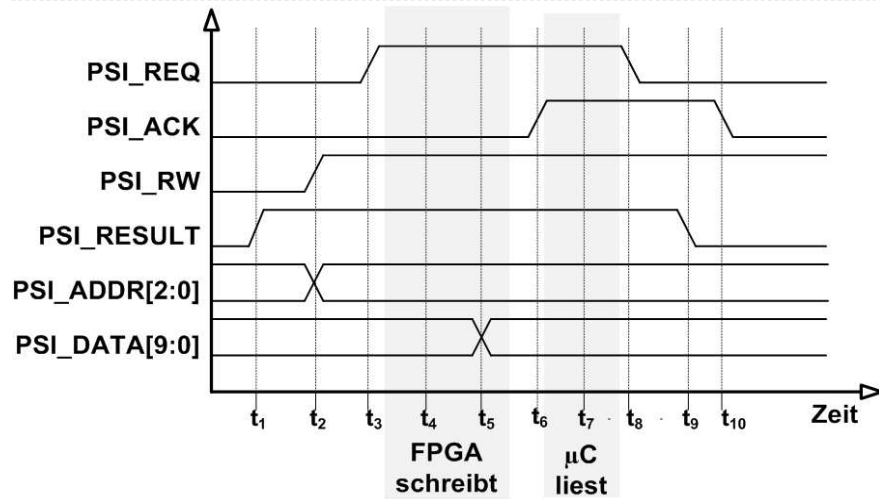


26

CE-DS 4 Synchronisation



## Datentransfer-Phasen im Handshake-Zyklus (2)

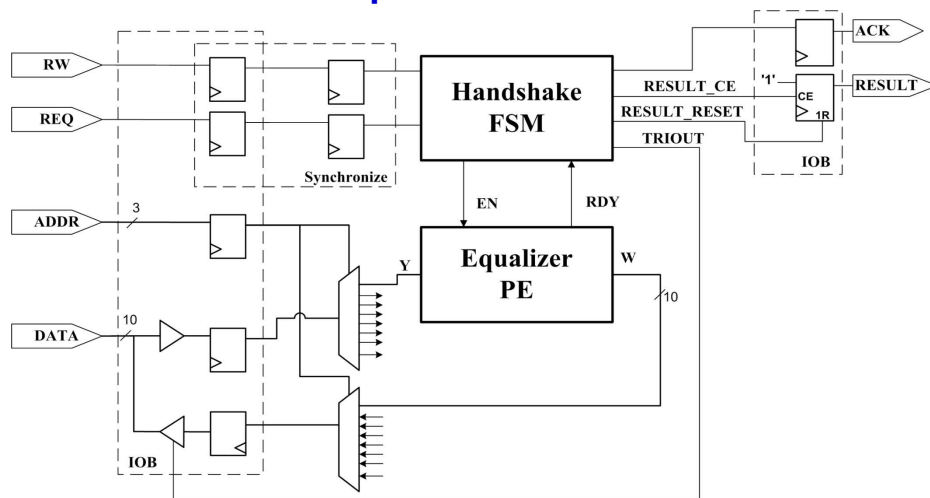


27

CE-DS 4 Synchronisation

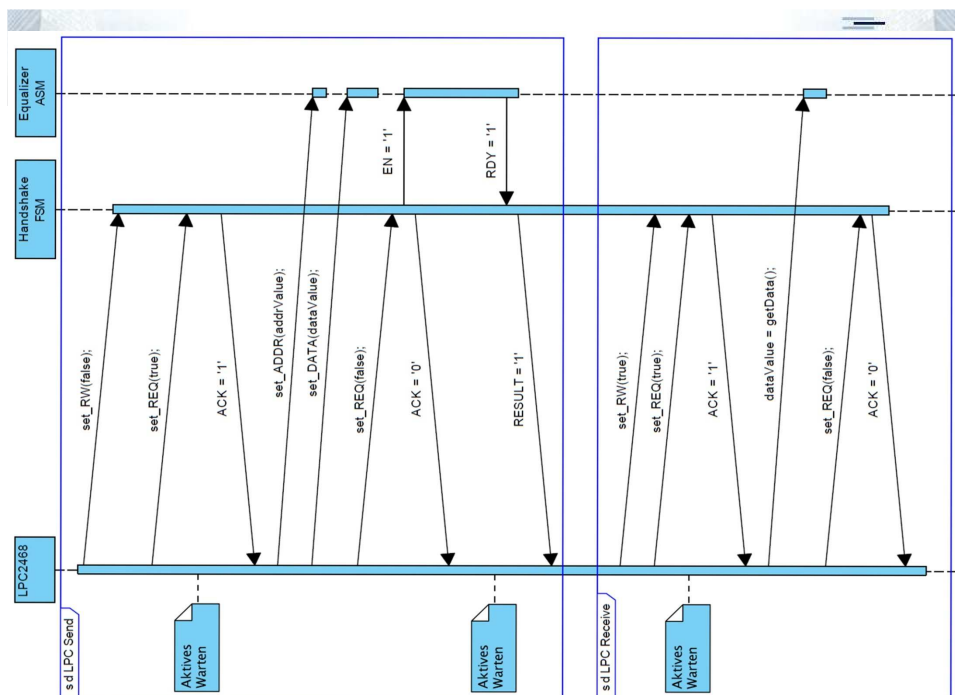
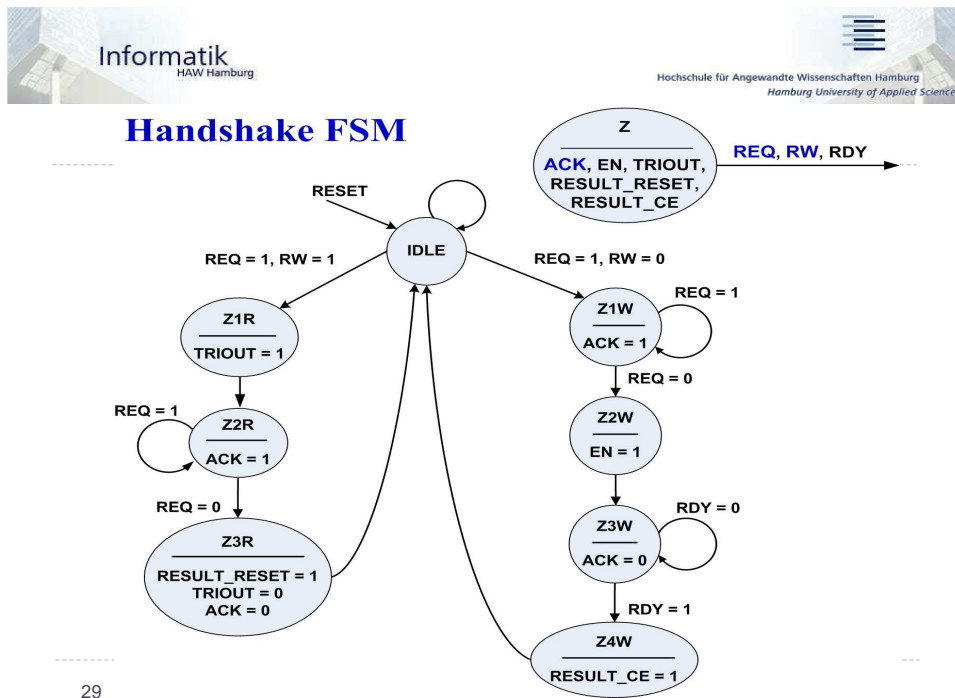


## Asynchrone Kommunikation µC - FPGA



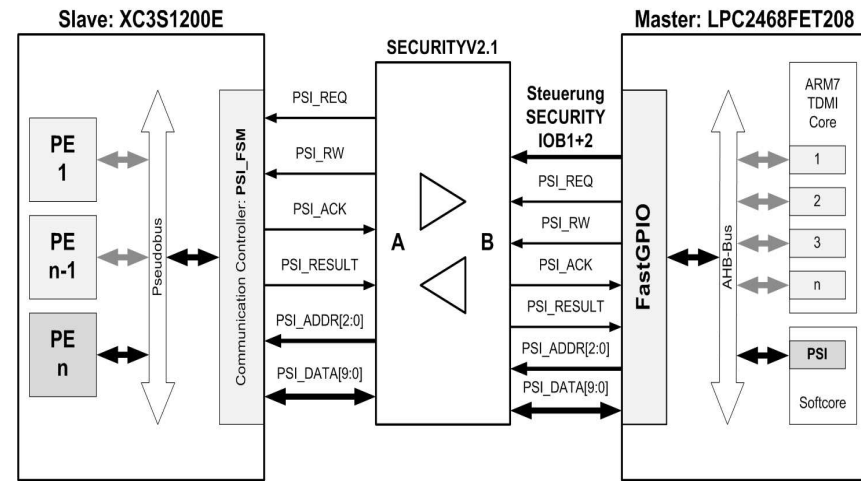
28

CE-DS 4 Synchronisation





## Security Board Interface

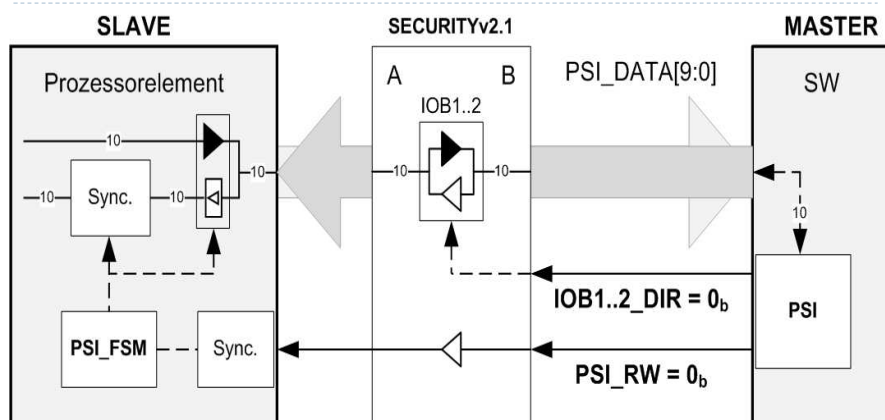


31

CE-DS 4 Synchronisation



## Transfersteuerung $\mu$ C - FPGA



Schreibvorgang: B -> A

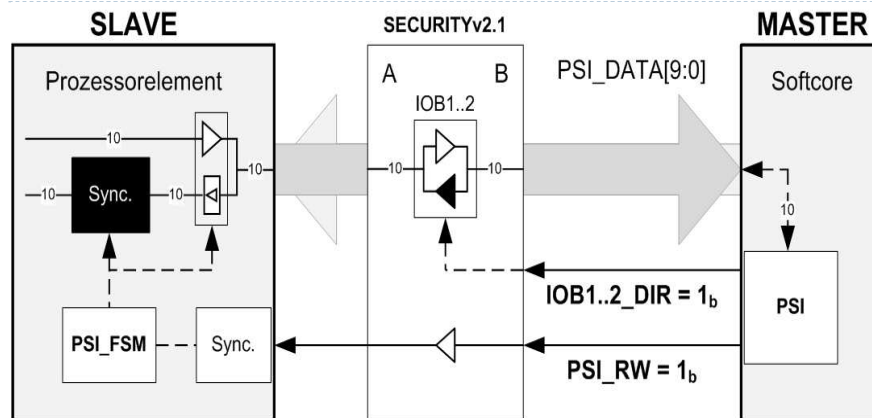
32

CE-DS 4 Synchronisation





## Transfersteuerung FPGA - $\mu$ C



Lesevorgang: A -> B

33

CE-DS 4 Synchronisation



## Write – Sequenz in ISR

```
// Macro-Definitionen in hwconfig.h
...
                                /** LPC-2468 Write to FPGA **/
PSI_ENABLE_WRITE;                /* Datenflussrichtung: B -> A */
PSI_SET_REQ;                      /* 1. Phase: Anforderung */
PSI_WAIT_ACK_SET;                 /* 2. Phase: Bestätigung */
PSI_WRITE_DATA( outdata );        /* 10 Bit Datum schreiben */
PSI_CLR_REQ;                      /* 3. Phase */
PSI_WAIT_ACK_CLR;                 /* 4. Phase; ACK = '0' Polling */
...
```

34

CE-DS 4 Synchronisation

## Read – Sequenz in ISR

```

/** LPC-2468 Read from FPGA */
PSI_WAIT_RESULT_SET; /* Ergebnisse verfügbar ? */

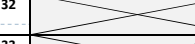
PSI_ENABLE_READ; /* Datenflussrichtung: A -> B */
PSI_SET_REQ; /* 1. Phase: FPGA schreibt */
PSI_WAIT_ACK_SET; /* 2. Phase */
PSI_READ_DATA( *indata ); /* 10 Bit Datum lesen */
PSI_CLR_REQ; /* 3. Phase: Lesebestätigung */
PSI_WAIT_ACK_CLR; /* 4. Phase: Lesesequenz beendet */
/* Datenausgabe an DAC und PWM */

```

35

CE-DS 4 Synchronisation

## Pin-Belegung

Daten-/Steuersignale				Development Boards								
Name	Typ	Anmerkung	Transfer	TI-LPC uC Port[Pin]	SECURITYv2.1 Bustreiber	Connector Type	Pin	NEXYS2 FPGA			Connector FX2-100 Port-Pin	
								Signal	Pin			
PSI_REQ	S	Auftrag	unidirektional	P1[12]	IOB4	X2	24	IOB4<3>	F11	J1A-34		
PSI_ACK	S	Auftragsbestätigung	unidirektional	P1[8]	IOB5	X2	25	IOB5<2>	C14	J1A-41		
PSI_RW	S	Schreiben/Lesen	unidirektional	P1[11]	IOB4	X2	23	IOB4<3>	F11	J1A-34		
PSI_RESULT	S	Auftragsfertigstellung	unidirektional	P2[12]	IOB5	X3	9	IOB5<1>	A14	J1A-40		
PSI_ADDR[0]	S	Adresse des Prozessorelements	unidirektional	P1[15]	IOB4	X2	33	IOB4<0>	B11	JA1-31		
PSI_ADDR[1]				P1[14]			34	IOB4<1>	C11	JA1-32		
PSI_ADDR[2]				P1[13]			29	IOB4<2>	E11	JA1-33		
PSI_DATA[0]	D	Datum	bidirektional	P0[5]	IOB1	X3	36	IOB1<0>	A4	J1A-07		
PSI_DATA[1]				P1[10]			X2	26	IOB1<1>	C3	J1A-08	
PSI_DATA[2]				P0[13]				5	IOB1<2>	C4	J1A-09	
PSI_DATA[3]				P0[14]		7		IOB1<3>	B6	J1A-10		
PSI_DATA[4]				P0[19]		X3	15	IOB1<4>	D5	J1A-11		
PSI_DATA[5]				P0[20]			14	IOB1<5>	C5	J1A-12		
PSI_DATA[6]				P0[21]	X2	19	IOB1<6>	F7	J1A-13			
PSI_DATA[7]				P0[22]		18	IOB1<7>	E7	J1A-14			
PSI_DATA[8]				P0[29]		IOB2	6	IOB2<0>	A6	J1A-15		
PSI_DATA[9]				P0[30]			15	IOB2<1>	C7	J1A-16		
IOB1_DIR	S	Transferrichtung IOB1- PSI_DATA[7:0]	unidirektional	P2[3]	IOB1	X3	32					
IOB2_DIR	S	Transferrichtung IOB2- PSI_DATA[9,8]	unidirektional	P2[4]	IOB2	X3	33					



**PSI\_WRITE\_DATA( outdata );**

**/\* 10 Bit Datum schreiben\*/**

**//Daten Ein- und Ausgabe**

**#define PSI\_WRITE\_DATA( d ) FIO0MASK = ~PSI\_DATA\_PINS\_PORT0;**

**\**

**FIO0PIN = (d & (3<<8))<<(29-8) | (d & (0xf<<4))<<(19-4) | (d & 0xc)<<11 | (d & 1)<<5;**

**\**

**FIO0MASK = 0;**

**\**

**FIO1MASK = ~PSI\_DATA\_PINS\_PORT1;**

**\**

**FIO1PIN = (d & 2)<<9;**

**\**

**FIO1MASK = 0;**