

Tutorium zu Computer-Engineering im SS19 Termin 5

Jakob Otto

HAW Hamburg

28. Mai 2019

- Aufgabe 5

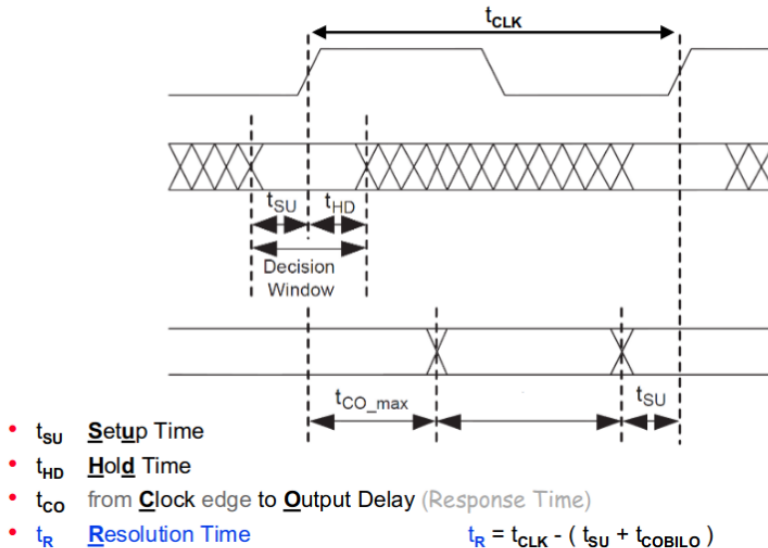


Aufgabenzettel

Metastabile Zustände

- Signale sind nicht unmittelbar stabil
- Set-up hold-time usw.
- Signal kann also zur „falschen“ Zeit abgetastet werden
- → Metastabile Zustände

Metastabile Zustände



Asynchrone kommunikation

- Kommunikation zwischen Endpunkten oft asynchron
- Beide Seiten nutzen eigene Taktfrequenz
- Synchronisation wird benötigt

4-Phasen-Handshake

- Kommunikationsprotokoll zum Übermitteln von Daten
- Bidirektional über einen Datenbus
- Verhindert
 - ▶ Metastabile Zustände
 - ▶ Kurzschlüsse

Welche Leitungen?

Benötigt verschiedene Leitungen zur Datenübertragung

- RD/nWR - signal zum lesen/schreiben signalisieren
- REQ - Request vom Master zu Slave
- ACK/nRDY - acknowledge vom Slave zum Master
- Data - 16 bit Datenbus

Ablauf vom Handshake

Die 4 Phasen sind:

- 1 Master initiiert RD/nWR = '1'
Datenbus auf high-impedance
REQ auf '1'
- 2 Slave reagiert
legt geforderte Daten auf Datenbus
acknowledged (ACK = '1')
- 3 Master quittiert Empfang
REQ = '0'
- 4 Slave nimmt Daten vom Bus
Datenbus auf High-impedance

lesender Zugriff

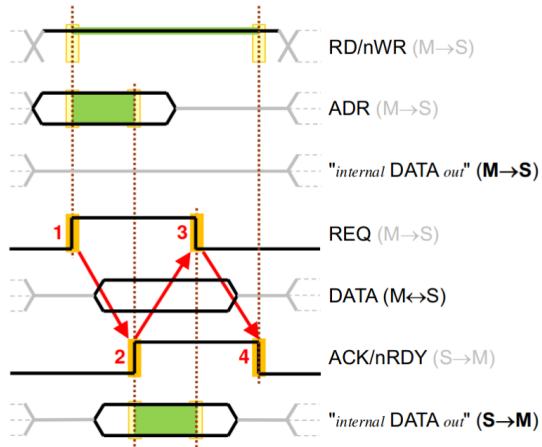


Abbildung: 4-Phasen-Handshake lesender Zugriff

```
uint16_t  rxdat;
/* wait for the fpga to be ready */
while(READY == 0);

/* setup the connection to the FPGA */
OUTPUT_DISABLE;
SET_READ;

// handle transaction by 4 phase handshake
REQ_ENABLE;
while(ACKNOWLEDGE == 0);
rxdat = (GPIOE->IDR & 0x0000FFFF);
REQ_DISABLE;
while(ACKNOWLEDGE != 0);

return rxdat;
```

Ablauf vom Handshake

Die 4 Phasen sind:

- 1 Master initiiert $RD/nWR = '0'$
legt Daten auf Bus
REQ auf '1'
- 2 Slave quittiert
ACK = '1'
- 3 Master setzt Bus auf High-impedance
REQ = '0'
- 4 slave quittiert quittung
ACK = '0'

schreibender Zugriff

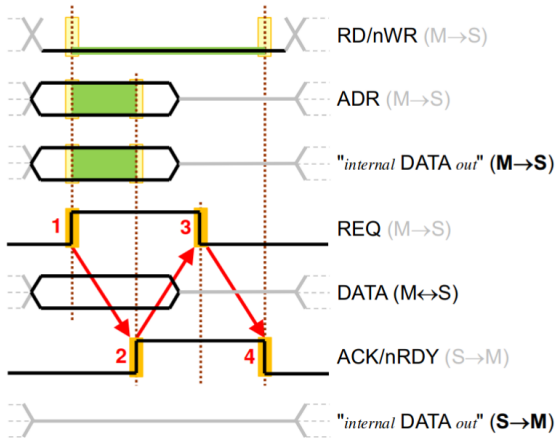


Abbildung: 4-Phasen-Handshake lesender Zugriff

```
/* setup connection to the FPGA */  
SET_WRITE;  
  
OUTPUT_ENABLE;  
/* set data */  
GPIOE->ODR = txdat;  
  
// handle transaction by 4 phase handshake  
REQ_ENABLE;  
while (ACKNOWLEDGE == 0);  
REQ_DISABLE;  
while (ACKNOWLEDGE != 0);
```

Tri-State Treiber für Datenbus

```
tristate:
process (oe_s, dato_s) is
begin
    if oe_s = '1' then
        data <= dato_s;
    else
        data <= (others=>'Z');
    end if;
end process tristate;
--
dati_s <= data;
```

