

Aufgabenblatt 13: Datenübertragung

Aufgabe 1: VHDL Protokollautomat

Zwischen zwei Entities werden Daten übertragen. Das Protokoll wird auf beiden Seiten durch den unten stehenden Ausschnitt aus der VHDL-Architekturbeschreibung realisiert.

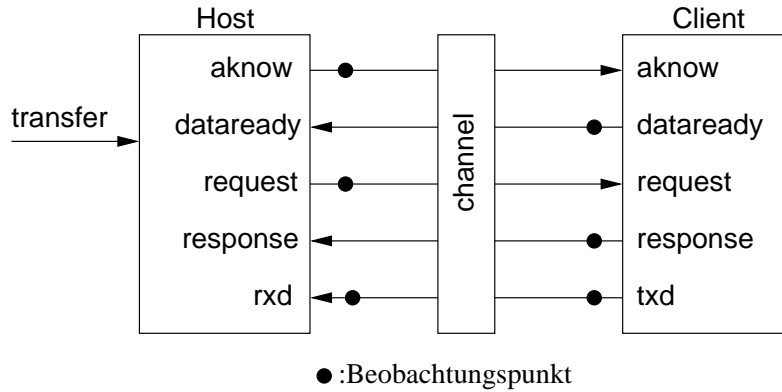


Abbildung 1: Verbindungsstruktur zwischen Host und Client.

Die Kanalverzögerung zwischen den Entities T_K beträgt 50ns.

Die Treiberverzögerung ist $T_{DEL}=25ns$,

die Datentreiberverzögerung ist $DATADEL = 75ns$.

Der Datenaustausch erfolgt durch die Verbindung $TxD \rightarrow RxD$.

Ausschnitt aus der Architekturbeschreibung der HOST-Entity:

```
...
architecture Host_arch of Host is
    type STATES is (IDLE, CALL, CONFIRM, DATA, AKN);
    signal state: STATES := IDLE;
    signal transfer : BOOLEAN;
    signal incoming : std_logic;
begin
    ...
    TRANSFERFP: process (transfer,state,response,rxd,dataready)
    begin
        case state is
            when IDLE =>
                aknow <= '0' after TDEL;
                request <= '0' after TDEL;
                if transfer then
                    state <= CALL;
                end if;
            when CALL =>
                if response='0' and dataready='0' then
                    request <= '1' after TDEL;
                else
                    state <= CONFIRM;
                end if;
            when CONFIRM =>
                if response = '1' then
                    request <= '0' after TDEL;
                end if;
                if dataready = '1' then
```

```

        state <= DATA;
    end if;
when DATA =>
    if dataready = '1' then
        incoming <= RxD;
        state <= AKN;
    end if;
when AKN =>
    aknow <= '1' after TDEL;
    if dataready = '0' then
        aknow <= '0' after TDEL;
        state <= IDLE;
    end if;
when others =>
    state <= IDLE;
end case;
end process;
...

```

Ausschnitt aus der Architekturbeschreibung der CLIENT-Entity:

```

...
AB: process
begin
    response <= '0' after TDEL;
    txd <= '0' after DATADEL;
    dataready <= '0' after TDEL;
    wait until request = '1' ;
        response <= '1' after TDEL;
    wait until (request'event and request = '0' );
        response <= '0' after TDEL;
        txd <= data after DATADEL;
    wait for 150 ns;
        dataready <= '1' after TDEL;
    wait until (aknow'event and aknow = '1' );
        dataready <= '0' after TDEL;
        txd <= 'Z' after DATADEL;
    wait until (aknow'event and aknow = '0' );
    wait for 150 ns;
end process;
...

```

1. Zeichnen Sie das ablaufende Protokoll maßstabsgetreu über einen Zeitraum von 800ns (siehe Abbildung 2) auf und markieren Sie die Abhängigkeiten (transfer geht bei 0ns auf '1').
Betrachten Sie dazu die Signalpegel an den Ausgängen der jeweils treibenden Entity (ausser: RxD am Eingang des Host und TxD am Ausgang des Client), also an den eingezeichneten Beobachtungspunkten!
2. Um welche Art von Protokoll handelt es sich (non interlocked, full interlocked, half interlocked)?

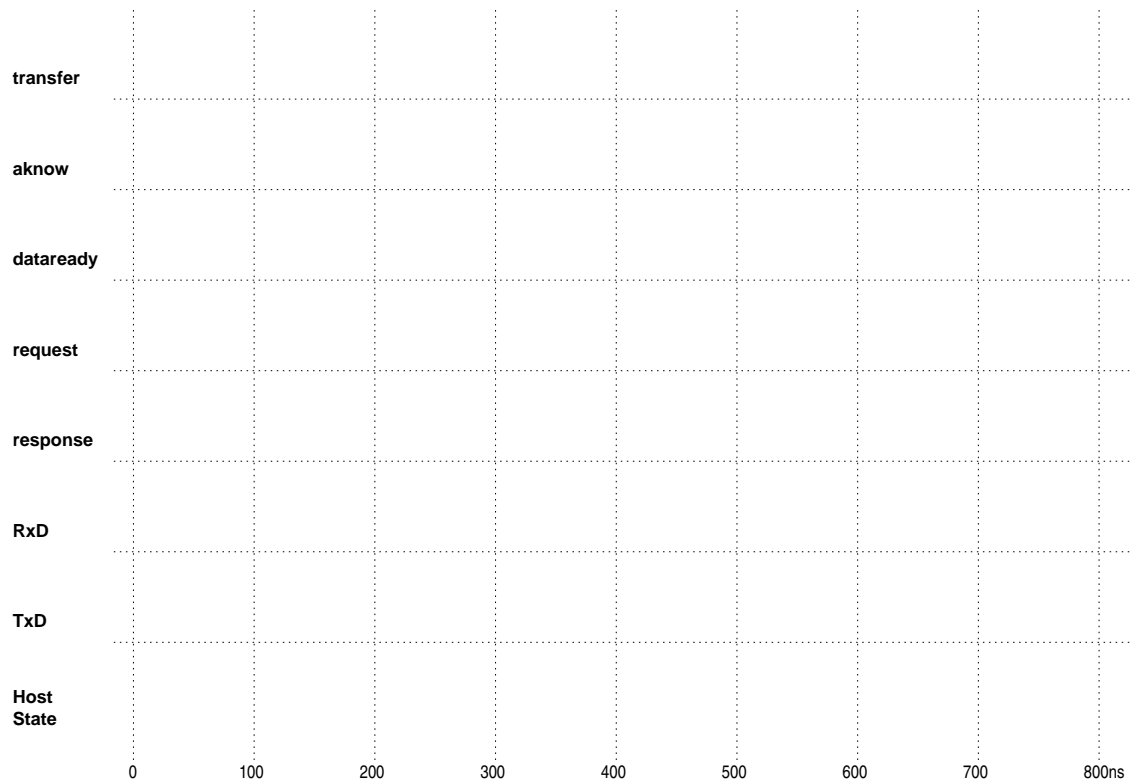


Abbildung 2: Timing des Protokolls.

Aufgabe 2: Memory-PCI-Bridge

Ein Prozessor-Bus (Memory-Bus) und ein PCI-Bus sind über eine Memory-PCI-Bridge verbunden. Der Memory-Bus arbeitet mit einer Taktfrequenz $f_{MEM} = 100$ MHz, der PCI-Bus mit einer Taktfrequenz von $f_{PCI} = 66.6$ MHz.

1. Wie lang ist ein Takt auf dem Memory-Bus?
2. Wie lang ist ein Takt auf dem PCI-Bus?
3. Wieviel Memory-Takte ist ein PCI-Takt lang?
4. Wieviel PCI-Takte ist ein Memory-Takt lang?
5. Zeichnen Sie die beiden Takte für 6 Memory-Takte untereinander!
6. Eine Transaktion auf dem Memory-Bus beginnt immer mit der positiven Flanke des Taktes; die auf dem PCI mit einer positiven Flanke des PCI-Taktes. Beide aber niemals exakt gleichzeitig.
Die Bridge muss nun Transaktionen vom Memory-Bus auf den PCI-Bus (und umgekehrt) durchschalten.
Da die Taktraten unterschiedlich sind, muss immer bis zum nächsten Start eines PCI-Zyklus gewartet werden, um dann die Transaktion durchzuführen.
Wieviel Memory-Bustakte ist maximal die Verzögerung, bis die Transaktion auf dem PCI-Bus beginnen kann?

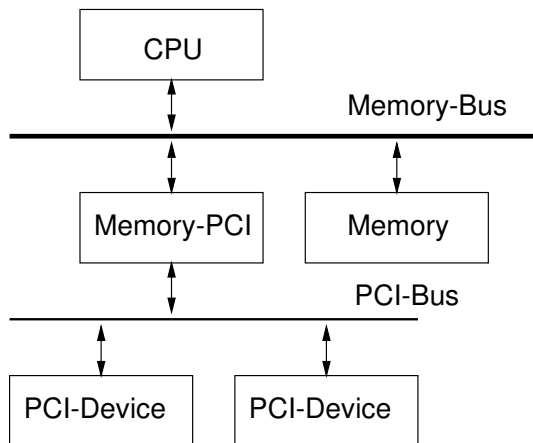


Abbildung 3:
Bus-Architektur.

7. Wieviel Memory-Bustakte dauert die Transaktion, wenn die Transaktion innerhalb eines PCI-Taktes abgeschlossen wird?
8. Wie lange dauert die gesamte Transaktion insgesamt im schlechtesten Fall?
 $D(1) = ?$
9. Wie lange dauert die gesamte Transaktion insgesamt, wenn die Transaktion n -Takte auf dem PCI-Bus benötigt?
 $D(n) = ?$
10. Über den Memory-Bus erfolgt die Transaktion als non-multiplexed Operation, auf dem PCI-Bus als multiplexed Operation. Der PCI-Bus ist 32 Bit breit.
Wie hoch ist die Netto-Datenrate (Byte pro Sekunde), die auf dem PCI-Bus erreicht werden kann? (In der Literatur wird oft nur die Brutto-Rate mit $266,6 \frac{\text{MByte}}{\text{s}}$ angegeben.)

Aufgabe 3: Interrupt-Behandlung mit Daisy-Chain

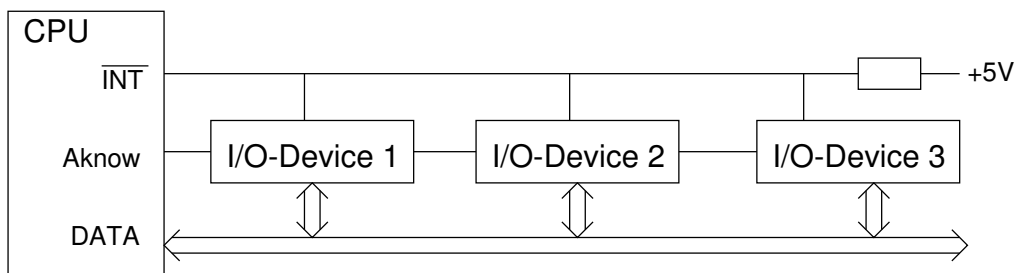


Abbildung 4: Rechnerarchitektur mit Interrupt-Behandlung mittels Daisy-Chain.

Jedes I/O-Device hat eine Durchlaufzeit der Daisy-Chain von 133ns. Legt das Device die Kennung auf den Datenbus, so benötigt es dazu 100ns. Die CPU übernimmt dann weitere 100ns später die Kennung und schlägt die Sprungadresse in der Vektortabelle nach.

1. Wie lange dauert es maximal vom Interrupt bis die *Interrupt Service Routine (ISR)* angesprungen wird, wenn eine Befehlsausführung 333ns dauert und I/O-Device 3 einen Interrupt auslöst (Nachschlagen in der Interrupt-Vektortabelle dauert 50ns)?