

Aufgabe 1

(22 Punkte)

Xilinx Zynq Plattform

1. Erläutern Sie stichpunktartig den Unterschied zwischen Hardcore- und Softcore-Prozessoren. Reflektieren Sie dies anhand des Xilinx Zynq Bausteins.

Softcore-Prozessor: IP Core, der in FPGA-Fabric programmiert ist.

Hardcore-Prozessor: ~~ARM~~ Prozessor im chip, sowie Zynq mit zwei A9-ARM-Cores.

2. Erklären Sie wie die Adressierung mehrerer Slaves konkret über AXI erfolgt und welche Einschränkungen es gibt.

Die Adressierung erfolgt über Memory-Mapping, durch zuweisen eines bestimmten Adressbereichs. Da der Adressbereich begrenzt ist müssen alle Slaves sich den Adressraum teilen.

3. Wie viele verschiedene Arten von Interrupts können vom AXI GPIO-Core generiert werden?

Zwei: (ohne Software)
SPI und PPI.

Aufgabe 2

(24 Punkte)

VHDL Entwurf und High-Level Synthese

1. Syntaktisch korrekten VHDL-Code zu schreiben ist relativ einfach. Jedoch wird bei der Synthese nicht immer die erwartete Hardware erzeugt. Erläutern Sie diesen Sachverhalt, vorzugsweise mit 1-2 Beispielen.

Bsp. Verzögerungsbasierendes Design: $b \leftarrow a \text{ and } \text{not } a$
 wird wegoptimiert, da d. transiente Verhalten nicht berücksichtigt wird.

2. High-Level Synthese (HLS) ermöglicht dem Entwickler die Transformation (Synthese) von C/C++/SystemC-Code in VHDL. Für welche Anwendungsfälle eignet sich HLS besonders gut? Wann sollten Sie aber (zwingend) VHDL nutzen?

HLS eignet sich hervorragend für den schnellen Entwurf, hat aber seine Grenzen, wenn bspw. Timing Constraints nicht eingehalten werden können, muss man u.U. von Hand an den VHDL-Code

3. Betrachten Sie folgenden VHDL-Prozess. Nehmen Sie an, dass alle Signale 16 Bit breit sind (außer clk).

```

PROCESS (clk, a, b, c)
BEGIN
  x <= a + b;
  IF rising_edge(clk) THEN
    IF a = c THEN
      y <= b + c;
      z <= c;
    ELSIF b > c THEN
      y <= a - x;
    END IF;
  END IF;
END PROCESS;

```

Nun wird aus diesem Prozess eine Schaltung für die Implementierung auf einem FPGA synthetisiert. Wie viele Flipflops werden von dieser Schaltung verwendet? Wie viele Latches? Begründen Sie Ihre Antwort.

16 Latches für $x \leftarrow a + b$
 16 FFs für y
 16 FFs für z

\Rightarrow 16 Latches
 32 FFs

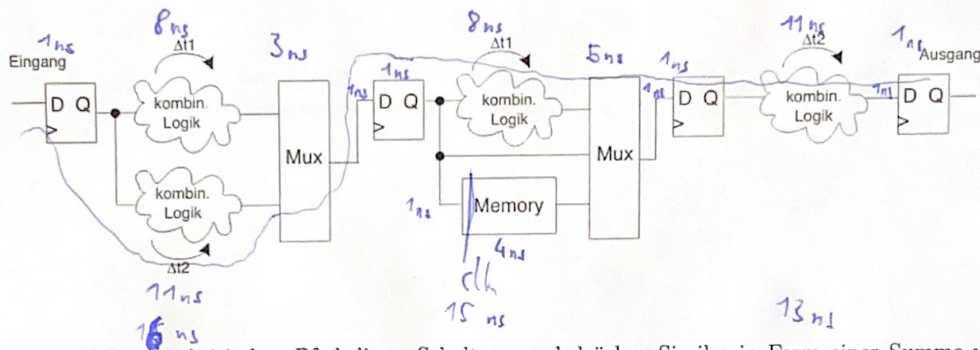
Aufgabe 3

(28 Punkte)

Dynamisches Verhalten und Timing von Schaltungen

1. Gegeben ist die unten abgebildete Schaltung. Die obligatorischen Steuersignale wurden der Übersicht halber weggelassen. Der als Memory eingezeichnete Block entspricht einem RAM. Nehmen Sie für die Komponenten folgende Parameter an:

Verzögerung (Propagation Delay) von Flipflops bzw. Registern	$d_{REG} = 1ns$
Setup Time von Flipflops bzw. Registern	$d_{SG} = 1ns$
Verzögerung des 2:1 Mux	$d_{MUX2} = 3ns$
Verzögerung des 3:1 Mux	$d_{MUX3} = 5ns$
Verzögerung der kombinatorischen Logik	$\Delta t_1 = 8ns$
Verzögerung der kombinatorischen Logik	$\Delta t_2 = 11ns$
Leseverzögerungen des RAM (von ADDR bis DOUT, asynchron)	$d_{RAM} = 4ns$
Setup Time vom RAM (bezogen auf Signale am Eingang und der nächsten Taktflanke)	$d_{SM} = 1ns$



Bestimmen Sie den kritischen Pfad dieser Schaltung und drücken Sie ihn in Form einer Summe von Variablen aus, die Verzögerungen und Setup-Zeiten aller enthaltenen Komponenten darstellen. (Hinweis: nur die Formel ist hier gefragt)

~~$d_{krit} = d_{reg} + \Delta t_2 + d_{mux_2} + d_{sg} + d_{reg} + \Delta t_1 + d_{mux_3} + d_{sg} + d_{reg} + \Delta t_2 + d_{sg} + d_{reg}$~~

$d_{krit} = d_{reg} + \Delta t_2 + d_{mux_2} + d_{sg}$

2. Berechnen Sie die Verzögerung des kritischen Pfades. (Hinweis: hier ist die Rechnung gefragt)

$d_{krit} = 1ns + 11ns + 3ns + 1ns = 16ns$

3. Welche minimale Taktperiode und maximale Taktfrequenz ergibt sich daraus?

~~Ohne Taktunsicherheit:~~

$T_{max} = 16ns \rightarrow f_{max} = \frac{1}{T_{max}} = \frac{1}{16ns} = \frac{1}{16} MHz = 62,5 MHz$

Aufgabe 4

$n \leq$ Schleifendurchläufe $C = 60LE$

(26 Punkte)

Partitionierung

$$f(n) = \frac{1}{2} \left(n \cdot 60LE \right) + \left(\frac{1}{4} \cdot \frac{1}{4} \right) \cdot \frac{1}{2}$$

1. Zur Bewertung von Partitionierungen können Kostenfunktionen verwendet werden. Gehen Sie von folgendem Beispiel aus: Die benötigten Logikressourcen für eine in Hardware realisierte Schleife seien gegeben mit 60 Logikelementen (z.B. CLBs). Im Zielsystem stehe ein integrierter Prozessor zur Verfügung. Dieser basiere auf 4000 Logikelementen. Es ist davon auszugehen, dass Schleifen in Hardware um den Faktor 4 schneller laufen als in Software. Stellen Sie die Kostenfunktion auf und geben Sie an ab welcher Schleifenanzahl (unrolled) die Kostenfunktion hinsichtlich Logikressourcen und Geschwindigkeit minimal wird.

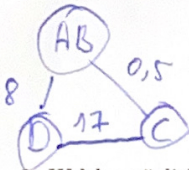
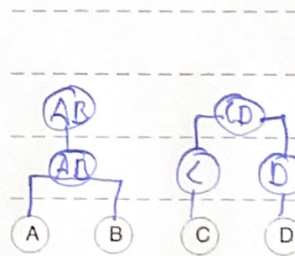
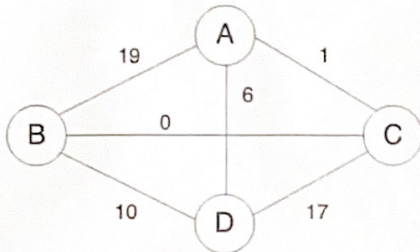
$C_{n, 4000 LEs}$
 $60LEs$
 \Rightarrow ~~schleifen~~

$$K(C, L) = k_1 \cdot h_C(C, C) + k_2 \cdot h_L(L, L)$$

$k_1 = n \cdot 60LE$ $k_2 = n \cdot \frac{1}{4}$

min Kosten min Zeit max Zeit

2. Wenden Sie die Methode des hierarchischen Clusterings zur Lösung eines Partitionierungsproblems auf das gezeigte Diagramm an. Wählen Sie für die Zusammenfassung jeweils die Knoten mit der größten Closeness als Distanzmaß.



3. Welche möglichen Probleme ergeben sich beim hierarchischen Clustering? Gehen Sie insbesondere darauf ein, welchen Einfluss die Struktur der Daten oder das gewählte Distanzmaß auf die Ergebnisse haben kann.

Da die Closeness oft nach ~~Ordnung~~ der gemeinsamen Nutzung von Ressourcen ~~und~~ oder den Verbindungen zweier Blöcke errechnet wird, beeinflusst die Datenstruktur maßgeblich die Distanz.