Bitte hier unbedingt Matrikelnummer und Adresse eintragen, sonst keine Bearbeitung möglich.

Postanschrift: FernUniversität, D-58084 Hagen

Name, Vorname

Straße, Nr.

PLZ, Wohnort

FERNUNIVERSITÄT
in Hagen
EINGANG

INF

FERNUNIVERSITÄT
in Hagen
D-58084 Hagen

Fachbereich Informatik

Kurs: 01709 „Technische Informatik III"

Nachklausur am 18.09.2004

Zutreffendes unbedingt ankreuzen !

Hörerstatus:

- ☐ Vollzeitstudent
- ☐ Teilzeitstudent
- ☐ Zweithörer
- ☐ Gasthörer
- ☐ Bachelor
- ☐ Lehramt
- ☐ . . . . . .

Klausurort:

- ☐ Berlin
- ☐ Bochum
- ☐ Frankfurt
- ☐ Hamburg
- ☐ Karlsruhe
- ☐ Kassel
- ☐ Bonn
- ☐ München
- ☐ Bregenz
- ☐ Wien
- ☐ . . . . . .

Aufgabe	1	2	3	4	5	6	Summe
erreichbare Punktzahl	30	3	23	12	9	23	100
bearbeitet							
erreichte Punktzahl							

Note: _____

Hagen, den _____ Betreuer: _____

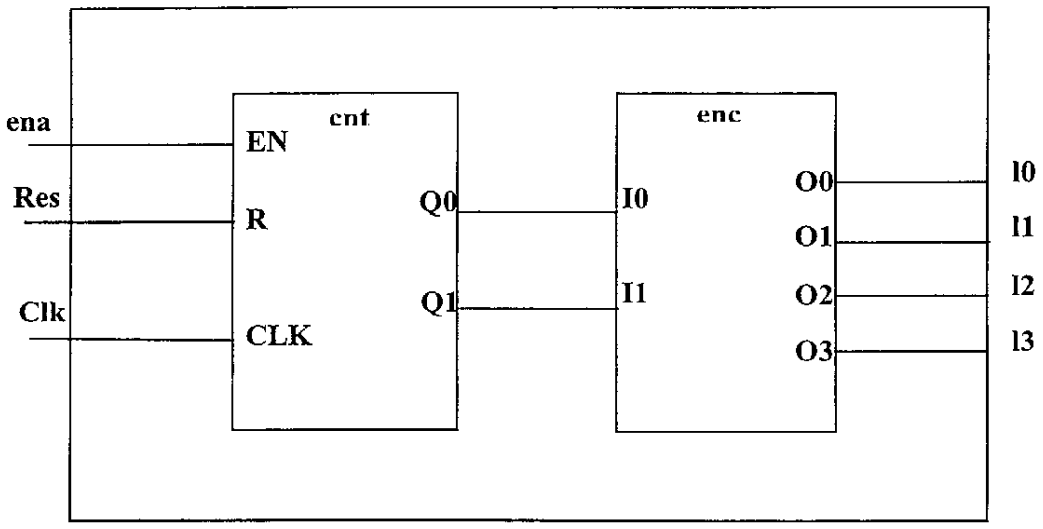
©2004 FernUniversität in Hagen

Aufgabe 1 (Grundlagen) 30 *Punkte*

Es soll ein VHDL-Design entwickelt werden, welches ein Eingangssignal zählt und mit Hilfe von 4 LEDs, LED1 bis LED4 den aktuellen Zählerstand anzeigt:

- Der Eingangstakt **clk** soll von 0 bis 3 gezählt werden (steigende Taktflanken).
- Bei Erreichen des Zählerstands 3 wird bei der nächsten Taktflanke automatisch auf 0 zurückgesetzt.
- Das Signal **res** dient zum asynchronen Zurücksetzen des Zählers (bei R='1').
- Das Signal **ena** dient zum Ein- (bei EN='1') und Ausschalten des Zählers.
- Die Ausgabesignale **l0-l3** dienen zum Ansteuern der LEDs.

Es wurde folgendes Schaltungsdesign entworfen:



4P

a) Geben Sie die Entities für den Zähler **cnt** und für den Encoder **enc** an!

16P

b) Entwerfen Sie eine Verhaltensbeschreibung für den Zähler **cnt**.

Hinweise:

- steigende Taktflanken lassen sich mit folgender Bedingung abfragen: *if (CLK'event and CLK = '1') then...*
- Der Zähler lässt sich am besten mit Hilfe eines Zustandsautomaten beschreiben.
  Variable: state = 0, state = 1...
- Die Ausgabe (Q0, Q1) ist binär codiert.

10P

c) Die 4 LEDs sollen unter folgenden Bedingungen angeschaltet werden:

LED1 (l0) = An, wenn Zählerstand = 0;
LED2 (l1) = An, wenn Zählerstand = 1;
LED3 (l2) = An, wenn Zählerstand = 2;
LED4 (l3) = An, wenn Zählerstand = 3;

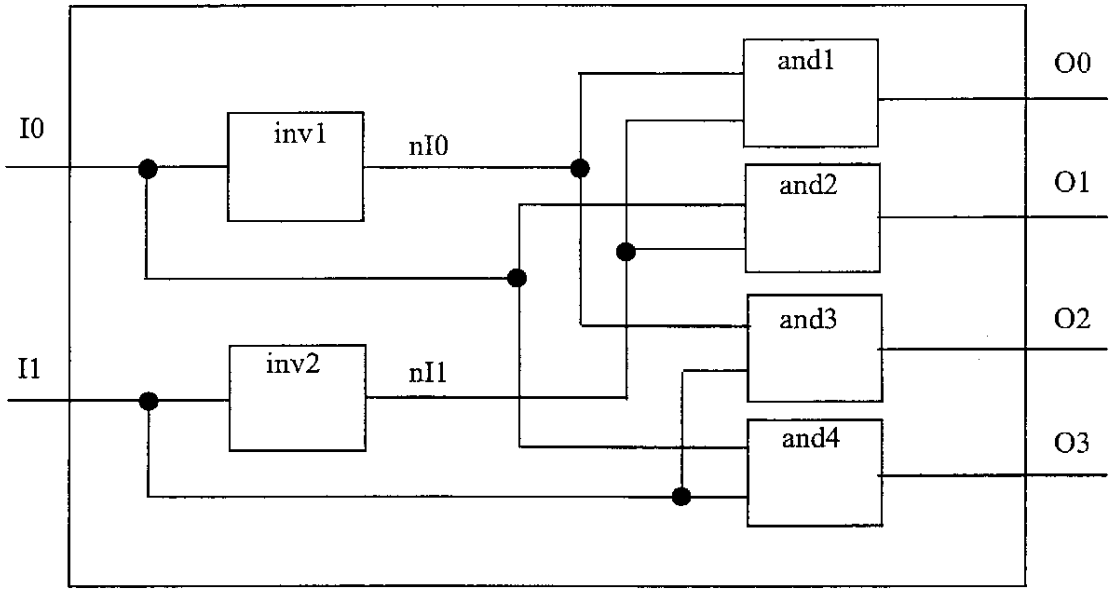
Für die Entwicklung von stehen zwei Typen von Bausteinen (Inverter und logisches AND) zur Verfügung:

```
ENTITY and IS
     PORT (a, b : IN  bit;
           y    : OUT bit);
END and;

ENTITY inv IS
     PORT (a    : IN  bit;
           y    : OUT bit);
END inv;
```

Geben Sie eine Strukturbeschreibung für den Encoder **enc** an!

Folgendes Schaltbild realisiert einen korrekten Encoder:



Aufgabe 2 (Grundlagen) 3 *Punkte*

1P

a) Wofür steht die Abkürzung EPIC?

2P

b) Was unternehmen der Programmierer, bzw. der Compiler, um parallele Ausführung bei EPIC zu erreichen?

Aufgabe 3 (Pipelining) *23 Punkte*

Gegeben sei folgendes DLX-Programm:

```
; Pipeline-Konflikt Testprogramm
main:
S1:         add         r1,r1,8
S2:         beqz        r1,Finish
S3:         add         r1,r1,3
S4:         multf       f2,f1,f4
S5:         multf       f1,f8,f4
S6:         multf       f2,f1,f2
S7:         multf       f4,f11,f4
S8:         multf       f5,f12,f4
S9:         addf        F1,F2,F3
S10:        movf        F1,F4
Finish:
S11:        trap        0
```

Geben Sie den Datenabhängigkeitsgraphen für o.a. Programm an. Kennzeichnen Sie im Abhängigkeitsgraphen die unterschiedlichen Arten dabei durch die Bezeichnungen δ^t, δ^a, δ^o wie im Kurstext. S11, bzw. Trap #0 soll dabei nicht in die Untersuchung mit eingehen.

Aufgabe 4 (Cache) 12 *Punkte*

5P

a) Erläutern Sie die Begriffe:
   - n-Wege-assoziative Organisation,
   - voll-assoziative Organisation und
   - direkt abbildende Organisation.

1P

b) Was versteht man unter „LRU"?

2P

c) Wie kann man sie implementieren?

2P

d) Diskutieren Sie (wenn Vorhanden) die Vor- und Nachteile großer Cache-Line-Größen.

2P

f) Wann bezeichnet man eine Cache-Speicherverwaltung als **cache-kohärent?**

Aufgabe 5 (MESI-Protokoll) 9 Punkte

Gegeben sei ein speichergekoppeltes Multiprozessorsystem mit zwei Prozessoren, die über einen Bus mit einem globalen Speicher verbunden sind. Zur Wahrung der Cache-Kohärenz wird das MESI Protokoll verwendet. Die Caches der beiden Prozessoren haben je eine Größe von drei Cache-Lines, die genau ein Speicherwort aufnehmen können. Die Caches werden von der niedrigsten Cache-Line aufwärts gefüllt, falls noch freie Lines zur Verfügung stehen. Im Falle eines voll besetzten Caches werden sie nach der Strategie LRU (least recently used) überschrieben. Ergänzen Sie die folgende Tabelle, wobei die Buchstaben die Zustände M = exclusive modified, E = exclusive unmodified, S = shared unmodified, I = invalid repräsentieren. Die Zahlen hinter den Aktionen und Zuständen bezeichnen Speicheradressen.

		Cache 1			Cache 2		
Prozessor	Aktion	Line 1	Line 2	Line 3	Line 1	Line 2	Line 3
-	-	E/8	E/12	I/-	E/6	I/-	I/-
1	read 18						
2	write 18						
1	write 8						
2	read 10						
1	read 10						
2	read 8						
1	write 8						
2	write 10						

Aufgabe 6 (Multiprozessorsysteme) 23 Punkte

3P

a) Bei der Nachrichtenübertragung zwischen nicht benachbarten Sender- und Empfängerknoten müssen in den Zwischenknoten Puffer eingerichtet sein, um Nachrichten, Pakete oder zumindest einzelne Übertragungseinheiten von Paketen zwischenzuspeichern. Bei der Zwischenspeicherung muss der Knoten dafür sorgen, dass sein Puffer nicht überläuft. Dies geschieht durch eine zusätzliche **Flusssteuerung** (*flow control*). Je nach Organisation der Flusssteuerung kann man verschiedene Übertragungsmodi unterscheiden.
Nennen Sie die drei Übertragungsmodi aus dem Kurstext.

4P

b) Gegeben sei ein Programm. Es benötigt auf einem Einprozessorsystem die Ausführungszeit T(1) = 1000. Dieses Programm wird auf ein Multiprozessorsystem portiert. Es kann entweder mit 10 oder mit 20 Prozessoren abgearbeitet werden und benötigt hierfür T(10) = 650 bzw. T(20) = 625. Berechnen Sie mit Hilfe von Amdahls Gesetz den prozentualen Anteil a des nur sequentiell ausführbaren Programmteils in beiden Fällen.
Hinweis: Für eine einfachere Rechnung geben Sie a bitte als Bruch an.

2P

c) Weshalb erhalten Sie unterschiedliche Werte für a?

3P

d) Berechnen Sie den Speedup S(10) und S(20). Berechnen Sie den maximal erreichbaren Speedup unter der Annahme a=0,6 (a=3/5).
Hinweis: Für eine einfache Rechnung geben Sie Ihre Lösungen bitte als Bruch an.

10P

e) Berechnen Sie die Effizienz E(10) und E(20). Wie viele Prozessoren können bei a=0,6 (a=3/5) maximal eingesetzt werden bevor die Effizienz unter 1/2 sinkt?

1P

f) Wie beurteilen Sie die Parallelisierbarkeit des Programms?

– ENDE –