

Einführung in die Technische Informatik			14.05.2008
<input type="radio"/> Informatik (120 Minuten) <input type="radio"/> UF Informatik und Informatikmanagement (120 Minuten) <input type="radio"/> Wirtschaftsinformatik (90 Minuten)			
Kennnr.	Matrikelnr.	Nachname	

1	[4]	[]
2	[2]	[]
3	[9]	[]
4	[4]	[]
5	[6]	[]
6	[10]	[]
7	[12]	[]
8	[4]	[]
9	[6]	[]
10	[2]	[]
11	[2]	[]
12	[2]	[]
13	[9]	[]
14	[6]	[]
15	[12]	[]
16	[4]	[]
17	[3]	[]
18	[3]	[]
Summe	[100]	[]
Note		

Deckblatt mit Kugelschreiber sofort ausfüllen!

Bitte deutlich schreiben, unleserliche Antworten werden nicht gewertet!

Buch, Mitschriften, Ausdrücke von Folien sind nicht zugelassen!

Wirtschaftsinformatiker starten mit Frage 6!

1. Ist es einfacher, eine disjunktive oder eine konjunktive Aussageform in ein in NAND/NAND-Struktur aufgebautes PLA zu übertragen? Begründen Sie Ihre Antwort. (4 Punkte)
2. Welche Möglichkeiten kennen Sie, Verbindungen in einem IC nach der Fertigung zu trennen bzw. zu schließen? Wodurch unterscheiden sich diese? (2 Punkte)
3. Sie haben einen 3-zu-8 Binärdecoder, 8 NAND-Gatter mit jeweils zwei Eingängen und Open-Collector-Ausgang sowie ein NOT-Gatter zur Verfügung. Konstruieren Sie aus diesen Bausteinen einen 8-auf-1-Digitalmultiplexer. Achten Sie auf vollständige Darstellung der Bausteine und Anschlüsse (Hinweis: Verdrahtete Logik!). (9 Punkte)

4. Nennen Sie je 2 Beispiele für Tabellen- und Funktionsspeicherbausteine. (4 Punkte)

5. Welchen Baustein beschreibt der folgende VHDL-Code? Geben Sie die Funktionen aller Ein- und Ausgänge genau an! (6 Punkte)

```
library ieee;
use ieee.std_logic_1164.all;

entity mystery is
  port(C, SI : in std_logic;
        SO : out std_logic;
        PO : out std_logic_vector(7 downto 0));
end mystery;

architecture behav of mystery is
  signal tmp: std_logic_vector(7 downto 0);
begin
  process (C)
  begin
    if (C'event and C='1') then
      for i in 0 to 6 loop
        tmp(i+1) <= tmp(i);
      end loop;
      tmp(0) <= SI;
    end if;
  end process;
  SO <= tmp(7);
  PO <= tmp;
end behav;
```

Anmerkung: Das Attribut 'event' eines VHDL-Signals ist TRUE, wenn für dieses Signal ein Zustandswechsel eingetreten ist.

6. Es soll ein Mikro-Programm für den Mikro16 Prozessor entworfen werden, das eine Bitfolge um $n \geq 0$ Stellen *arithmetisch* nach links verschiebt. Die zu shiftende Bitfolge liegt in Register R4. Die Anzahl auszuführender arithmetischer Shift-Operationen ist in Register R5 abgelegt. Initial sind die Register R0 ist mit der Konstante 0, R1 mit der Konstante 1 und R2 mit der Konstante -1 belegt. Zusätzlich enthält Register R10 den Wert $(8000)_{16}$. (10 Punkte)

Mikro16-Programm:

7. Gegeben sind drei Versuche zur Synchronisation von zwei Prozessen *A* und *B*. Prozess-Synchronisation heißt in diesem Fall, die Codestücke müssen sicherstellen, dass sich höchstens ein Prozess (entweder *A* oder *B*) zur selben Zeit im kritischen Abschnitt `critical_section()` befindet. Zusätzlich darf auch kein Deadlock auftreten. Dabei sind `turn`, `lock`, `lockA`, und `lockB` gemeinsame (shared) Variablen der jeweiligen Prozesse *A* und *B*. Für jeden der drei Versuche: Begründen Sie kurz, warum der Versuch korrekt ist oder erklären Sie, wo der Fehler liegt. (12 Punkte)

Hinweis: Es kann sich sowohl um Fehler handeln, die dazu führen, dass beide Prozesse gleichzeitig im kritischen Abschnitt laufen, als auch um Fehler, die zu einem Deadlock führen.

(a) Versuch 1 (4 Punkte):

```
Init
int turn = 1;

Prozess A
while (turn != 1) { }; // wait
critical_section(); // kritischer Abschnitt
turn = 2;

Prozess B
while (turn != 2) { }; // wait
critical_section(); // kritischer Abschnitt
turn = 1;
```

Kommentar/Begründung:

(b) Versuch 2 (4 Punkte):

```
Init
boolean lock = false;

Prozess A
while (lock == true) { }; // wait
lock = true;
critical_section(); // kritischer Abschnitt
lock = false;

Prozess B
while (lock == true) { }; // wait
lock = true;
critical_section(); // kritischer Abschnitt
lock = false;
```

Kommentar/Begründung:

(c) Versuch 3 (4 Punkte):

```

Init
boolean lockA = false;
boolean lockB = false;

Prozess A
lockA = true;
while (lockB == true) { }; // wait
critical_section(); // kritischer Abschnitt
lockA = false;

Prozess B
lockB = true;
while (lockA == true) { }; //wait
critical_section(); // kritischer Abschnitt
lockB = false;

```

Kommentar/Begründung:

8. Geben Sie die Werte in allen gezeigten Registern und Speicherzellen im Endzustand an, die sich aus dem Anfangszustand nach der Abarbeitung der folgenden Instruktion *I* ergeben. (4 Punkte)

Instruktion *I*: $R12 \leftarrow \text{memory}[\text{memory}[R13] + (F3)_{16}]$

Anfangszustand			
Reg.	Inhalt	Adresse	Inhalt
...	...	$(4711)_{16}$	$(1704)_{16}$
R12	$(\text{CAFE})_{16}$
R13	$(\text{BEEF})_{16}$	$(4804)_{16}$	$(1234)_{16}$
...
		$(\text{BEEF})_{16}$	$(4711)_{16}$
		$(\text{BEF0})_{16}$	$(4804)_{16}$

Endzustand			
Reg.	Inhalt	Adresse	Inhalt
...	...	$(4711)_{16}$	
R12	$6C60$
R13	$000F$	$(4804)_{16}$	
...
		$(\text{BEEF})_{16}$	
		$(\text{BEF0})_{16}$	

4711
4804

9. Gegeben ist das Klasse C Netzwerk mit der Adresse 192.168.11.0. Teilen Sie dieses Klasse C Netzwerk so auf, dass

- (a) Subnetze für jeweils zumindest 104 Hosts entstehen.
- (b) Subnetze für jeweils zumindest 24 Hosts entstehen.

Geben Sie die jeweils maximal mögliche Anzahl der Subnetze und die jeweiligen Subnetzmasken an. (6 Punkte)

(a) max. Anzahl Subnetze mit jeweils 104 Hosts: 2

Subnetzmaske: ~~255.255.255.128~~ 255.255.255.128

(b) max. Anzahl Subnetze mit jeweils 24 Hosts: 8

Subnetzmaske: 255.255.255.224

10. Was versteht man unter einer *Bus Arbitration Logic* (bzw. einem Busarbitrier)? (2 Punkte)

11. Wann wird ein Programm (oder eine Funktion) *recurrent* genannt? (2 Punkte)

12. Was versteht man unter dem Begriff *frequency hopping*? Bei welchem drahtlosen Übertragungsverfahren wird *frequency hopping* verwendet? (2 Punkte)

13. Ein Speicherverwaltungssystem mit Paging verwendet 32-bit virtuelle Adressen und 24-bit physikalische Adressen. Die Größe einer Page beträgt 8 KB (8192 Bytes). Gegeben ist folgende Pagetable:

Page-Nummer	Frame-Nummer
$(1)_{16}$	(not present)
$(2)_{16}$	(not present)
$(3)_{16}$	$(1B)_{16}$
...	...
$(E9)_{16}$	(not present)
$(EA)_{16}$	$(207)_{16}$
$(EB)_{16}$	(not present)
$(EC)_{16}$	$(7C)_{16}$
$(ED)_{16}$	(not present)
$(EE)_{16}$	$(380)_{16}$
...	...

Wandeln Sie mit Hilfe der Pagetable die folgenden 3 virtuellen 32-bit Adressen in die entsprechenden physikalischen Adressen um. Geben Sie dabei jeweils an, welche Page-Nummer zum Einsatz kommt oder ob ein *page fault* auftritt! (9 Punkte)

(a) $(001DDIAB)_{16}$:

(b) $(00004A02)_{16}$:

(c) $(000078BF)_{16}$:

14. Skizzieren Sie ein 2-fach *interleaved memory* mit einer Gesamtspeichergröße von 2K! (6 Punkte)

(d) RR (Round Robin) mit Zeitquantum $q = 4$ Einheiten (4 Punkte)

Prozess	Laufzeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
A	4																					
B	8																					
C	9																					
D	6																					
E	2																					

16. Geben Sie die 4 notwendigen Bedingungen für die Entstehung von Deadlocks an! (4 Punkte)

-
-
-
-

17. Nennen Sie drei Ausgänge zur Statusanzeige einer ALU. (3 Punkte)

-
-
-

18. Was versteht im Bereich Sicherheit unter dem Begriff *Interruption*? Geben Sie zwei Beispiele für Angriffe an, die in diese Kategorie fallen! (3 Punkte)