1704-06 9/97

Aufgabe K-1

Man kann mit Pertrinetzen die Aufenthaltsdauern einer Marke in einer Stelle P_1 genähert nach Bild K-1 „messen".

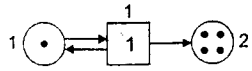

Bild K-1

a) Erläutern Sie Bild K-1, insbesondere auch den maximalen Zeitmeßfehler.
(1 Punkte)

Betrachten Sie zwei unabhängig reparierbare Komponenten (*Rechner*), die einzeln mit gleicher Intensität an einer Aufgabe arbeiten, für die ein Rechner die Dauer **d** brauchen würde. Zeichnen Sie für folgende Fälle die entsprechenden Petrinetze auf:

b) Wie man die Gesamtrechenzeit (seit dem Einschalten des Systems) messen kann,
(5 Punkte)

c) wie man erreicht , daß eine *Alarmstelle* markiert wird, wenn die Dauer **d** zu spät zustande käme; genauer gesagt nach der Zeit **d/2+z**, wobei **z** die Zeitreserve für kurze Ausfälle ist.
(4 Punkte)

Aufgabe K-2

In ein System von acht würfelförmig vernetzten Rechnern soll ein möglichst „großer" Binärbaum eingebettet werden.

a) Zeichnen Sie räumlich sowie planar ein Beispiel mit möglichst vielen Blättern, wobei im ersteren Bild (Würfel) die Kanten des Baumes fett oder farbig hervorzuheben sind und im letzteren Bild (Binärbaum) nur die o.g. fetten Kanten vorhanden sein sollen.
(3 Punkte)

b) Erläutern Sie, warum diese Bäume im wesentlichen gleich aussehen.
(2 Punkte)

c) In der Baumspitze soll der Rechner dank einiger interner Redundanz die Unverfügbarkeit U^2 haben, alle anderen Knotenrechner sollen die Unverfügbarkeit U haben. Die Kanten (Verbindungsleitungen) sind ideal zuverlässig.
Wie groß ist in quadratischer Näherung (in U) die Systemunverfügbarkeit U_s, wenn nur die Endknoten (Blätter des Baumes) gemeinsam mit dem Rechner an der Spitze nützliche Arbeit leisten können und zwar mindestens 2 von ihnen. Begründen Sie Ihre Rechnung ausführlich.
(5 Punkte)

Aufgabe K-3

Der Zustand der Reparaturwarteschlange eines 3-Komponentensystems sei durch Bild K-2 gegeben.

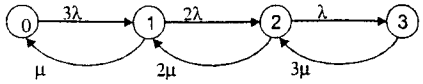

Bild K-2

a) Schreiben Sie die Differentialgleichungen für den Wahrscheinlichkeitsfluß auf.
(3 Punkte)

b) Lösen Sie diese Gleichung für den Fall der Stationarität, wenn $\dot{P}_i = 0$ für $i \in \{0,\ldots,3\}$.
Beachten Sie dabei die Normierungsregel $P_0+P_1+P_2+P_3=1$.
(4 Punkte)

c) Wie lautet die stationäre System-Unverfügbarkeit, wenn die drei Komponenten speziell ein 2-von-3-System bilden? Das Endergebnis soll als Polynom in der Form $U=\lambda/(\lambda+\mu)$ dargestellt werden.
(3 Punkte)

Aufgabe K-4

Schreiben Sie für einen 8bit-Prozessor mit einem Akkumulator A ein Mikroprogramm DIV, das die Division zweier 8bit-Zahlen ohne die explizite Verwendung eines Divisionbefehls durchführt. Verwenden Sie den in der Tabelle angegebenen Befehlssatz.

Der Mikroprozessor besitzt einen internen Speicher, der bei der Adresse X beginnt und bei X+25 endet.
Der Dividend und der Divisor stehen im Zwischenspeicher unter der Adresse X bzw. X+1.
Der Quotient und der Divisionsrest stehen im Arbeitsspeicher an den Adressen X+2 und X+3.
Dividend, Divisior, Quotient und Divisionsrest sind vorzeichenlose 8bit-Werte.
Eine Vergleichsoperation kann nur zwischem einem Register und einer Speicherzelle durchgeführt werden. Speicher-Speicher-Transferoperationen sind nicht möglich.

```
LOADA X          A <- M(X) (Ladebefehl)
STOREA X         A -> M(X) (Speicherbefehl)
SETA X           A := X (Zuweisungsbefehl)
SUBA X           A := A - M(X) (Subtraktionsbefehl)
ADDA X           A := A + M(X) (Additionsbefehl)
JUMP X           PC:= X (unbedingter Sprung)
JUMPN X          PC:=X, falls N-Flag gesetzt
CMPA X           N-Flag:=0  für A>=M(X)
                 N-Flag:=1  für A< M(X)
                 Verändert nicht den Wert des Akkumu-
                 lators !
```

a) Geben Sie zunächst das Flußdiagramm für die Routine an.
(4 Punkte)

b) Geben Sie das Mikroprogramm mit **ausführlichen** Kommentaren an.
(6 Punkte)

Aufgabe K-5

Entwerfen Sie eine Einheit für einen Mikroprozessor, die die Operationen *pusha* und *popa* auf dem Stack (Kellerstapelspeicher) überwacht.

Der Befehl *pusha* erhöht den Stackzeiger SP (Register des Mikroprozessors) und speichert den Wert des Akkumulator auf der durch den Stackzeiger SP adressierten Addresse ab. Der Befehl *popa* lädt den Wert der Speicherzelle, die durch den Stackzeiger SP adressiert wird, in den Akkumulator und erniedrigt danach den Stackzeiger SP. Der Stack wächst von der niedrigen zur hohen Adresse. Der Stackzeiger SP zeigt auf den zuletzt abgelegten Wert im Stack.

Die Einheit besitzt ein Register *OVERFLOW_LIMIT* (OVL), welches die oberere Grenze im Speicher für den Stack angibt und ein Register *UNDERFLOW_LIMIT* (UNL), welches die untere Grenze für den Stack angibt, d.h. die unterste Speicherzelle, die für die Ablage eines Wertes auf dem Stack gültig ist (siehe Bild K-3). Für den Fall, daß ein Wert oberhalb der Grenze *OVL* abgelegt wird, soll ein Signal *SIG_OVL* den Überlauf des Stacks anzeigen. Für den Fall, daß ein Wert unterhalb der Grenze *UNL* abgelegt wird, soll dementsprechend ein Signal *SIG_UVL* den Unterlauf des Stacks anzeigen.

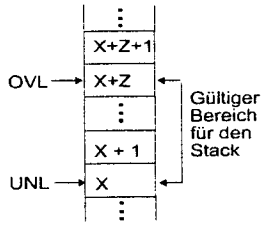

Bild K-3

a) Wie lauten die Bedingungen für einen vollen Stack, einen leeren Stack, Stacküberlauf und Stackunterlauf.
(4 Punkte)

b) Schreiben Sie für die Operation pusha und popa jeweils Mikroprogramme. Fügen Sie die Befehle check_ovl und check_unl an den passenden Stellen ein.
(2 Punkte)

c) Zeichen Sie ein Blockdiagramm für diese Einheit, die die Operationen auf dem Stack überwacht und die obengenannten Signale SIG_OVL und SIG_UVL erzeugt. Als Elemente stehen ihnen Addierer, Negierer, Halbaddierer, Volladdierer und Register zur Verfügung.
(4 Punkte)