Choice Coordination Problem

Annahmen: $k = 2$; $n = 2$; synchroner Fall;

Inhalt des i-ten Speicherplatzes: $\langle M_i, T_i \rangle$;

Protokoll für Prozessor P_i [lokale Variablen mem, b]

0. $mem := i$; $b := 0$;
1. **if** $M_{mem} = \checkmark$ **then**

    // ✓ wurde bereits geschrieben

    STOP;

    else if $M_{mem} = 0 \wedge b = 1$ **then**

    // Gelesen wird eine Null, zuletzt geschrieben wurde eine Eins

    $M_{mem} := \checkmark$; **STOP**;

    else

    // Schreibe zufälliges Bit

    $b :=$Random(0,1); $M_{mem} := b$;

3. $mem := 1 - mem$; zurück zu 1.

Choice Coordination Problem

Annahmen: $k = 2$; $n = 2$; asynchroner Fall;

Inhalt des i-ten Speicherplatzes: $\langle M_i, T_i \rangle$;

Protokoll für Prozessor P_i [lokale Variablen mem, b, t]

0. mem :=Random(0,1); $b := 0$; $t := 0$;
1. Warte auf R/W-Zugriffsberechtigung für M_{mem};
2. **if** $T_{mem} > t$ **then**

    // Speicherinhalt von M_{mem} ist neuer als der von P_i

    $t := T_{mem}$; $b := M_{mem}$;

   else if $T_{mem} < t$ **then**

    // Speicherinhalt von M_{mem} ist älter als der von P_i

    $M_{mem} := \checkmark$; **STOP**;

   else if $T_{mem} = t \wedge M_{mem} = 0 \wedge b = 1$ **then**

    // Gelesen wird eine Null, zuletzt geschrieben wurde eine Eins

    $M_{mem} := \checkmark$; **STOP**;

   else

    // Tritt nur ein, falls $T_{mem} = t$

    $t := t + 1$; $T_{mem} := t$; b :=Random(0,1); $M_{mem} := b$;

3. Löse Schreibsperre für M_{mem};

   $mem := 1 - mem$; zurück zu 1.