3.4. Größeres steuerungstechnisches Beispiel

Problemstellung PN30

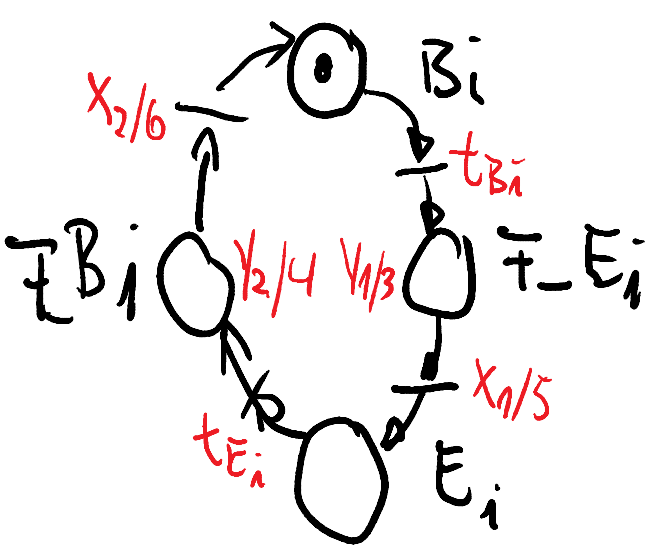
Sensor-, Aktor- und Zeiztbezeichnungen PN31
Binäre Eingänge für W1 und W2 -> x1 … x4 und x5 … x8 ind der gleichen Reihenfolge w.o.

Stehen, wenn kein Vor- oder Rückwärtsfahren

Abkürzungen für die Platznamen: PN32

Modell entwickeln: ungesteuertes System + Erweiterung zum gesteuerten System

1. Suche sequentielle Komponenten -> ZM
   ➔ W1, W2



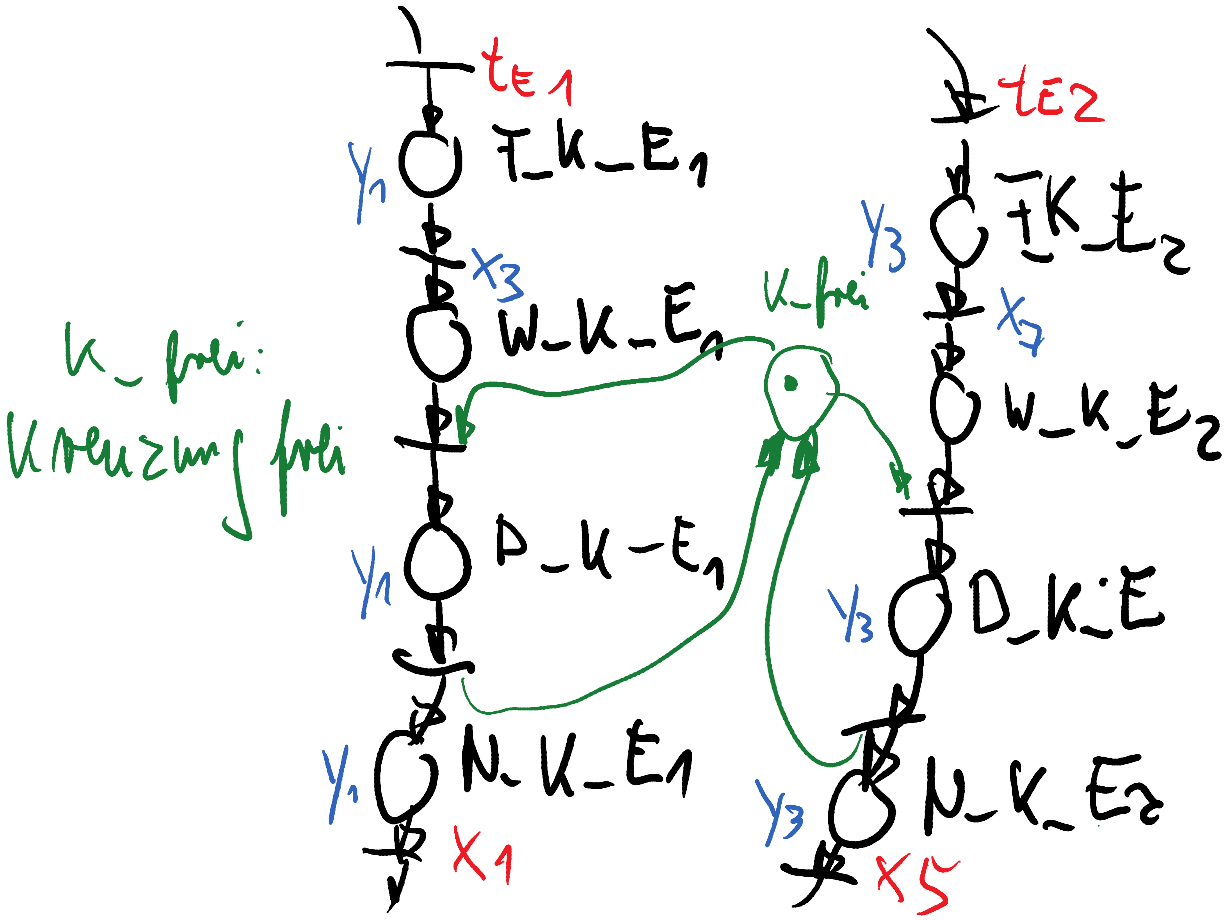
Bi: Beladen Wi
FB/Ei: Fahren Richtung Beladen/Entladen Wi
Ei: Entladen

2. wx, wy, wt zuordnen, soweit schon möglich

wt, wx, wy

3. Abhängigkeiten der sequentiellen Komponenten untereinander durch Koppelnetz (System von ZM)

(exklusive Kreuzungsnutzung)
Variante 1: es fährt immer nur ein Wagen -> nicht zeitoptimal
Variante 2: Aufteilen der Strecke



Richtung beladen i analog

4. wx, wy, wt anpassen bzw. ergänzen. -> y1/3 anpassen (Warteplatz !!)

Gesamtergebnis: PN33

5. Test mit Simulation

Gewollte Schaltfolgen möglich, nicht gewünschte m nicht möglich
Ja, aber mit Konflikt, falls beide Wagen gleichzeitig an der Kreuzung ankommen.
Exklusives Befahren der Kreuzung? Keine Verletzung gebracht.

Kein Beweis !!

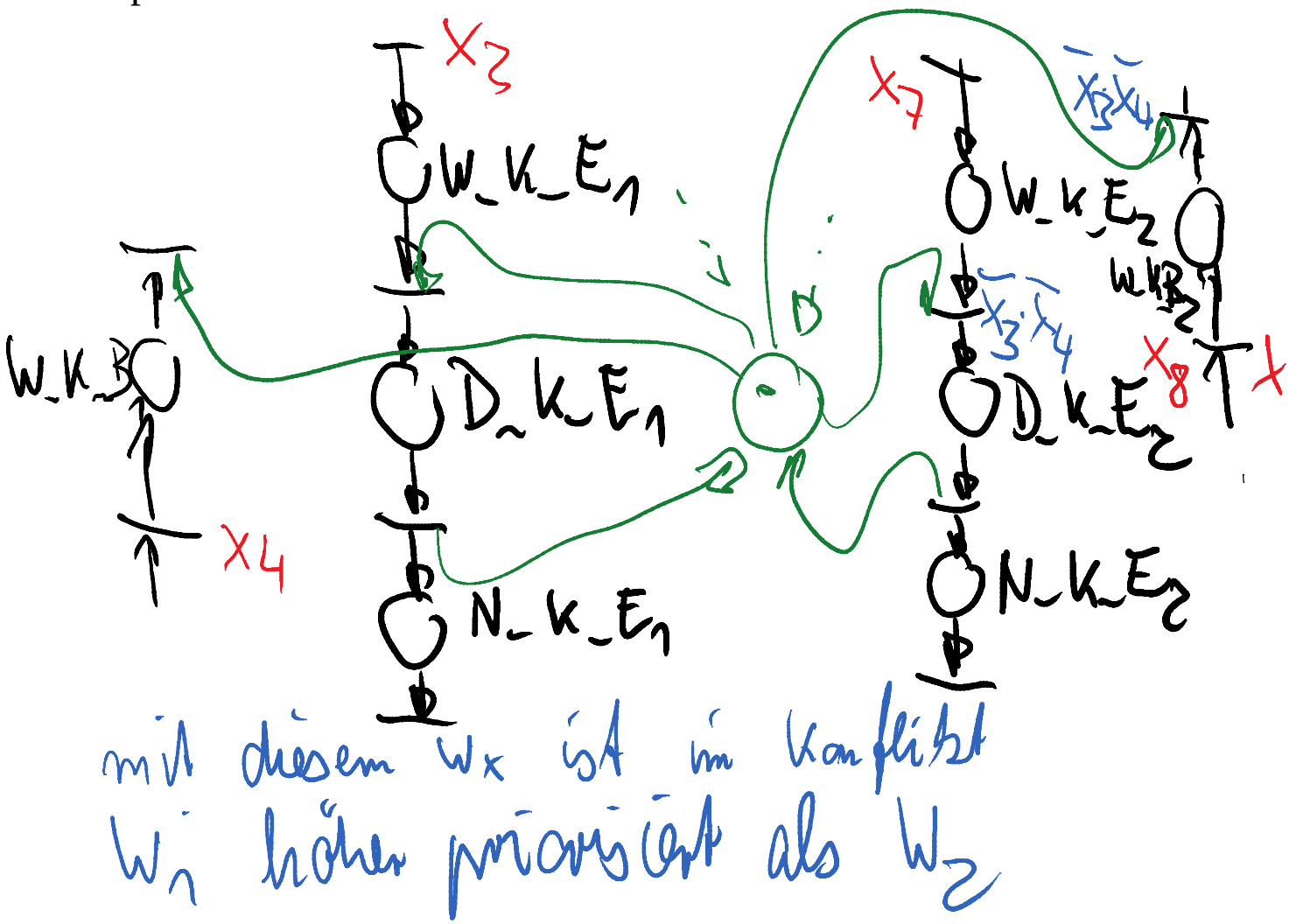
6. Formale Verifikation

Wichtig bei sicherheitskritischen Anwendungen (Personenschaden, großer ökonomischer Schaden bei Fehlfkt. möglich)

Ergebnisse wie bei 5., aber jetzt bewiesen (INA war nicht vorhanden, Verwendung frühere Verifikationsergebnisse)

7. Konflikt(e) lösen

Hier Bsp. über wx



8. Synthese eines Steuerprogramms aus dem PN

Bei sicherheistkritischen Anwendungen: verifiziertes Synthesetool, damit PN-Verhalten 100% erhalten bleibt und die verifizierten Eigenschaften erhalten bleiben.

Grundprinzip der softwaretechnischen Realisierung des PN-Verhaltens (auf einem Einprozessorsystem)