

## Parallele Automaten

Bsp. F2\_80

2 Sequentielle Automaten: Aut. 1,2

Takt: Synchrone Automaten

Reset: Erzeugen Anfangszustände von außen

2 logische (binäre) Eingänge  $X_{1,2}$

2 binäre logische Ausgänge  $Y_{1,2}$

2 Koppelsignale zwischen den beiden Automaten  $S_{1,2}$

Funktion der beiden Automaten + Kopplung werden als Automatengraphen beschrieben (F2\_90)

2 Mooreautomaten ( $Y_i$  und  $S_i$  nur zustandsabhängig)

Hier Verzicht auf Eigenschleifen. Es gilt: der aktuelle Zustand bleibt so lange erhalten, bis eine wegführende Kante aktiv ist (der Boolesche Ausdruck an ihr ist gleich 1)

$X_i$  und  $y_i$  wie normaler sequentieller Automat

Kopplung:  $S_i$  sind Ausgänge (zustandsabhängig) von Automaten und gleichzeitig Eingänge (erscheint in den Booleschen Ausdrücken der Kanten, wechselseitig E und A in unterschiedlichen Automaten)

Erläuterung Automatenverhalten zu Zeitverläufen:

(Zeitverläufe zeitabhängige Veränderung von Eingängen Ausgängen, Koppelsignale und Zuständen in Abhängigkeit vom Takt)

$X_i$  werden von außen variiert, geeignetermaßen so, dass die Automaten alle Zustände und Kanten durchlaufen

$Y_i$ ,  $S_i$  und  $Z_i$  (hier mit Automat  $i$  bezeichnet) ergeben sich dann aus den  $X_i$  und den Automaten, beschrieben durch die Automatengraphen)

Ablauf:

A1 in A1\_1:  $y_1=1$

A2 in A2\_1:  $y_2=1$

$X_1, X_2 = 0$ ,  $S_1, S_2 = 0$  -> keine Kante in A1,2 schaltfähig

(schaltfähig bedeutet: Boolescher Ausdruck der Kante ist = 1, zugeordneter Zustandsübergang ist mgl. (wie in RO bezeichnet ?? siehe Skript dort))

$X_1:=1$  ->  $K(A1_1, A1_2)$  schaltfähig und schaltet (SfuS) mit der nächsten aktiver Taktflanke (naTF):  $y_{1,2}$  bleiben 1,  $S_1$  wird 1

- ➔  $K(A1_2, A1_3)$  sfus, da Ausdruck an ihr immer = 1 und
- ➔  $K(A2_1, A2_2)$  sfus, da  $S_1=1$
- ➔  $Y_1=0$ ,  $Y_2=0$ ,  $S_1$  bleibt 1

$K(A1_3, A1_4)$  nicht schaltfähig, da  $S_2=0$

$K(A2_2, A2_3)$  nicht schaltfähig, da  $X_1=1$

$X_1:=0$  ->  $K(A2_2, A2_3)$  sfus

- ➔  $S_2=1$
- ➔  $K(A1_3, A1_4)$  sfus, da  $S_2=1$
- ➔  $K(A2_3, A2_1)$  nicht schaltfähig, da  $X_2=0$

$X_2:=1$  ->  $K(A2_3, A2_1)$  sfus

- ➔  $Y_1=1$ ,  $Y_2=1$ ,  $S_2=0$ ,  $S_1$  bleibt 0

Weiter Anfangszustände von Aut.1,2, wiederholung, auch modifiziert mgl.

3. Prozessor- und Prozessorzugeordnete Baugruppen

### 3.1. Grundarchitekturen

F3\_10

Harvard- und Princeton-Architektur: benannt nach den Universitäten, in denen die Architekturen entwickelt und erstmalig angewendet wurden. (Ende der 1940iger Jahre)

4 Blöcke, in beiden Architekturen:

1. Prozessor (Verarbeitungseinheit, Central Processing Unit, CPU)
  - Erzeugen von Ergebnissen aus Operanden nach vorgegebenen Operationen (Befehle)
  - Auswahl von Operanden und Steuerung der Reihenfolge der Operationen
  - Lesen von Befehlen und Operanden (Eingangsoperanden, E-Operanden) und Schreiben von Ergebnissen (Ausgangsoperanden, A-Operanden) von und zum Speicher bzw. Ein-/Ausgabe
  
2. Programmspeicher (Program Memory)
  - Speicher für auszuführende Operationen (Befehle, genauer: Maschinenbefehle: Masch. Befehle sind Binärcodes, die der Prozessor direkt versteht und damit abarbeiten kann)
  - Bereitstellen von ausgewählten Befehlen (Lesen von Befehlen)
  
3. Datenspeicher (Data Memory)
  - Speichert Zwischenoperanden
  - Bereitstellen von ausgewählten Operanden (Lesen von Operanden)
  - Übernahme von Ergebnissen auf ausgewählte Speicherplätze (Schreiben von Operanden)
  
4. Ein- und Ausgabe (Input/Output)
  - Schnittstelle zur Umwelt für Ein- und Ausgabe-Operanden
  - Auswahl, Lesen und Schreiben dieser E/A-Operanden

Bus:

Lt. Def. Öffentliches Personennahverkehrsmittel

Öffentlich: mehrere Teilnehmer

Nahverkehrsmittel: kurze Strecken, variables Ein- und Aussteigen

- Im Rechner:

Transportmittel: von Daten, Befehle

Öffentlich: verschiedene Baugruppen benutzen den Bus

Nah: rechnerintern

Variabel: Beziehungen zwischen den Baugruppen

Prinzip von Bussen (F3\_30)

- Hier Kommunikation zwischen Prozessor und den anderen Blöcken (Pr./Dat.-Speicher, EA, Blöcke evtl. in mehrere Unterblöcke aufgeteilt)
- Datenbus: Überträgt Befehle und Operanden
- Adressbus: Überträgt Auswahlinformation für Befehle und Operanden (siehe auf F3\_20)
- Steubus: sind spezielle Steuersignale zur Synchronisation des Datenaustausches binär)

Adressbus unidirektional: immer vom Prozessor zu den Blöcken

Datenbus bidirektional: vom Prozessor zu den Blöcken und umgekehrt

Steubus: Einzelsignale, einzelne vom Prozessor zu den Blöcken und einzelne von den Blöcken zu Prozessor

Im Weiteren Adressierung und konkrete Signalbelegung auf Bussen