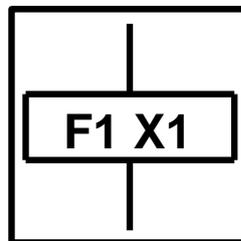


SIMULINK Erweiterungsblockbibliothek



Funktionsplan

**FUP Blockbibliothek V. 2.3
alle Rechte vorbehalten
Dr. G.-H. Geitner ETI/TU Dresden**

**Lehrstuhl Automatisierte Elektroantriebe
Prof. Dr.-Ing. habil. Dr.-Ing. E. h. R. Schönfeld**

Inhalt

◆ Einführung.....	3
◆ Wesentliche Eigenschaften der SIMULINK Blockbibliothek Funktionsplan	4
◆ Anwendungshinweise zur Modellierung des Bewegungsablaufes	6
◆ Die Blockbibliothek	11
◆ Beispiel 1: Zwei Alternativzweige	12
◆ Beispiel 1a: Zwei Alternativzweige mit Rückführung nicht über Anfangszustand	13
◆ Beispiel 2: Zwei Parallelzweige	14
◆ Beispiel 2a: Zwei Parallelzweige mit Rückführung nicht über Anfangszustand	15
◆ Beispiel 2b: Parallelverzweigung ohne Ereignis im Parallelzweig	16
◆ Beispiel 3a: Verschachtelte Verzweigungen I	17
◆ Beispiel 3b: Verschachtelte Verzweigungen II	18
◆ Beispiel 4: Anwendung des Kommentarblockes	19
◆ Beispiel 5: Gesamtsystem mit einem Funktionsplan, zwei Antrieben und Zeitgebern	20
– Bild 1: Sollbewegungsablauf.....	20
– Bild 2: Gesamtstruktur	21
– Bild 3: Antriebe (Prototyp der Bewegungszelle)	22
– Bild 4: Prozeßanpassung (Ereignisgenerierung)	23
– Bild 5: Funktionsplan (FUP)	24
– Bild 6: Simulationsergebnisse (Zustandsgrößen / Ereignisse)	25
◆ Beispiel 6: Gesamtsystem mit Teilfunktionsplänen, drei Antrieben, logischen Gleichungen für Ereignisse und Sammelfehlerbehandlung	26
– Bild 1: Gesamtsystem	27
– Bild 2: Prozeßanpassung (Ereignisgenerierung)	28
– Bild 3/4: Teilfunktionspläne Referenzpunktfahrt / Anlauf	29
– Bild 5/6: Teilfunktionspläne Normalbetrieb / Stillsetzen	30
– Bild 7/9: Simulationsergebnisse	31
◆ Beispiel 7: Einfache verzweigte Fehlerbehandlung	32
◆ Blockbeschreibungen (Kürzel / Blockname in Maske / Erläuterungen (Blockbeschreibungen / Blockhilfe))	33

WebLink: http://eeiwzg.et.tu-dresden.de/ae2_files/ae_7.htm (Homepage Funktionsplan)

Einführung

Beschreibung

Für ereignisgesteuerte Systeme mit mehrdimensionalen Bewegungsabläufen wurde im Fachausschuß 4.20 "Bewegungssteuerungen für Be- und Verarbeitungsmaschinen" der VDI/VDE-Gesellschaft Meß- und Automatisierungstechnik (GMA) die VDI/VDE-Richtlinie 3684 "Beschreibung ereignisgesteuerter Bewegungsabläufe mit Funktionsplänen" als interdisziplinär verständliches und anschauliches Dokumentationsmittel und Entwurfswerkzeug erarbeitet. Die Richtlinie ist über den Beuth Verlag Berlin verfügbar.

Simulation

Mit einer Blockbibliothek Funktionsplan auf Grundlage der in der Richtlinie 3684 vorgeschlagenen Systembeschreibung ist ein durchgängiger Entwurfsweg bis zur Überprüfung des Systemverhaltens ereignisgesteuerter Systeme mittels Simulation einschließlich frühzeitiger Erkennung von Fehlern im Entwurfsprozeß möglich. Die FUP Blockbibliothek liegt als Erweiterungs-Blockbibliothek für SIMULINK vor und wurde hinsichtlich ihrer wesentlichen Eigenschaften zur Simulation von ereignisgesteuerten Prozessen speziell in mechatronischen Systemen entwickelt.

Umfang

In Version 2.2 umfaßt die Bibliothek 25 Blöcke, 9 S-Funktionen und 6 MEX-Files in C-Kode. Die Blockikonen gewährleisten nach kurzer Einarbeitungszeit einen sicheren Einsatz der Blöcke, damit kann auf Blockkürzel im Sinne von Richtlinie und Übersichtlichkeit verzichtet werden. Eine Blockhilfe ist für alle Blöcke On-line verfügbar. Die Modellierung von Bewegungsabläufen ist schon mit wenigen Anwendungshinweisen möglich.

Beispiele

Die Anwendung der Blöcke veranschaulichen 10 Beispiele mit u.a. Parallel- und Alternativverzweigungen, Rückführungen ohne den Anfangszustand, Definition von Teilfunktionsplänen, "Ansteuerung" von Bewegungszellen (Antrieben) mit Simulation von Regelschleifen, Ereignisgenerierung und Fehlerbehandlung.

Vorteile

Die Blockbibliothek Funktionsplan ist eine kostengünstige und richtliniennahe Alternative zu Stateflow und ermöglicht eine vom Normalablauf separate Fehlerbehandlung - ohne Verlust an Übersichtlichkeit.

Wesentliche Eigenschaften der SIMULINK Blockbibliothek Funktionsplan

Das Softwarewerkzeug FUP Blockbibliothek wurde zur Simulation von ereignisgesteuerten Prozessen, speziell in mechatronischen Systemen entwickelt. Es stellt eine Erweiterung zu MATLAB/SIMULINK dar. Wesentliche Eigenschaften sind:

Blockorientierung

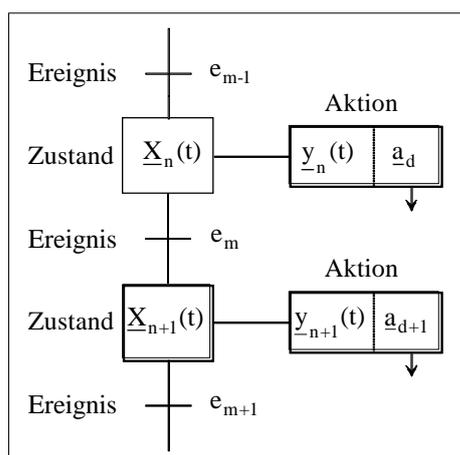
Es handelt sich um ein offenes System mit blockorientiertem Aufbau und der Möglichkeit zur Generierung von C-Code für den Echtzeiteinsatz und den Schutz von Anwenderwissen.

Richtliniennähe

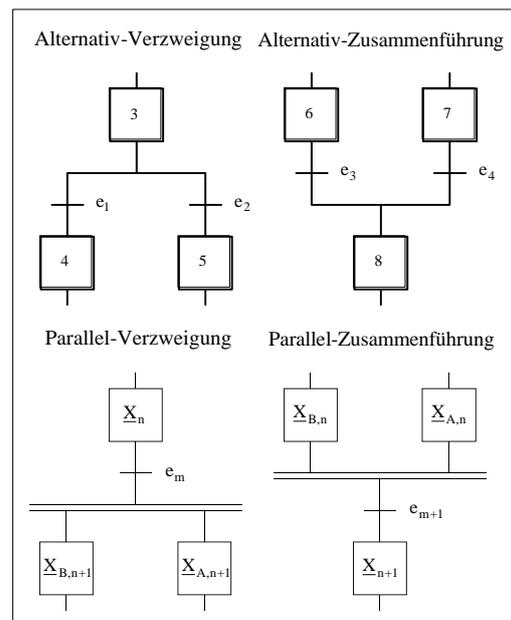
Für die graphische Darstellung (Blockikonen) wird die Nähe zur VDI/VDE-Richtlinie 3684 "Beschreibung ereignisgesteuerter Bewegungsabläufe mit Funktionsplänen" angestrebt. Diese Richtlinie wurde durch den GMA Fachausschuß 4.20 "Bewegungssteuerungen für Be- und Verarbeitungsmaschinen" ausgearbeitet und kann über den Beuth Verlag bezogen werden.

Ausnahmen

Beispiele für wesentliche unvermeidbare Abweichungen der graphischen Darstellung zwischen Blockbibliothek Funktionsplan und Richtlinie 3684 sind die Anschaltung des Kommentarblockes KB an den Zustandsausgängen (statt an der rechten Seite eines Zustandsblockes, z.B. ZA), die generelle graphische Kennzeichnung der Wirkungsrichtung durch Pfeile (statt nur bei Wirkungsrichtung nach oben) und die Zuführung von Ereignissen an die Ereignisblöcke mittels "Goto"-Blöcken (Ereignisnamen statt ohne - in eckigen Klammern und in Blockumrandung). Der Kommentarblock kann Parameter unterschiedlicher Definition übergeben, im Gegensatz zur Richtlinie jedoch keine weiteren mathematischen Beziehungen. Zum Vergleich sind nachfolgend die Darstellungen nach VDI/VDE-Richtlinie 3684 angegeben.



Funktionsplan, Symbole nach VDI/VDE-Richtlinie 3684



Verzweigungen und Zusammenführungen nach VDI/VDE-Richtlinie 3684

Verschachtelung

Hinsichtlich der Verschachtelung von Verzweigungen und Zusammenführungen in einem Funktionsplan bestehen keine Restriktionen, d.h. entsprechend der Syntax können Parallel- und Alternativverzweigungen verschachtelt sein.

Teilfunktionspläne

Bezüglich der Bildung von Teilfunktionsplänen (Untersystemen) werden keine Restriktionen vorgegeben. Damit können große Funktionspläne beliebig aufgeteilt werden. Zu beachten ist, daß die Blöcke der Untersysteme dann keine spezielle Blockgraphik aufweisen. Zu empfehlen ist die Wahl eines Textes (disp-Befehl in Maskierung) zur gut wiedererkennbaren Kennzeichnung.

Zustandsnummern

Die Zustandsnummern sind, unter Ausschluß von Null, als natürliche Zahl (i.R. 4-stellig) frei wählbar. Es besteht kein Zusammenhang zwischen der SIMULINK-internen, automatischen Blocknummerierung und den frei wählbaren Zustandsnummern.

Funktionsplannummern

Durch Vorgabe von Funktionsplannummer und Zustandsnummer besteht die Möglichkeit der gleichzeitigen Simulation mehrerer Funktionspläne. In unterschiedlichen Funktionsplänen können dabei wahlweise gleiche Zustandsnummern vergeben werden. Als Funktionsplannummer ist eine natürliche Zahl zwischen 1 und 9 zulässig.

Steuervariablen

Die durch den Kommentarblock KB - als Schnittstelle zu den Bewegungszellen (Antrieben) - generierten Parameter können sowohl logische, als auch numerische Steuervariablen darstellen. Bei Bedarf sind diese Steuervariablen im Gegensatz zu Zustandsnummern zwischen unterschiedlichen Funktionsplänen und Prozeßmodellen gültig, wenn gleiche Namen vereinbart werden.

Laufzeiten

Die durchgängige Programmierung der Funktionalität mit S-Funktionen und MEX-Files auf Basis von C-Code und ein sparsamer Einsatz von Standardblöcken tragen zu minimierten Laufzeiten bei.

Fehlerbehandlung

Die Fehlerbehandlung ist in vom Normalablauf separaten Teilfunktionsplänen möglich. Von jedem Zustand aus kann ein Übergang in einen von mehreren Pfaden zur Fehlerbehandlung erfolgen, ohne daß dabei die Übersichtlichkeit der graphischen Darstellung leidet.

Anwendungshinweise zur Modellierung des Bewegungsablaufes mit der SIMULINK Blockbibliothek Funktionsplan

Voraussetzungen

Zur Untersuchung und Entwicklung ereignisgesteuerter Bewegungsabläufe mit der Blockbibliothek Funktionsplan müssen zwei Voraussetzungen erfüllt sein.

Software:

Konzipiert als Erweiterungs-Blockbibliothek für SIMULINK muß das Softwarepaket MATLAB/SIMULINK ab Version 5.2/2.2 unter W95 oder WNT4.0 vorhanden sein. Zusatz-Blocksets oder Toolboxen sind für die Blockbibliothek Funktionsplan nicht erforderlich.

Vorkenntnisse:

Der Anwender sollte Grundkenntnisse zur Bedienung von SIMULINK besitzen.

Installation

Die Installation der Blockbibliothek Funktionsplan erfolgt in zwei Schritten.

Schritt 1:

Nach dem Befehl "Unlock Library" im Menü "Edit" des Fensters der "Simulink Block Library" kann die Ikone der Erweiterungsbibliothek Funktionsplan aus dem mitgelieferten, geöffneten File "fup_bibliothek_v_xy.mdl" in das Fenster der "Simulink Block Library" kopiert werden. Dabei steht "xy" für die Versionsnummer der Erweiterungsbibliothek. Nach Positionierung der neuen Bibliotheksikone an gewünschter Stelle muß SIMULINK zunächst geschlossen werden, um die gesamte "Simulink Block Library" wieder zu sperren.

Schritt 2:

Im zweiten Schritt muß eine neu eingerichtete Direktorie mit frei wählbarem Namen in den MATLAB Suchpfad aufgenommen und mit den mitgelieferten Files "fp_xyz.dll" und "fpi_xyz.dll" geladen werden. Dabei steht "xyz" für die jeweilige Bezeichnung. Die Anzahl diese Files ist versionsabhängig und im mitgelieferten "readme.txt" angegeben.

Gesamtstruktur

Die Gesamtstruktur zur Modellierung eines Bewegungsablaufes setzt sich prinzipiell aus den Teilen Funktionsplan - meist vorteilhaft in Teilfunktionspläne zerlegt - , Prozeßmodell und Prozeßanpassung zusammen. Im einfachsten Fall besteht das Prozeßmodell nur aus Einzelantrieben (Bewegungszellen) zur Ausführung von Stellbewegungen.

In der Gesamtstruktur können außerdem typisch Schalter zur Steuerung des Bewegungsablaufes (Auswahl von Varianten, Simulation von Fehlern etc.) sowie Blöcke zur Anzeige und Datenspeicherung angeordnet sein.

Funktionsplan/Teilfunktionspläne:

Der gesamte Funktionsumfang einer Bewegungssteuerung kann grob in verschiedene Betriebszustände unterteilt werden. Für jeden dieser Betriebszustände wird ein Teilfunktionsplan entworfen. Jeder Einzelzustand der Teilfunktionspläne kann einmalig pro Aktivierung über einen zugeordneten Kommentarblock (KB) Steuerinformationen an die Einzelantriebe (Bewegungszellen) übermitteln.

Der Funktionsplan erhält einen Ereignisvektor (EV) von der Prozeßanpassung. Ist der Funktionsplan in Teilfunktionspläne aufgeteilt, so ist auch eine Aufteilung des Vektors EV in Teilereignisvektoren, die den Teilfunktionsplänen zugeordnet sind, sinnvoll. Je nach Anwendungsfall kann der Funktionsplan einen Zustandsvektor mit ausgewählten Zustandsinformationen (ZV_FUP) an die Prozeßanpassung zurückliefern.

Prozeßmodell / Einzelantriebe (Bewegungszellen):

Die vom Funktionsplan/Teilfunktionsplan durch Kommentarblöcke (KB) übermittelten Steuerinformationen werden mittels FUP_Variablenblöcken (FUP_V) von den Einzelantrieben (Bewegungszellen) empfangen und interpretiert. Die Bedeutung dieser Steuerinformationen kann dabei in Abhängigkeit von den verwendeten Einzelantriebsmodellen bzw. Modellen der Bewegungszellen unterschiedlich sein. Grundsätzlich können sowohl Schaltinformationen (z.B. Start eines Rampengenerators mit vordefinierten Parametern), als auch Parameter selbst (z.B. Anstieg und Endwert einer Rampe) vorgegeben werden.

Für die Prozeßanpassung müssen die Einzelantriebe (Bewegungszellen) einen Zustandsvektor des Bewegungsablaufes (ZV_Pro) zur Verfügung stellen. Typische Elemente dieses Zustandsvektors können z.B. Soll- und Istwerte für Lage, Drehzahl und Beschleunigung sein.

Prozeßanpassung:

Die Prozeßanpassung bildet aus Vektor ZV_Pro und Vektor ZV_FUP den Ereignisvektor (EV) bzw. die Teilereignisvektoren (EV_x). In der Regel werden die Elemente von ZV_Pro in logischen Gleichungen verknüpft (Nutzung der erweiterten Ereignisgeneratoren, z.B. EGE_w) und die Elemente von ZV_FUP zum Start von Zeitgebern (ZG) verwendet.

Ereignistabelle:

Die zum Weiterschalten im Funktionsplan notwendigen Ereignisse werden entsprechend VDI/VDE Richtlinie 3684 in einer Ereignistabelle übersichtlich zusammengetragen.

Fehlerbehandlung:

Hinsichtlich der Beschreibung und Simulation von Fehlern existieren grundsätzlich zwei Möglichkeiten.

Spezieller Fehler:

Kann ein spezieller Fehler nur an einer bestimmten Stelle im Funktionsplan auftreten und ist bei entsprechender Reaktion ein Weiterarbeiten möglich, so ist es sinnvoll, diesen Teilablauf in den vorhandenen Funktionsplan aufzunehmen.

Allgemeiner Fehler:

Für Fehler, die an beliebiger Stelle auftreten können und zum Abschalten führen, steht der Block Allgemeiner Fehlerzustand (ZF) zur Verfügung. Nach Aktivierung dieses Blockes und ereignisabhängiger Verzweigung in unterschiedliche Fehlerfunktionsabläufe ist der Anfangszustand oder ein beliebiger anderer sinnvoller Zustand erneut aktivierbar. Verzweigungen in unterschiedliche Fehlerfunktionsabläufe in Abhängigkeit der vor dem Auftreten des Fehlers aktiven Blöcke können mit Hilfe des Blockes Fehlervorzustand (FVZ) über die Berechnung Boolescher Gleichungen erfolgen.

Blockparametrierung

Jeder Block der Blockbibliothek Funktionsplan ist durch ein Blockkürzel gekennzeichnet. Durch die Nähe der graphischen Darstellung zur VDI/VDE Richtlinie 3684 und die Gestaltung der Ikonen der über die Richtlinie hinausgehenden Blöcke, können diese Blockkürzel bereits nach kurzer Einarbeitungszeit unsichtbar geschaltet werden (nach Blockmarkierung Befehl

"Hide Name" im Menü Format). Damit wird der Informationsgehalt der graphischen Darstellung auf den eigentlichen Funktionsinhalt des Bewegungsablaufes beschränkt.

Auswahl:

Wie in SIMULINK üblich können die Blöcke beliebig oft aus einer Bibliothek in ein neues File kopiert werden. Hierzu wird der gewünschte Block, nach Öffnung der Bibliothek durch Doppelklick auf die Bibliotheksikone, markiert und mit gedrückter linker Maustaste in ein anderes Bildschirmfenster gezogen. Die Verbindung der Blockein- und -ausgänge erfolgt anschließend ebenfalls mittels Maus. Die graphische Anordnung der Blöcke, insbesondere bei Verzweigungen, ist nach individuellen Anforderungen frei gestaltbar.

Parametrierung:

Nach Doppelklick auf einen Block öffnet sich ein für jeden Block spezielles Eingabefenster mit Blockname und Blockbeschreibung. In diesem Fenster erfolgen die blockspezifischen Einstellungen, wie z.B. Vorgabe von Zustandsnummer und Funktionsplannummer für Blöcke ZX (Allgemeiner Zustand), einer Stringvariablen und eines Anfangswertes für Blöcke FUP_V (FUP_Variablenblock), oder einer Zeit für Blöcke ZG (Zeitgeber). Die ausführliche On-line Blockhilfe ist ebenfalls über dieses Fenster erreichbar. Dieser Hilfetext liefert alle notwendigen Informationen zur Anwendung eines Blockes, wie z.B. Einsatzbedingungen, Angaben zu den Ein- und Ausgängen oder Randbedingungen für Eingabeparameter.

Standardblöcke

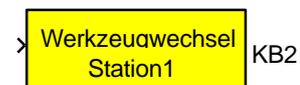
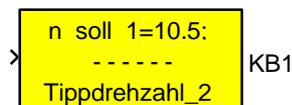
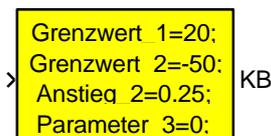
Auf die Gestaltung der Antriebe (Bewegungszellen) in SIMULINK wird an dieser Stelle nicht näher eingegangen. Hierfür stehen u.a. die Blockbibliotheken Discrete, Linear, Nonlinear und Power System Blockset zur Verfügung. Im Zusammenhang mit der Blockbibliothek Funktionsplan ist der Einsatz bestimmter Standardblöcke typisch. Das geht auch aus den 11 Beispielen hervor. Diese Standardblöcke und ihre Aufgaben sind:

- Mux:* Zusammenfassen von Skalaren und/oder Vektoren zu Vektoren
- Demux:* Vereinzeln von Vektoren zu Skalaren und/oder Vektoren
- Display:* Zahlenwertanzeige für Skalare und Vektoren
- Scope:* Aufnahme und Anzeige (Lupenfunktion) von Kurvenverläufen, sowie Bereitstellung linearer Zeitablaufdiagramme für Dokumentationszwecke
- Constant:* Vorgabe von Skalaren und Vektoren für Testzwecke und Schaltaufgaben
- Digital Clock:* Diskrete Zeitausgabe - Zählimpulse für Zeitgeber (ZG)
- Konnektoren:* Unsichtbare Verbindung von Blockein- und -ausgängen
- Manual Switch:* On-line Umschaltung per Mausklick für Testzwecke und Schalthandlungen

Kommentarblock

Der Kommentarblock (KB) kann als einziger Block der Blockbibliothek Funktionsplan in verschiedenen Erscheinungsformen auftreten.

1. Nur Wertevorgaben
2. Wertevorgaben und Kommentar
3. Nur Kommentar

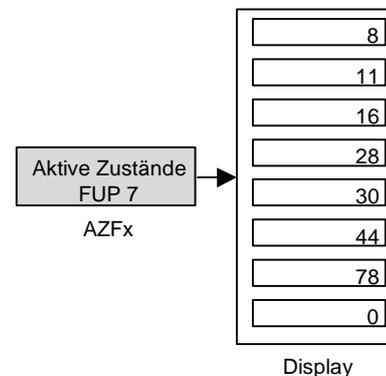


Zustandsanzeige

Zur Anzeige aktiver Zustände eines Funktionsplanes stehen zwei Möglichkeiten zur Verfügung.

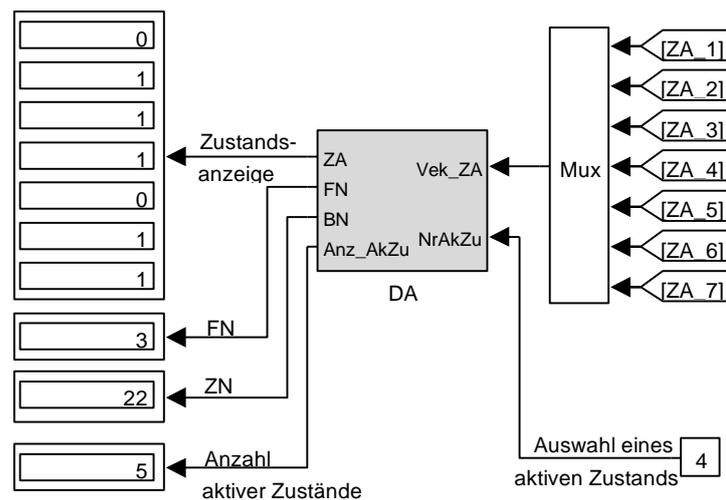
Anzeige aller aktiven Zustände eines Funktionsplanes:

Hierbei entsteht der geringste Aufwand. Pro Funktionsplan wird ein mit der jeweiligen Funktionsplannummer parametrierter Block Aktive Zustände (AZFx) an einen Display Block angeschlossen. Dabei ist zu beachten, daß die Anzahl der angezeigten Zustände wegen einer durch SIMULINK bedingten Einschränkung zur Zeit fest eingestellt werden muß. Die eingestellte Anzahl von 7 Zustandsnummern kann nach Rücksprache verändert werden.



Anzeige ausgewählter Zustände, auch unterschiedlicher Funktionspläne:

Hierfür müssen die interessierenden und über einen Block Mux zu einem Vektor zusammengefaßten Zustandsausgänge an den Eingang Vek_ZA des Blockes Dekodierung Anzeige (DA) geführt werden. Dieser Block liefert am Ausgang ZA einen Vektor mit den entsprechenden binären Zustandsinformationen und am Ausgang Anz_AkZu die Summe der aktiven interessierenden Zustände. Durch den Eingang NrAkZu



kann einer der aktiven Zustände über seine Elementennummer im Vektor der interessierenden Zustände (beginnend mit eins) ausgewählt und die zugehörige Funktionsplan- und Zustandsnummer über die Ausgänge FN und ZN zur Anzeige gebracht werden.

Test

Nach Entwurf und Programmierung eines Funktionsplanes muß zuerst ein statischer Test ohne Prozeßmodelle und Prozeßanpassung erfolgen. Andererseits sollten die für den anschließenden dynamischen Test des Funktionsplanes verwendeten Prozeßmodelle bereits ausreichend geprüft sein.

Statischer Test:

Zur Durchführung statischer Prüfungen von Funktionsplänen ist die Simulation von Ereignissen zum Weiterschalten im Funktionsplan durch Blöcke Constant deren Elemente direkt während des Simulationslaufes veränderbar sind oder über Blöcke Manual Switch mit skalaren Blöcken Constant von fester Einstellung logisch "0" und "1" an jeweils einem Eingang möglich.



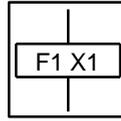
Im ersten Fall kann eine Gruppierung der Ereignisse in Vektoren die Übersichtlichkeit erhöhen. Die Zuführung des Ereignissignales zum Block Ereignis (EX) erfolgt dann über Vereinzeln (Block Demux) und Konnektoren, d.h. es sind nicht unbedingt so viele Blöcke Constant notwendig bzw. sinnvoll wie Ereignisse zu simulieren sind.

Im zweiten Fall entspricht die erforderliche Anzahl der Blöcke Manual Switch der Ereignisanzahl. Jeweils ebenso viele skalare Blöcke Constant mit logisch "0" und "1" werden zur Speisung der Manual Switch Eingänge benötigt. Während der Prüfung sollten die Ereignisse unmittelbar nach einem Übergang in einen neuen Systemzustand rückgesetzt werden. Mit dieser Vorgehensweise bleibt der Überblick über das Gesamtsystem erhalten und aus dem Testlauf resultierende Fehler werden vermieden. Für statische Tests ist die Einstellung "Stop time" = inf im Menü Simulation/Parameters/Solver notwendig.

Dynamischer Test:

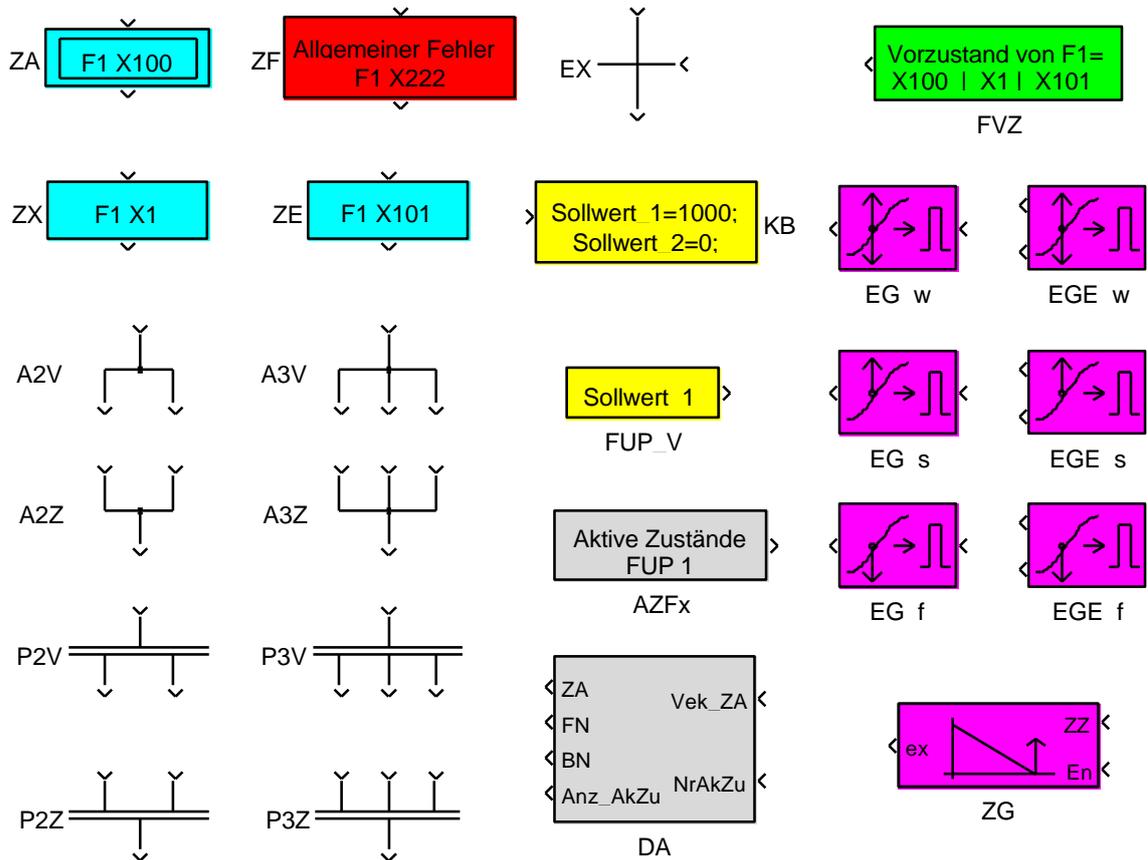
Zur dynamischen Prüfung von Funktionsplänen müssen die Ereignisse aus dem zeitabhängig veränderlich verstandenen Zustandsvektor des Bewegungsablaufes (ZV_Pro) generiert werden. Dabei ist zu beachten, daß die Ereignissignale eine bestimmte Mindestlänge aufweisen und ein rechtzeitiges Rücksetzen erfolgt. Hierfür stehen die Ereignisgenerator-Blöcke (z.B. EG_w) zur Verfügung. Zwischen zwei Blöcken Ereignis (EX) mit dynamischen Ereignissignalen ist ein Block Ereignis (EX) mit statischem Ereignissignal zulässig. Auf diese Weise ist z.B. die Auswahl von Varianten realisierbar. Hierfür, wie zur Bereitstellung von Start-/Stopumschaltern etc., können die unter Punkt statischer Test bereits erwähnten Blöcke Manual Switch mit skalaren Blöcken Constant von fester Einstellung logisch "0" und "1" an jeweils einem Eingang sinnvoll eingesetzt werden.

Die Blockbibliothek

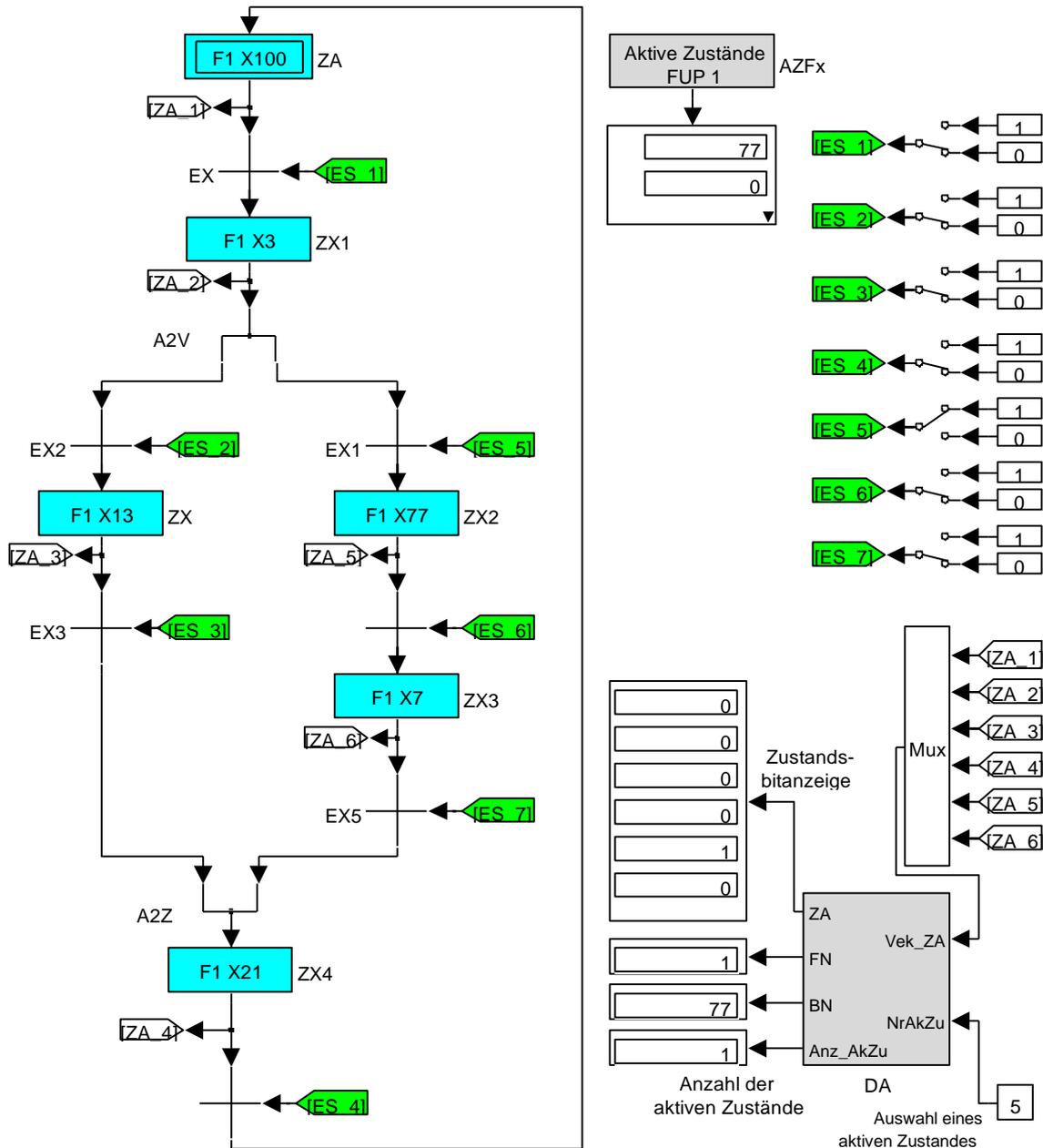


Funktionsplan

FUP Blockbibliothek V.2.3
 alle Rechte vorbehalten
 Dr. Geitner ETI/TU Dresden

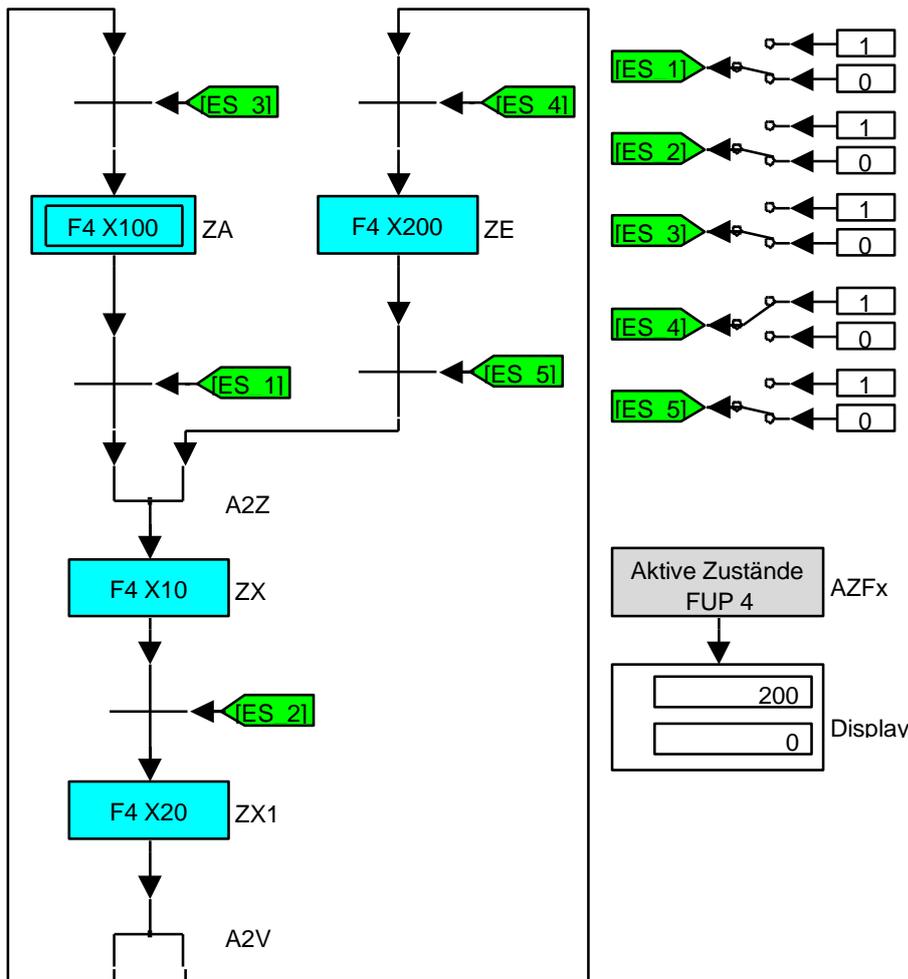


Beispiel 1: Zwei Alternativzweige



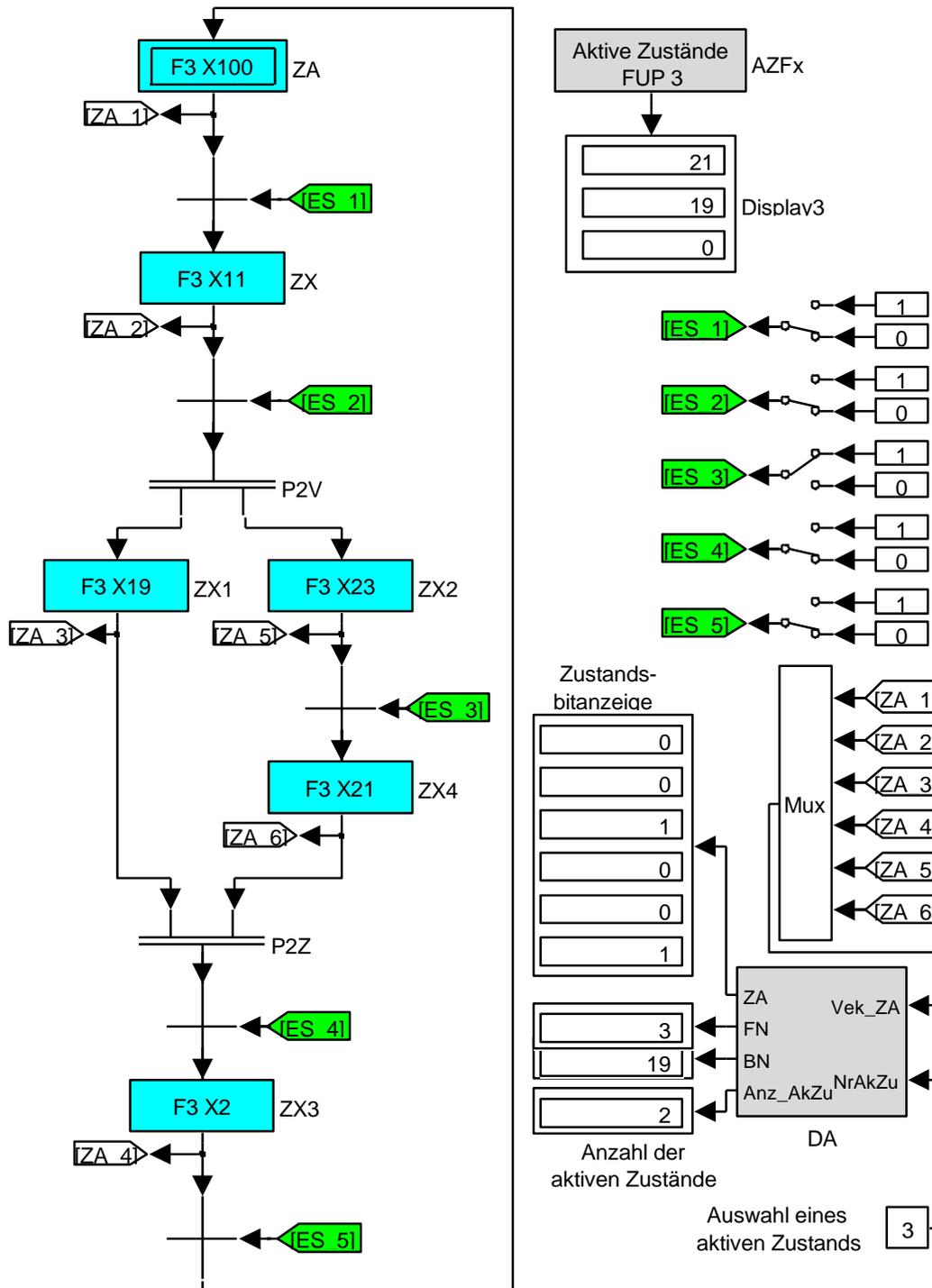
Statischer Test einer Alternativverzweigung mit Anfangszustand (X100) **außerhalb** der Verzweigung. Die Schleife wird über den Anfangszustand geschlossen, die Ereignisse (ES_1 bis ES_7) werden durch Manual Switch Blöcke simuliert und den Zuständen sind keine Aktionen zugeordnet. Zum Vergleich sind beide Anzeigeblöcke (AZFx und DA) eingesetzt.

Beispiel 1a: Zwei Alternativzweige mit Rückführung nicht über Anfangszustand (Einsatz des Blockes ZE)



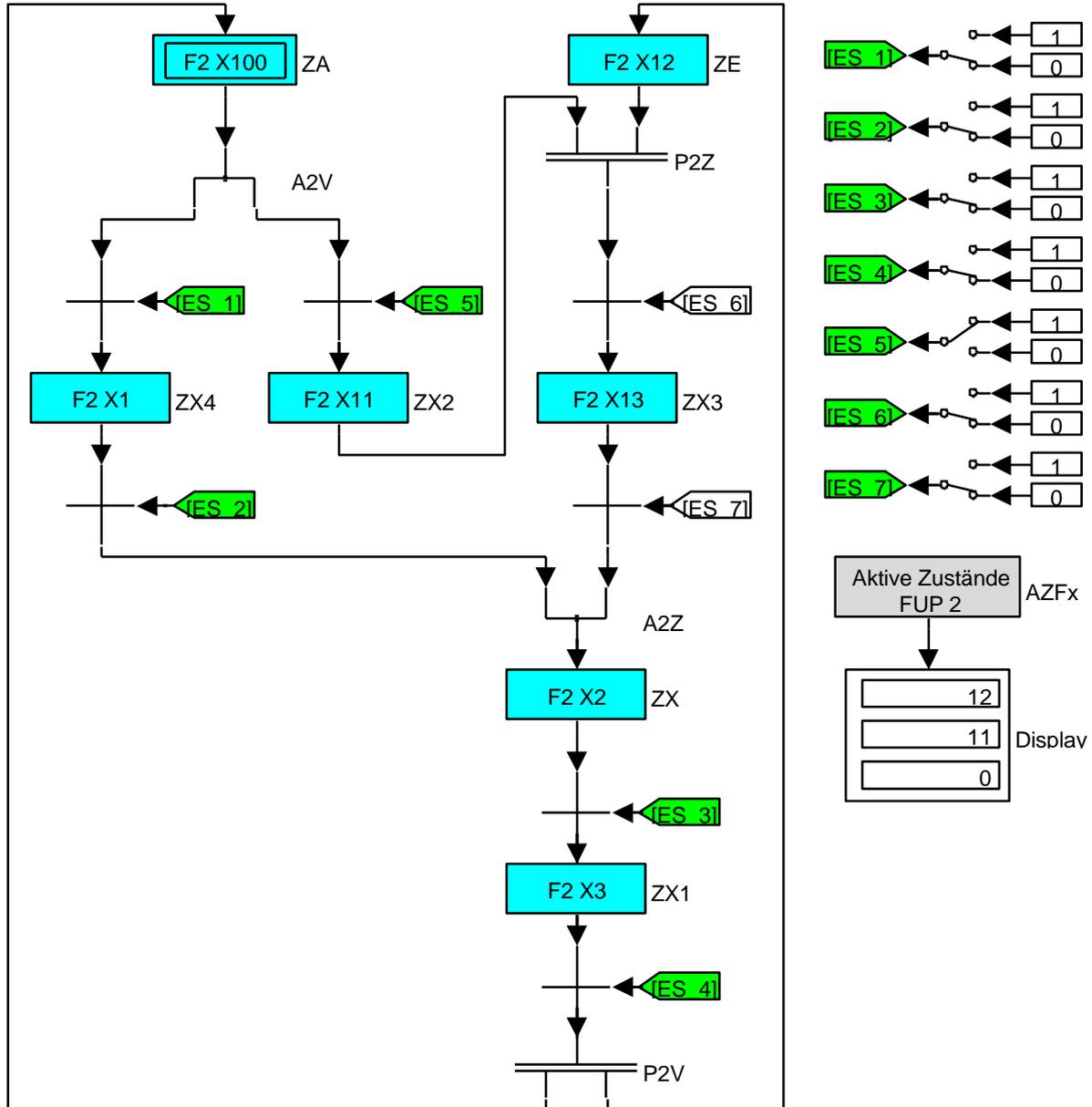
Statischer Test einer Alternativverzweigung mit Anfangszustand (X100) **innerhalb** eines Alternativzweiges. Die nicht über den Anfangszustand geschlossene alternative Schleife enthält **einen** Block ZE (X200). Die Ereignisse (ES_1 bis ES_5) werden durch Manual Switch Blöcke simuliert und den Zuständen sind keine Aktionen zugeordnet.

Beispiel 2: Zwei Parallelzweige



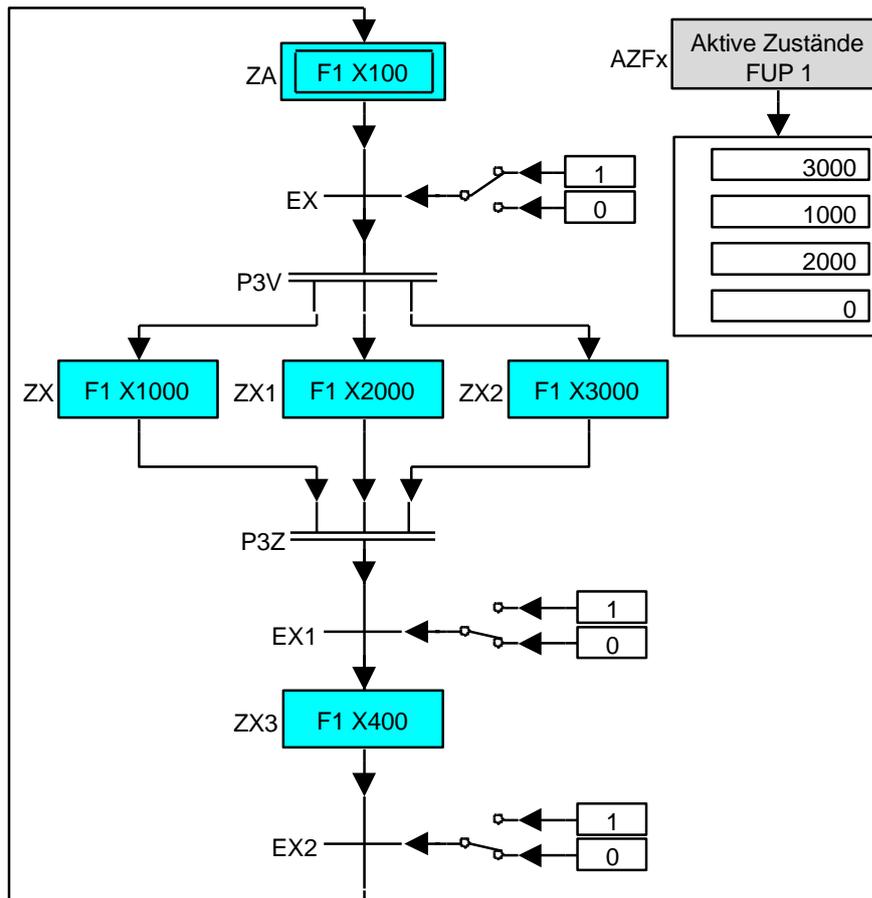
Statischer Test einer Parallelverzweigung mit Anfangszustand (X100) **außerhalb** der Verzweigung. Die Schleife wird über den Anfangszustand geschlossen, die Ereignisse (ES_1 bis ES_5) werden durch Manual Switch Blöcke simuliert und den Zuständen sind keine Aktionen zugeordnet. Zum Vergleich sind beide Anzeigeblöcke (AZFx und DA) eingesetzt.

Beispiel 2a: Zwei Parallelzweige mit Rückführung nicht über Anfangszustand (Einsatz des Blockes ZE)



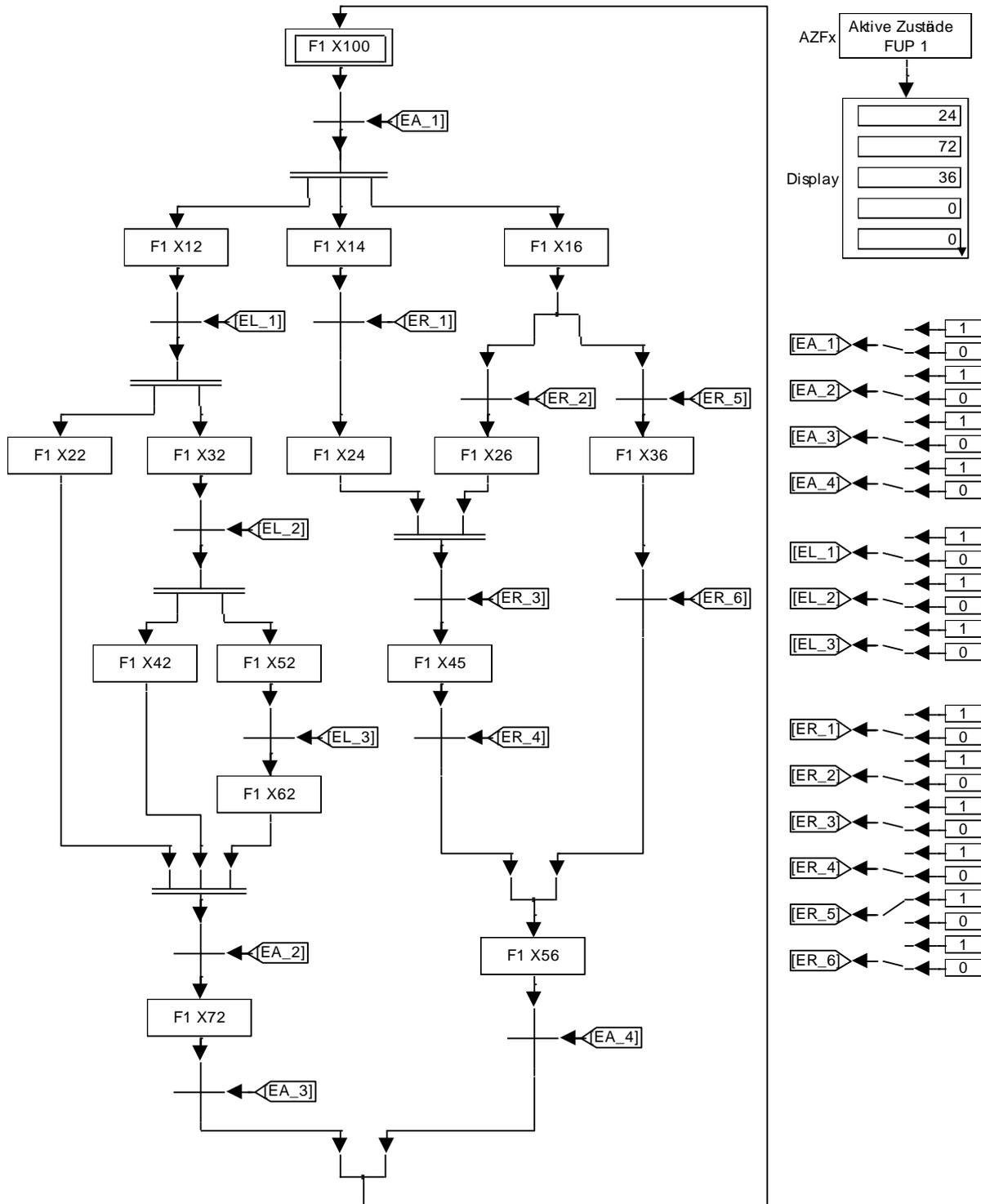
Statischer Test einer Parallelverzweigung mit Anfangszustand (X100) **innerhalb** eines Parallelzweiges. Die nicht über den Anfangszustand geführte Schleife enthält **einen** Block ZE (X12). Die Ereignisse (ES_1 bis ES_5) werden durch Manual Switch Blöcke simuliert und den Zuständen sind keine Aktionen zugeordnet.

Beispiel 2b: Parallelverzweigung ohne Ereignis in der Verzweigung



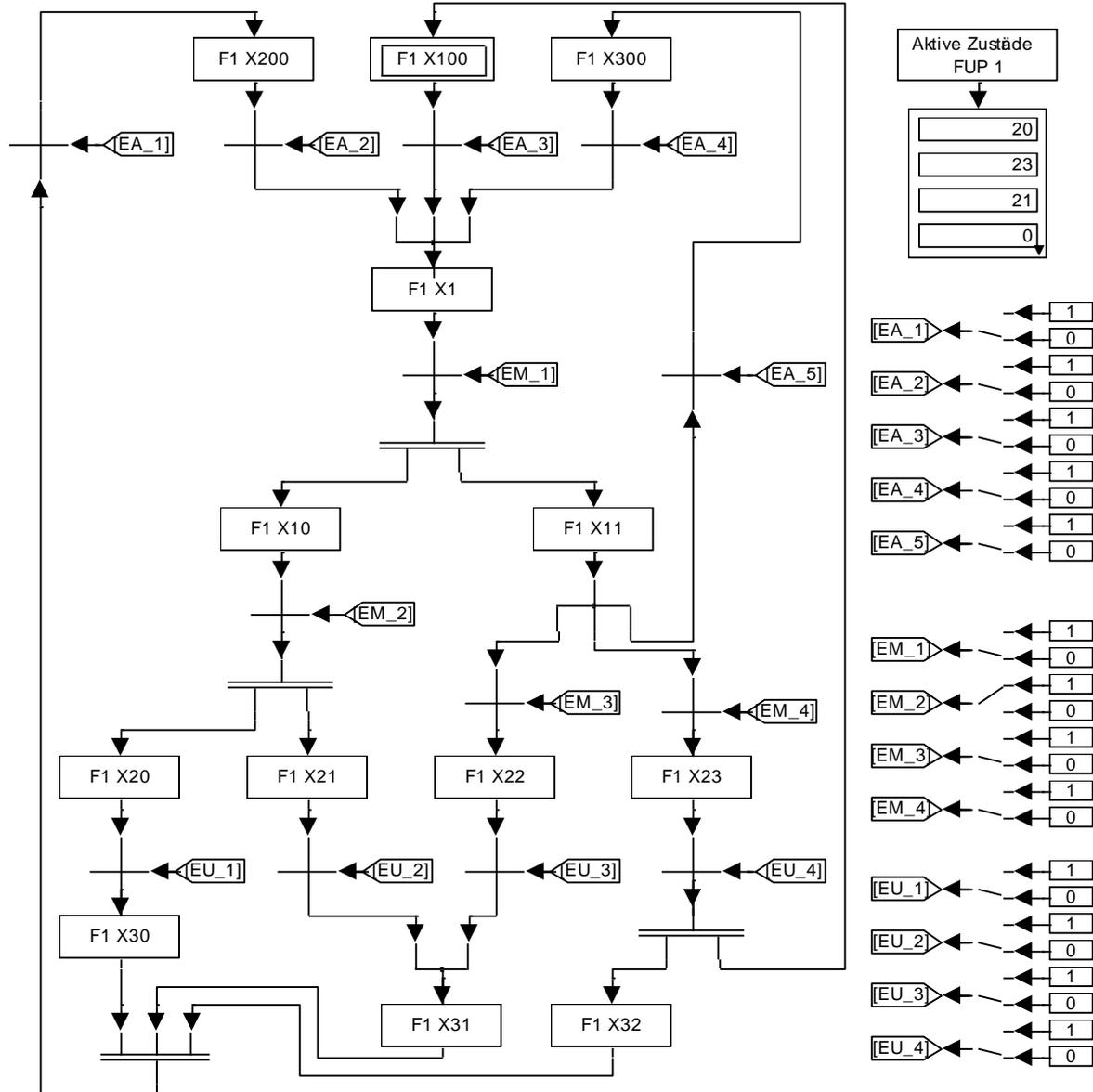
Statischer Test paralleler Zweige die kein Ereignis enthalten. Die Ereignisse werden durch direkt an die Ereignisblöcke (EX bis EX2) angeschlossene Manual Switch Blöcke simuliert. Diese Möglichkeit erlaubt die detaillierte Dokumentation ereignisunabhängiger paralleler Abläufe.

Beispiel 3a: Verschachtelte Verzweigungen I



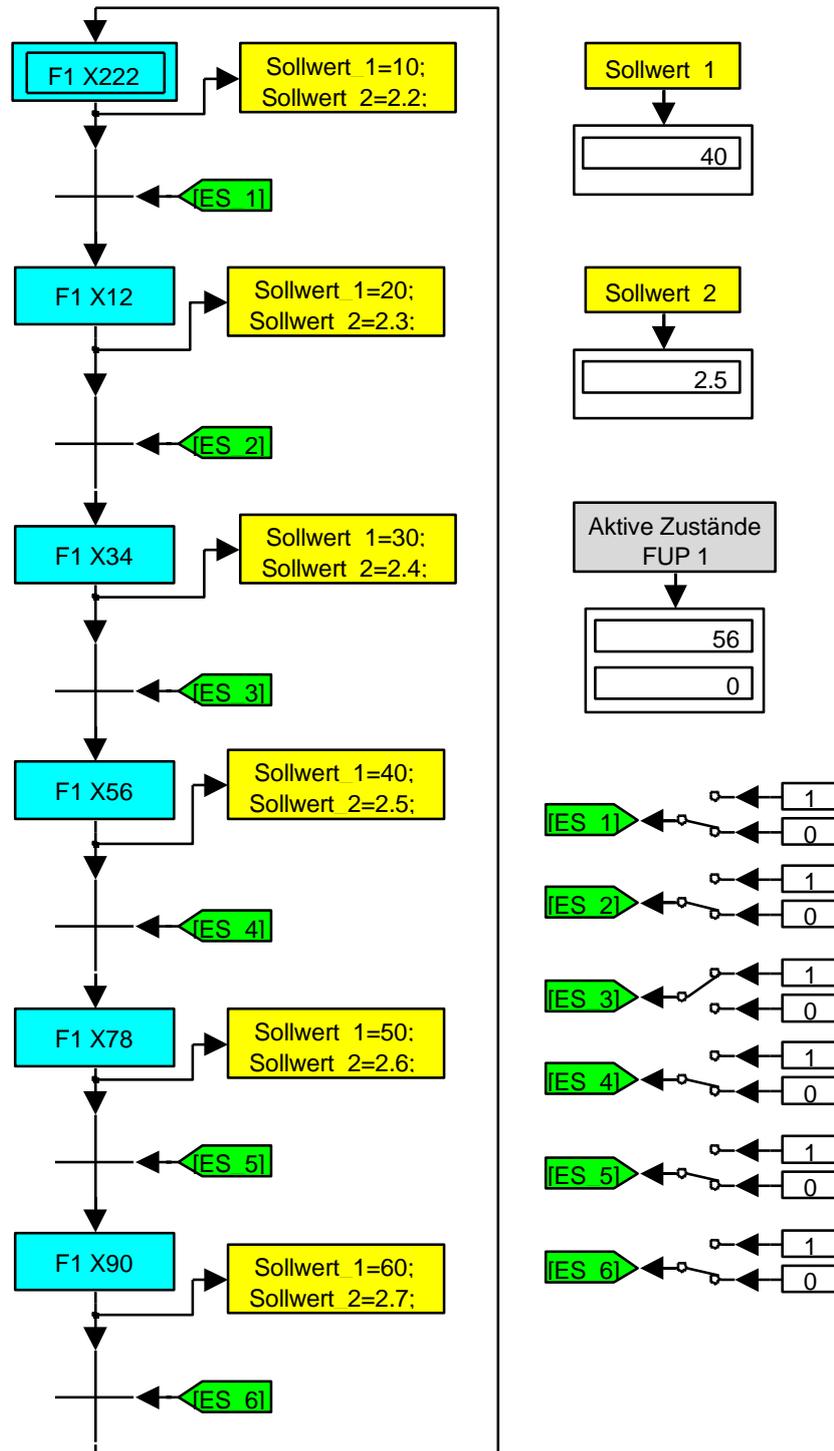
Statischer Test ineinander verschachtelter alternativer und paralleler Verzweigungen. Die einzige Schleife führt über den Anfangszustand (X100). Die Ereignisse (EA_x, EL_y und ER_z) werden durch Manual Switch Blöcke simuliert und den Zuständen sind keine Aktionen zugeordnet. Die Blöcke können, bedingt durch den Funktionsablauf, **mehrmals** aktivgeschaltet werden. **Ein** nachgeschaltetes Ereignis deaktiviert den Block unabhängig davon.

Beispiel 3b: Verschachtelte Verzweigungen II



Statischer Test ineinander verschachtelter alternativer und paralleler Verzweigungen. Die nicht über den Anfangszustand (X100) geschlossenen Schleifen enthalten **je einen** Block ZE (X200, X300). Die Ereignisse (EA_x, EM_y und EU_z) werden durch Manual Switch Blöcke simuliert und den Zuständen sind keine Aktionen zugeordnet. Die Blöcke können, bedingt durch den Funktionsablauf, **mehrmals** aktivgeschaltet werden. **Ein** nachgeschaltetes Ereignis deaktiviert den Block unabhängig davon.

Beispiel 4: Anwendung des Kommentarblockes



Statischer Test mit den Zuständen zugeordneten Aktionen. Die Zustände weisen den Variablen Sollwert_1 und Sollwert_2 Werte zu. Die jeweils gültigen Werte werden mit FUP_V Blöcken empfangen und durch Display Blöcke angezeigt. In den FUP_V Blöcken sind **Anfangswerte** einstellbar. Die Ereignisse (ES_1 bis ES_6) werden durch Manual Switch Blöcke simuliert.

Beispiel 5: Gesamtsystem mit einem Funktionsplan, zwei Antrieben und Zeitgebern

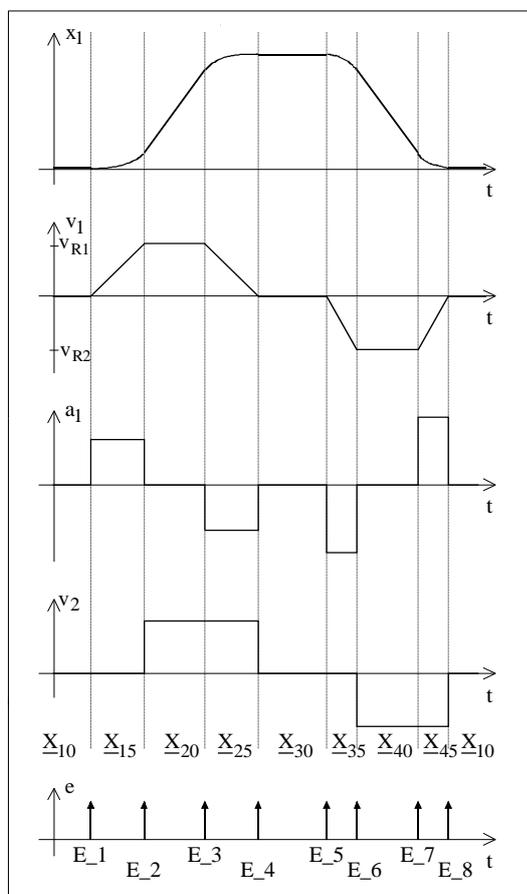
Dynamischer Test mit zwei Stellbewegungen in Anlehnung an Bild 3 und 4 der VDI/VDE-Richtlinie 3684. Der geforderte Bewegungsablauf ist in Beispiel 5 / Bild 1 dargestellt. Antrieb 1 wird durch Beschleunigung (a_1), Geschwindigkeit (v_1) und Lage (x_1), Antrieb 2 nur durch die Geschwindigkeit (v_2) charakterisiert. Abschnitte mit konstanter Drehzahl v_1 sollen durch Zeitvorgaben determiniert werden - Anwendung der Blöcke Zeitgeber (ZG).

Die Gesamtstruktur zur Realisierung dieses Bewegungsablaufes besteht aus den Teilsystemen Antrieb 1, Antrieb 2, Prozeßanpassung und Funktionsplan (Beispiel 5 / Bild 2). Die in Beispiel 5 / Bild 3 angegebenen Antriebsmodelle sind zusammen aus 40 Standardblöcken aufgebaut und bestehen jeweils aus einer zweischleifigen Kaskadenregelstruktur mit Mittelwertbildung für Drehzahl und Strom sowie drehzahlproportionaler Reibung. Die verwendeten digitalen PI-Regler sind in den inneren Schleifen führungs- und in den äußeren Schleifen störoptimiert mit Vorfilter. Zur Optimierung wurde das Digitale Betragsoptimum (BOD) eingesetzt:

- Geitner, G.-H.: Entwurf digitaler Regler für elektrische Antriebe. VDE-Verlag, 1996
- http://eeiwzg.et.tu-dresden.de/ae2_files/ae_1_0.htm

Unter Literaturstelle 2 steht eine MATLAB Freeware Toolbox BOD mit m-Files für eine Regleroptimierung nach BOD zur Verfügung.

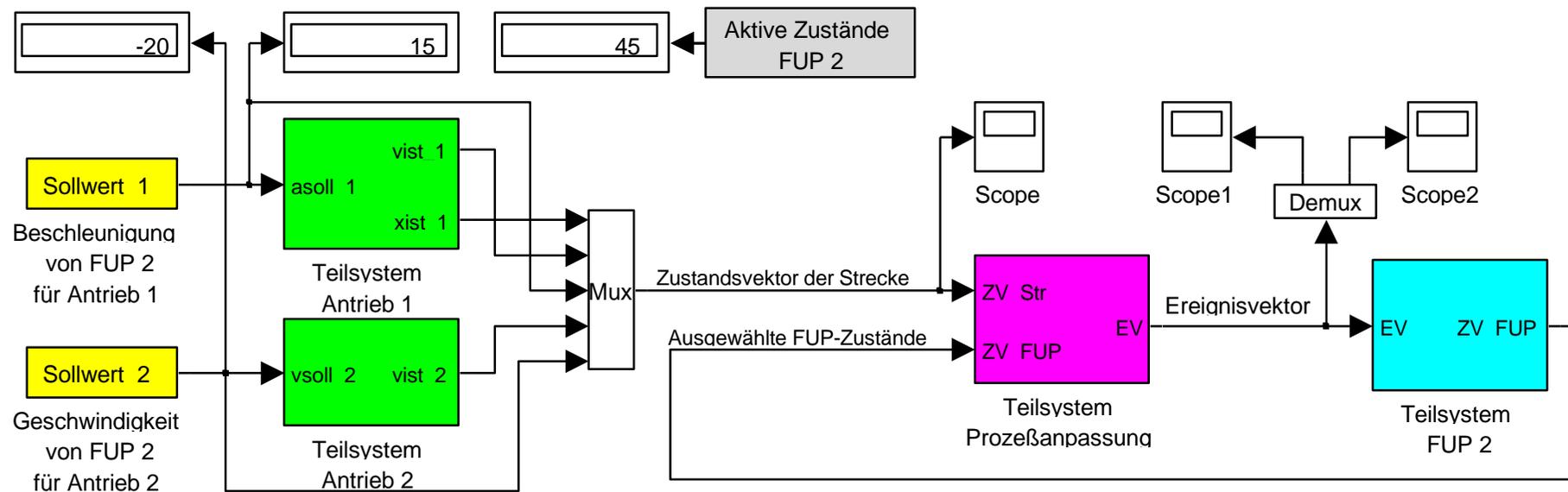
Soll- und Istwerte beider Antriebe bilden den „Zustandsvektor der Strecke“ (ZV_Str) dieses einfachen Beispiels. Das Teilsystem Prozeßanpassung (Beispiel 5 / Bild 4) erhält neben ZV_Str

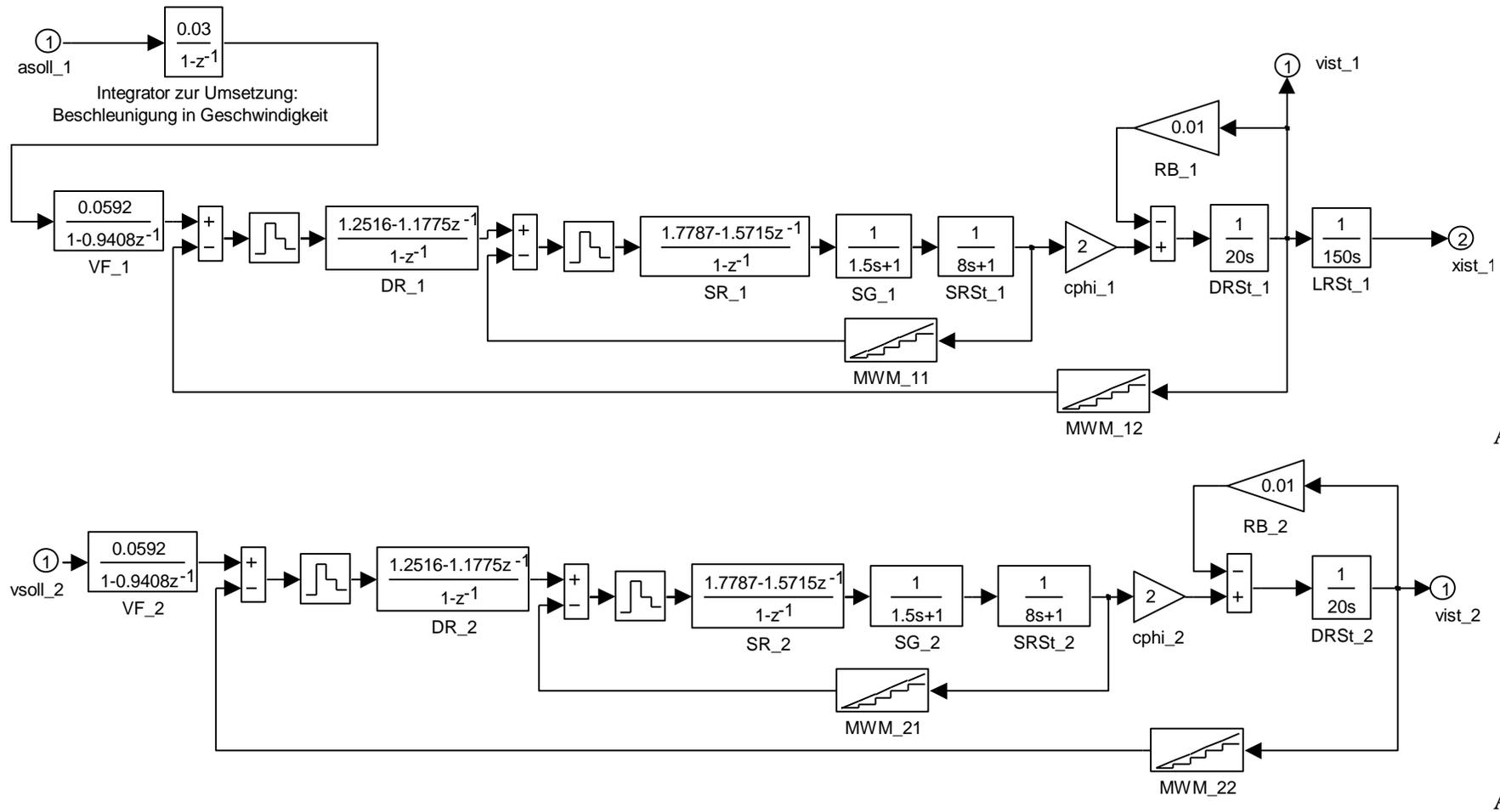


zusätzlich noch den Vektor „Ausgewählte FUP-Zustände“ (ZV_FUP) und generiert den für das Teilsystem Funktionsplan (Beispiel 5 / Bild 5) notwendigen Ereignisvektor EV aus ZV_Str und ZV_FUP. Der Ereignisvektor EV enthält als Elemente die zum Weiterschalten im Funktionsplan nötigen Ereignisse. Die verbale Beschreibung der Ereignisse ist in Beispiel 5 / Bild 5 in der Ereignistabelle zusammengefaßt. Im Teilsystem Funktionsplan können den Zuständen Kommentarblöcke KB zugeordnet werden. In den Kommentarblöcken definierte Wertezuweisungen werden mit Aktivierung eines Zustandes gültig und bewirken über zugehörige „Empfängerblöcke“ vom Typ FUP-Variablenblock (FUP_V) Umschaltungen in den Prozeßmodellen / Bewegungszellen / Antrieben. Da für das Beispiel die Veranschaulichung der Beschreibung und Simulation des Bewegungsablaufes im Vordergrund steht, werden die Antriebe über Beschleunigungs- bzw. Geschwindigkeitssollwert angesteuert und vereinfacht über Vorabschaltwerte auf Solldrehzahl gefahren. Die zugehörigen Simulationsergebnisse sind in Form von Zeitablaufdiagrammen für die Zustandsgrößen und die Ereignisse in Beispiel 5 / Bild 6 angegeben.

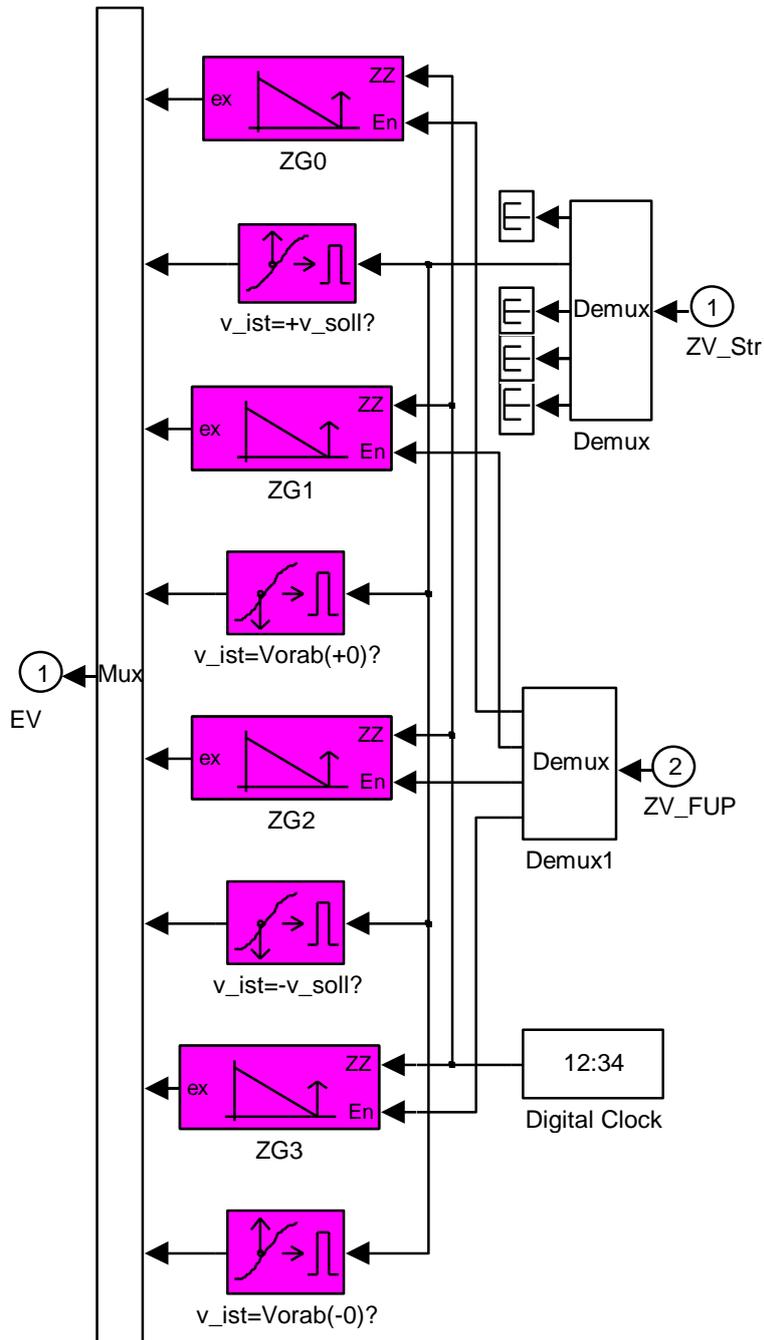
Beispiel 5: Bild 1 Sollbewegungsablauf

Beispiel 5: Gesamtsystem mit einem Funktionsplan, zwei Antrieben und Zeitgebern - **Bild 2 Gesamtstruktur**

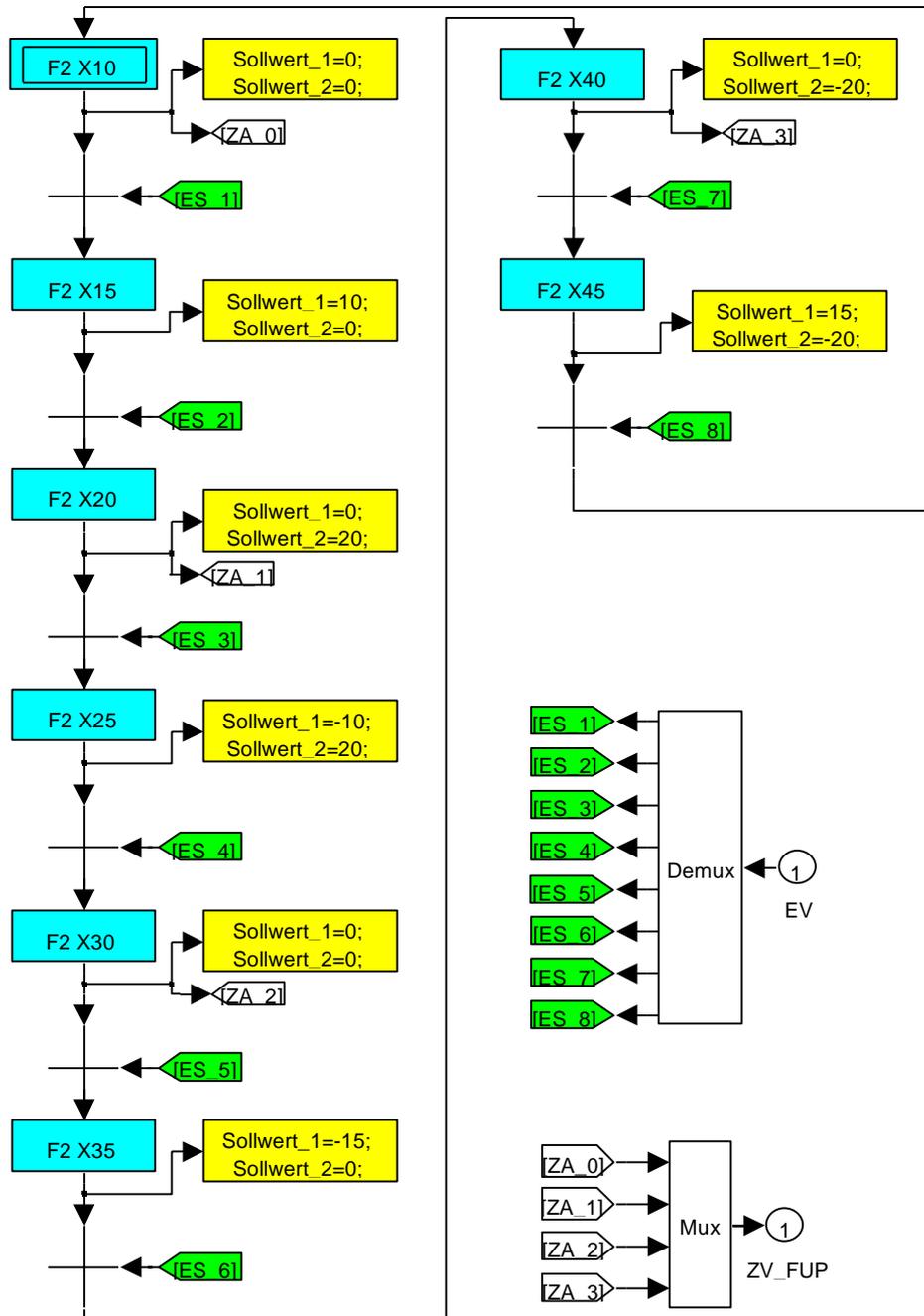


Beispiel 5: Gesamtsystem mit einem Funktionsplan, zwei Antrieben und Zeitgebern - Bild 3 Antriebe


Beispiel 5: Gesamtsystem mit einem Funktionsplan, zwei Antrieben und Zeitgebern
- Bild 4 Prozeßanpassung



Beispiel 5: Gesamtsystem mit einem Funktionsplan, zwei Antrieben und Zeitgebern
- Bild 5 Funktionsplan

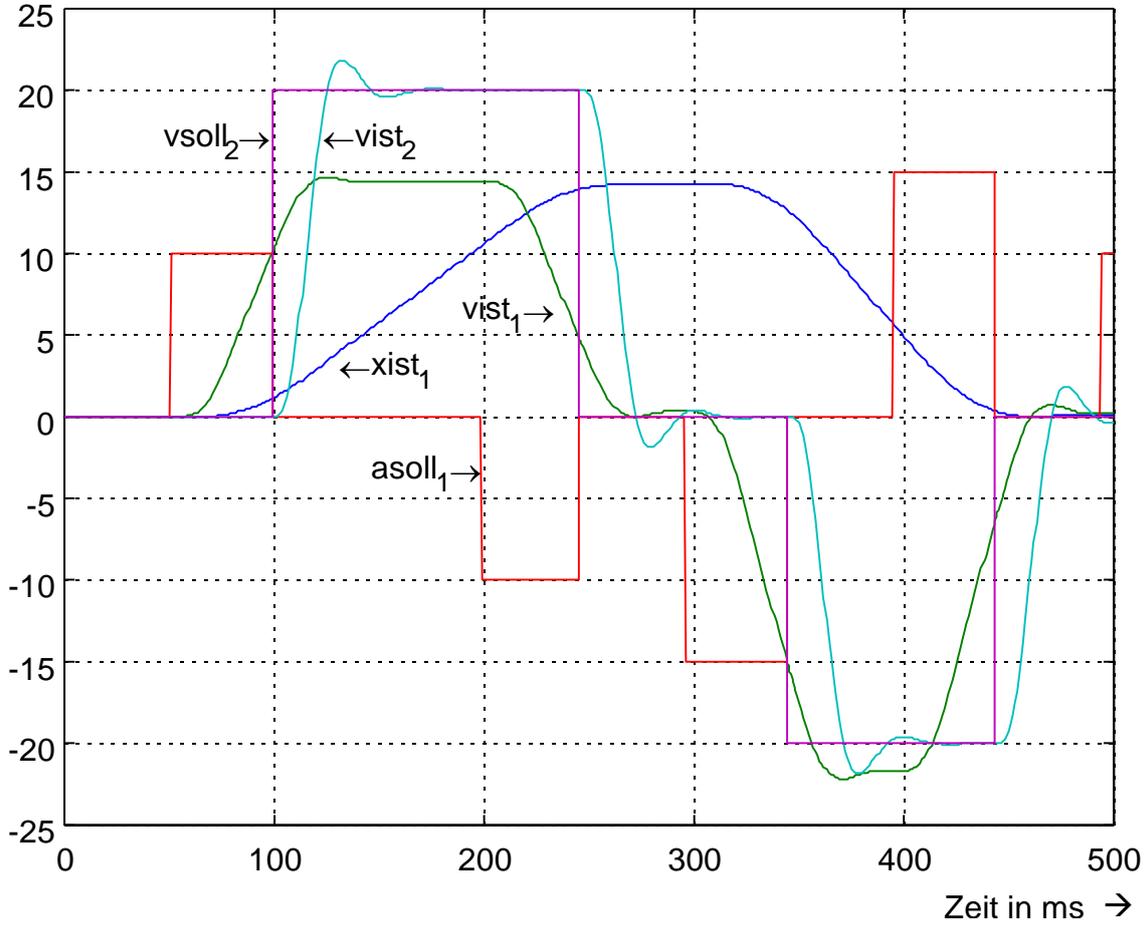


ES_1	Zeitgenerator ZG0 abgelaufen	ES_5	Zeitgenerator ZG2 abgelaufen
ES_2	v _{R1_Ab1} überschritten	ES_6	v _{R2_Ab1} unterschritten
ES_3	Zeitgenerator ZG1 abgelaufen	ES_7	Zeitgenerator ZG3 abgelaufen
ES_4	v _{R1_Ab2} unterschritten	ES_8	v _{R2_Ab2} überschritten

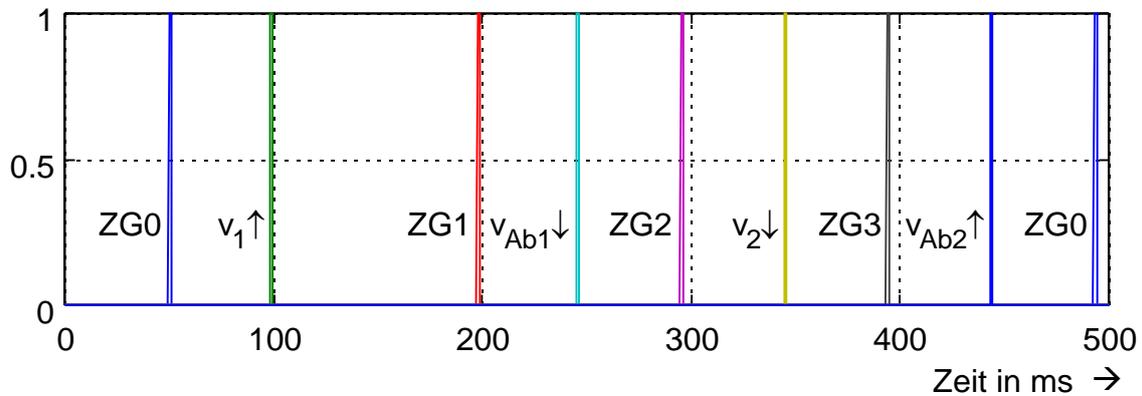
Ereignistabelle: Für das Weiterschalten des Funktionsplanes notwendige Ereignisse

Beispiel 5: Gesamtsystem mit einem Funktionsplan, zwei Antrieben und Zeitgebern
- Bild 6 Simulationsergebnisse

↑ Zustandsgrößen



↑ Ereignisse



Zeitablaufdiagramm eines modifizierten Beispiels nach VDI/VDE 3684

Beispiel 6: Gesamtsystem mit Teilfunktionsplänen, drei Antrieben, logischen Gleichungen für die Ereignisse und Sammelfehlerbehandlung

Dynamischer Test des prinzipiellen Bewegungsablaufes einer vertikalen Schlauchbeutelform-, -füll- und -verschleißmaschine. Der Bewegungsablauf der Schlauchbeutelmaschine kann grob durch die vier Betriebszustände Referenzpunktfahrt, Anlauf, Normalbetrieb und Stillsetzen beschrieben werden. Steht weiterhin das Interesse am Herangehen an die Beschreibung und Simulation des Bewegungsablaufes im Vordergrund, so kann zunächst auch auf eine Nachbildung der Mechanik und damit der Kopplung der Antriebe über ein Material verzichtet werden. Mit dieser Vereinfachung müssen folglich die geregelten Antriebe und die zur Ereignisgenerierung notwendigen Zustandsgrößen der Antriebe zur Verfügung stehen - Beispiel 6 / Bild 1.

Die Antriebe 1 und 2 entsprechen dem Antrieb der Bobine bzw. dem Antrieb des Abzugs und der Längsnahtfügeeinrichtung und liefern jeweils einen Geschwindigkeitswert (v_{ist_1} und v_{ist_2}). Der Antrieb 3 bildet den Siegelbackenantrieb nach und stellt einen Geschwindigkeits- und Lagewert (v_{ist_3} und x_{ist_3}) sowie einen Nullimpuls (NI3) zur Verfügung. Für Antrieb 3 kann ein zufälliger Lageanfangswert (AW zwischen 0 und 360) vorgegeben werden. Als vereinfachte Schnittstelle zu den Einzelantrieben wurde ein Geschwindigkeitssollwert vereinbart (v_{1} , v_{2} , v_{3}). Im Hinblick auf das methodische Herangehen ist das kein schwerwiegender Nachteil. Auf einen Antrieb 4 zur Dosierung des Füllgutes wurde hinsichtlich der Untersuchung des prinzipiellen Bewegungsablaufes noch ganz verzichtet.

Zur Simulation eines Fehlers, der zum Abschalten führt, dient der Zustand X222. Der Fehler kann mit den On-line Schaltern „FEHLER“ generiert und „QUIT“ quittiert werden. Die vereinfachte Fehlerbehandlung besteht im Rücksetzen der Geschwindigkeitssollwerte. Eine detailliertere Fehlerbehandlung ist bei Anwendung von Blöcken Fehlervorzustand (FVZ) möglich - vgl. Beispiel 7. Zum Weiterschalten in den Teilfunktionsplänen benötigte Ereignisse sind unter Beispiel 6 / Bild 2 angegeben.

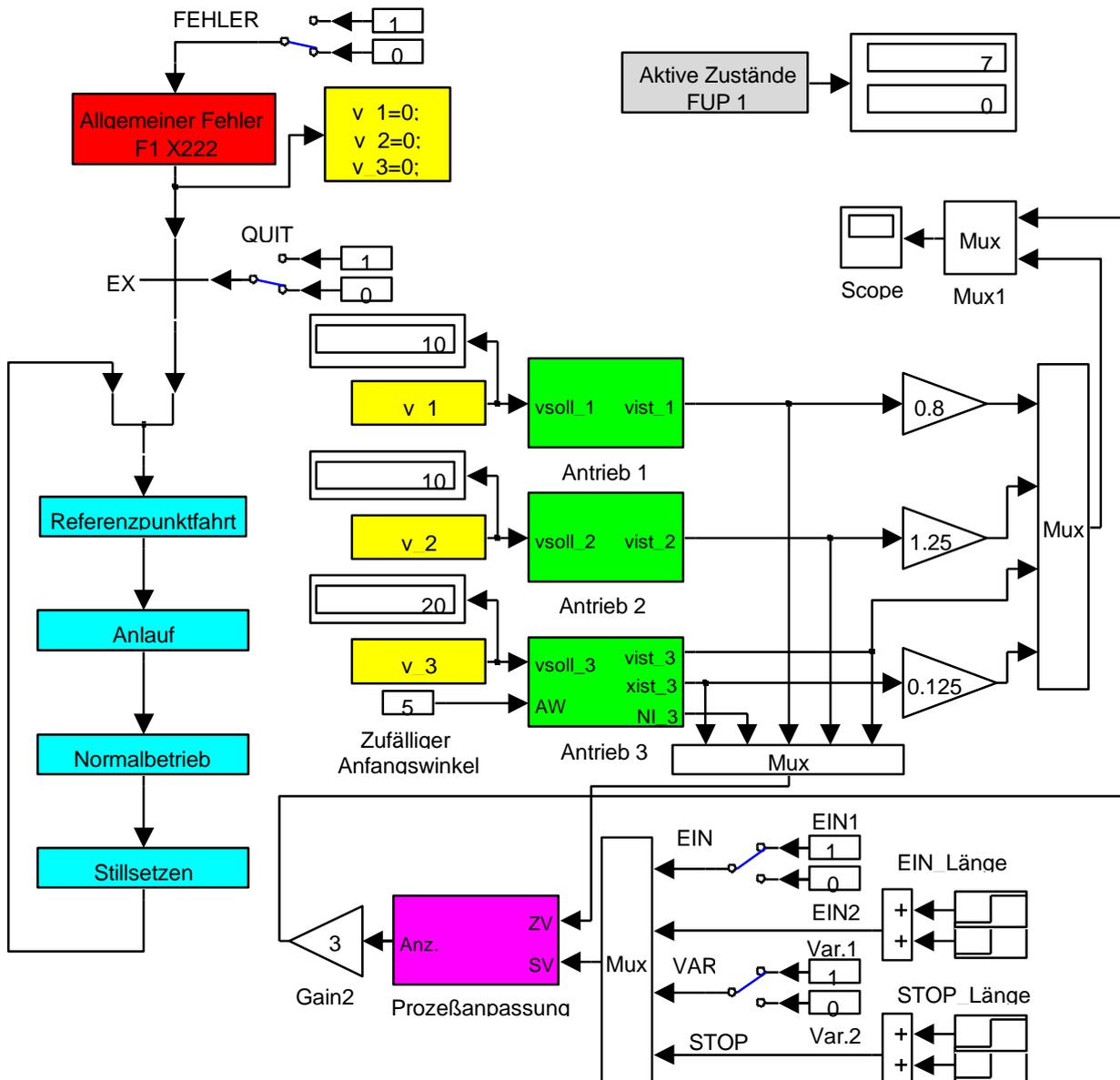
Beispiel 6 / Bild 3 zeigt den Teilfunktionsplan „Referenzpunktfahrt“. Vorbereitet sind zwei Varianten. Über X1 wird nur Antrieb 3 eingeschaltet und der Referenzpunkt rückwärts angefahren. Nachteilig ist die mit Material auftretende Schlingenbildung. Alternativ kann der Referenzpunkt über X2 erreicht werden. In diesem Fall werden alle Antriebe mit „Suchgeschwindigkeit“ vorwärts eingeschaltet. Der Nachteil dieser Variante ist Materialverlust. Über Schalter „EIN“ (Beispiel 6 / Bild 1) erfolgt die Auswahl einer Variante.

Der Teilfunktionsplan „Anlauf“ ist in Beispiel 6 / Bild 4 angegeben. Ausgehend von X12 "Anlauf-bereit" sind zwei Varianten vorbereitet. Über X15 erfolgt der Anlauf aller 3 Antriebe gleichzeitig. Die Antriebe 1 und 2 fahren auf Arbeitsgeschwindigkeit hoch, Antrieb 3 fährt mit höherer Geschwindigkeit zu Winkel 1 (Siegelbeginn). Vorteilhaft ist das schnelle Erreichen des ersten Siegelbeginns. Das setzt eine große zulässige Beschleunigung der Antriebe 1 und 2 voraus. Liegt dieser Fall nicht vor, so ist durch den Auswahlswitcher "VAR" (Beispiel 6 / Bild 1) ein serieller Anlauf über X3 und X4 vorgebar.

Der Teilfunktionsplan "Normalbetrieb" (Beispiel 6 / Bild 5) wird mit X6 erreicht und ist durch ständiges Durchlaufen von X7, X8 und X70 gekennzeichnet. Mit Vorgabe des "STOP"-Signals (Beispiel 6 / Bild 1) erfolgt ein Übergang in X9, der weitere Bewegungsablauf wird durch den Teilfunktionsplan "Stillsetzen" (Beispiel 6 / Bild 6) dokumentiert. Parallel werden über X10 und X11 die Antriebe 1 und 2 stillgesetzt sowie Antrieb 3 über X20 und X21 zu einem Referenzpunkt gefahren und stillgesetzt. Über X15 (alle Antriebe stillgesetzt) wird X100 (einschaltbereit) aktiviert.

Die Funktionspläne nach Beispiel 6 / Bild 3 bis 6 sind im Sinne eines Beispiels zur Demonstration der Herangehensweise und der Möglichkeiten bei Anwendung des Funktionsplanes zur Abbildung von Bewegungsabläufen aufgestellt. Sie sind damit technologisch keine mustergültige Lösung.

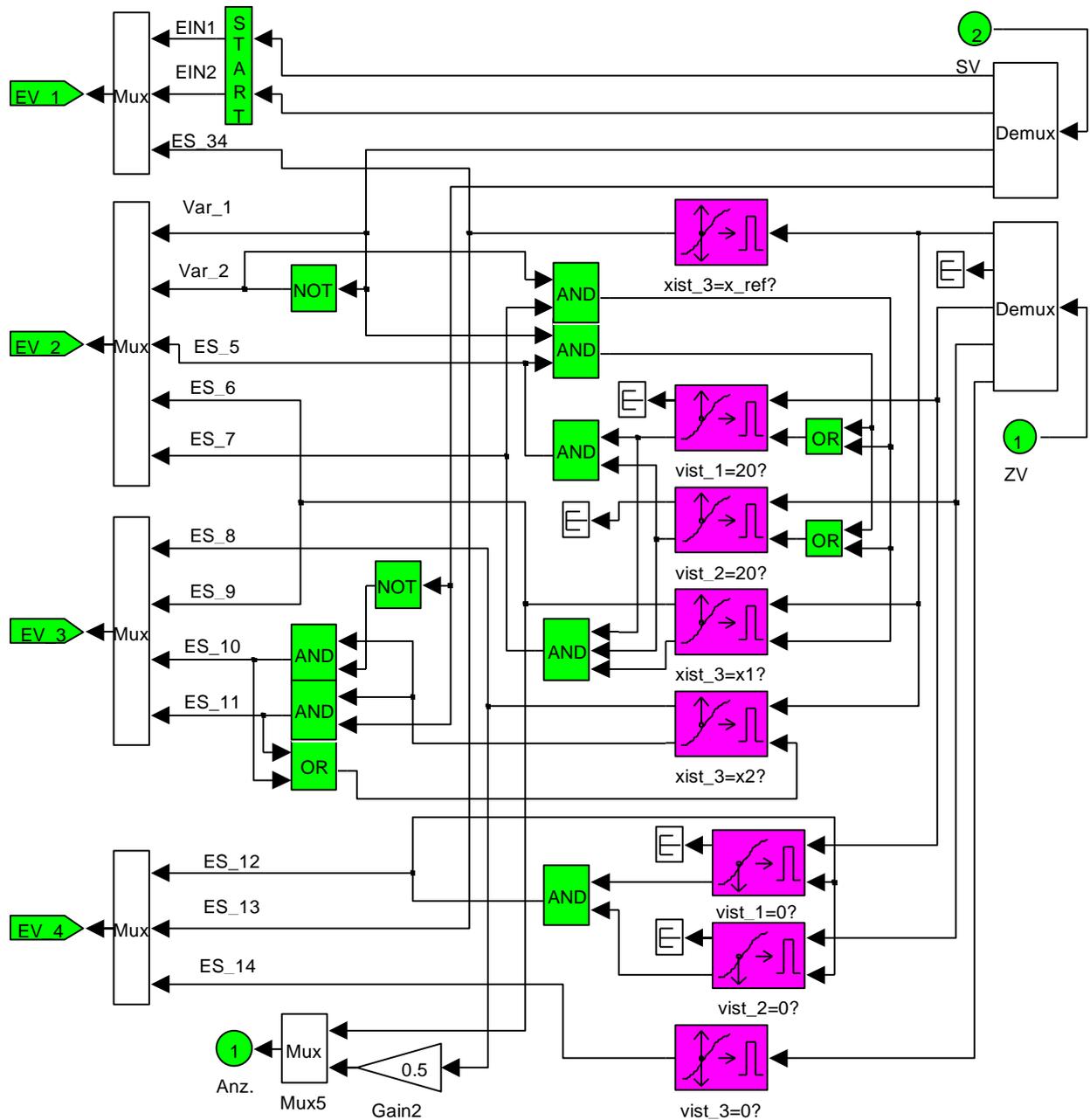
Beispiel 6: Gesamtsystem mit Teilfunktionsplänen, drei Antrieben, logischen Gleichungen für Ereignisse und Sammelfehlerbehandlung - **Bild 1 Gesamtstruktur**



So könnte etwa X30 auch eingespart oder ein erneutes Referenzpunktfahren nach Wiederanlauf bedingungsabhängig vermieden werden. Jeder Teilfunktionsplan erhält Teilereignisvektoren. Für Beispiel 6 ist - im Unterschied zu Beispiel 5 - keine Rückführung eines Zustandsvektors vom Funktionsplan zur Prozeßanpassung erforderlich.

Die mit den Schaltern „EIN“ und „VAR“ möglichen 4 unterschiedlichen Anfahrvorgänge sind deutlich aus den Zeitverläufen nach Beispiel 6 / Bild 7 erkennbar. Als Referenzwinkel ist Winkel „Null“ gewählt. Für eine zur Darstellung der Zeitverläufe wünschenswerte kurze Referenzpunktfahrt wurden für die beiden Schalterstellungen von „EIN“ unterschiedliche zufällige Anfangswinkel vorgegeben. Zur Unterscheidung der technologisch notwendig gleichen Arbeitsgeschwindigkeiten der 3 Antriebe wurden für Anzeigezwecke unterschiedliche Signalwichtungsfaktoren vorgesehen (Beispiel 6 / Bild 1). Das Verhalten bei normalem Beenden des Bewegungsvorganges mit „STOP“ und bei Abschaltung durch „FEHLER“ geht aus Beispiel 6 / Bild 8 bzw. 9 hervor.

Beispiel 6: Gesamtsystem mit Teilfunktionsplänen, drei Antrieben, logischen Gleichungen für Ereignisse und Sammelfehlerbehandlung - **Bild 2 Prozeßanpassung**



Neben den aus Beispiel 5/Bild 4 bekannten Ereignisgeneratorblöcken EG_w/s/f werden erweiterte Ereignisgeneratorblöcke EGE_w/s/f verwendet. Letztere stellen am Ausgang 2 ein statisches Ereignissignal bereit, welches über Eingang 2 rückzusetzen ist. Wie oben ersichtlich sind hiermit beliebige logische Verknüpfungen realisierbar.

ES_34	Antrieb 3 hat einen Referenzwinkel angefahren	ES_9	wie ES_6
ES_5	Antrieb 1 und 2 haben Arbeitsgeschwindigkeit erreicht	ES_10	ES_8 und kein STOP
ES_6	Antrieb 3 hat Winkel 1 erreicht (Siegelbeginn)	ES_11	ES_8 und STOP
ES_7	ES_5 und ES_6 gleichzeitig	ES_12	Antrieb 1 und 2 haben v=0 erreicht
ES_8	Antrieb 3 hat Winkel 2 erreicht (Siegelende)	ES_13	wie ES_34
		ES_14	Antrieb 3 hat v=0 erreicht

Ereignistabelle: Für das Weiterschalten des Funktionsplanes notwendige Ereignisse

Beispiel 6: Gesamtsystem mit Teilfunktionsplänen, drei Antrieben, logischen Gleichungen für Ereignisse und Sammelfehlerbehandlung

Bild 3 Teilfunktionsplan 1 - „Referenzpunktfahrt“

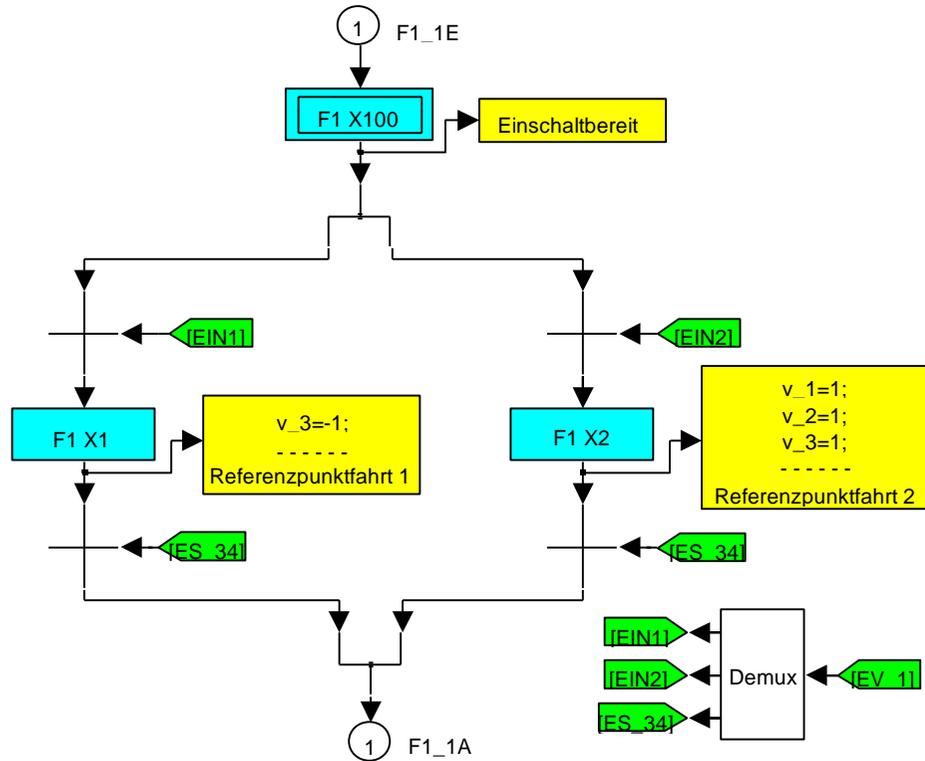
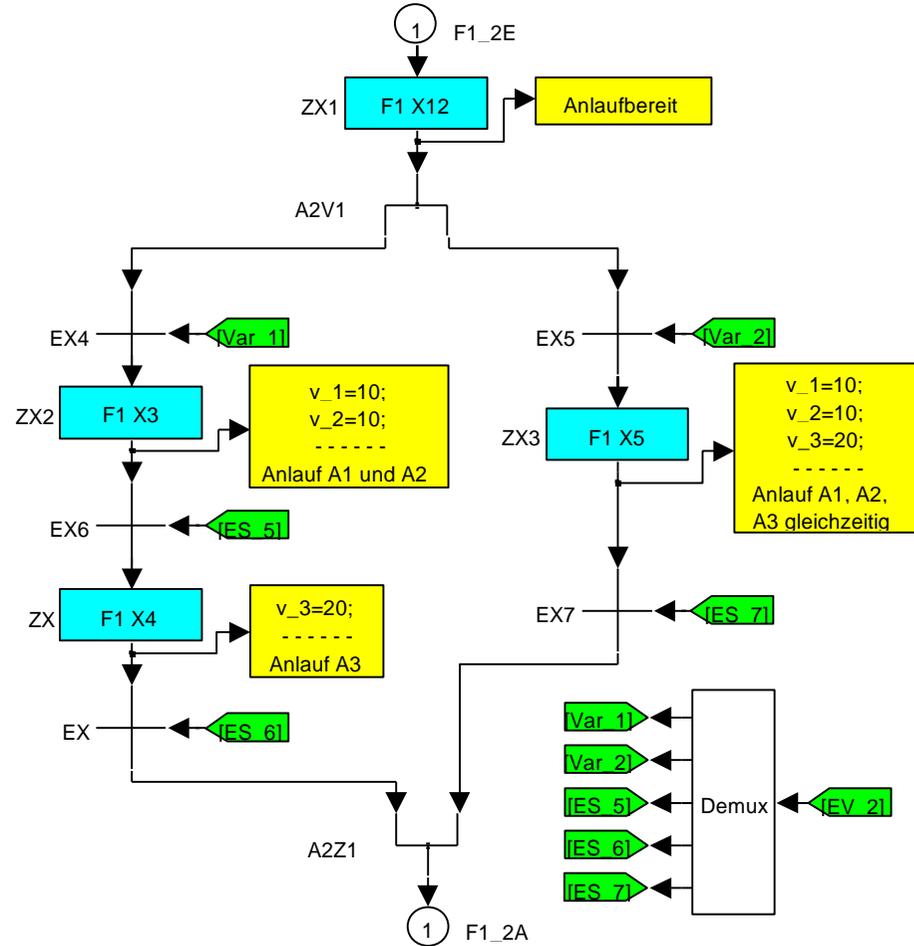


Bild 4 Teilfunktionsplan 2 - „Anlauf“



Beispiel 6: Gesamtsystem mit Teilfunktionsplänen, drei Antrieben, logischen Gleichungen für Ereignisse und Sammelfehlerbehandlung

Bild 5 Teilfunktionsplan 3 - „Normalbetrieb“

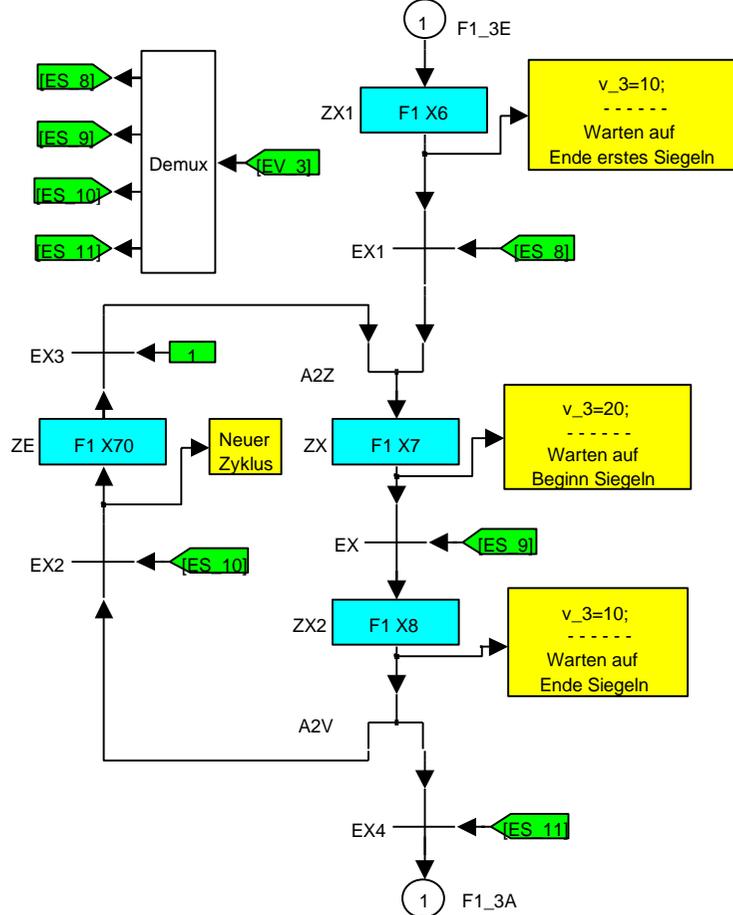
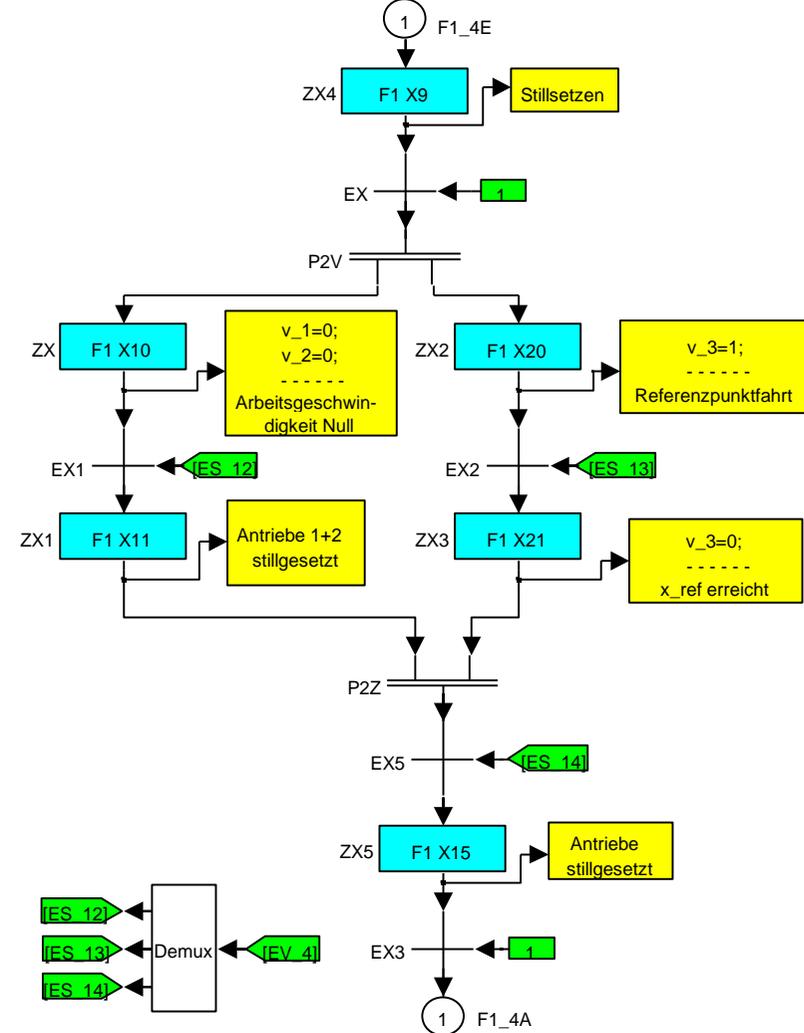


Bild 6 Teilfunktionsplan 4 - „Stillsetzen“



Beispiel 6: Gesamtsystem mit Teilfunktionsplänen, drei Antrieben, logischen Gleichungen für Ereignisse und Sammelfehlerbehandlung - Simulationsergebnisse

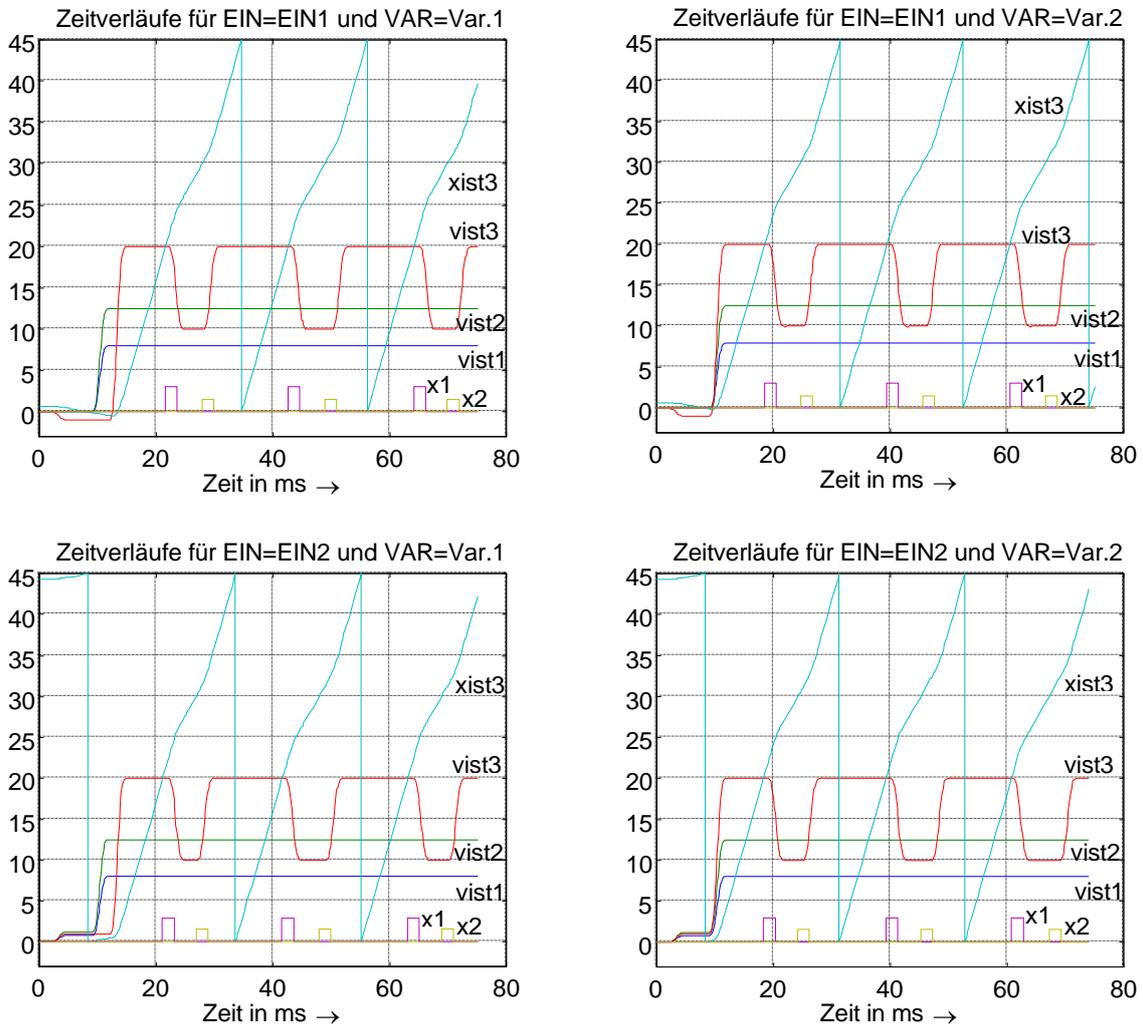


Bild 7 a bis d Anlaufvorgänge

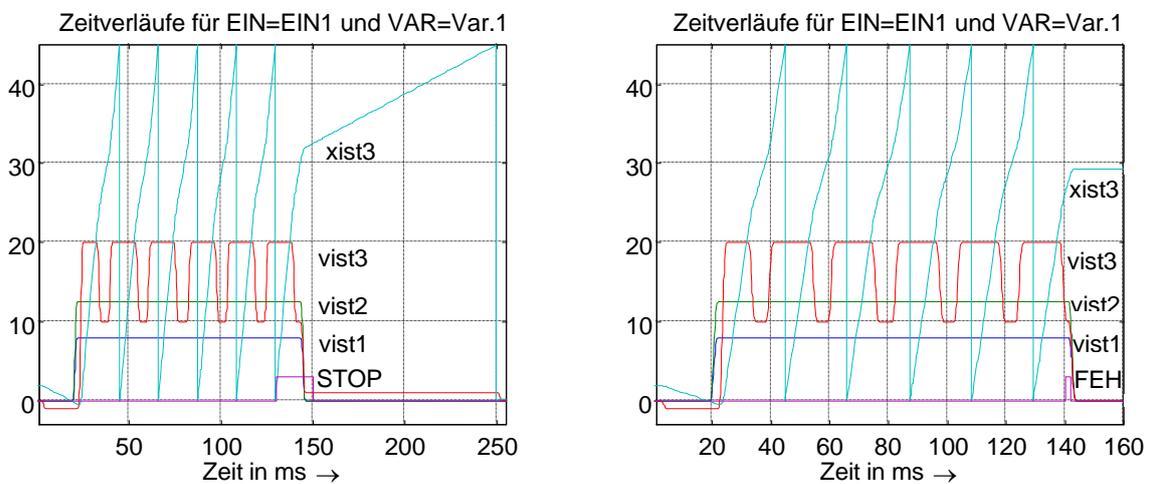
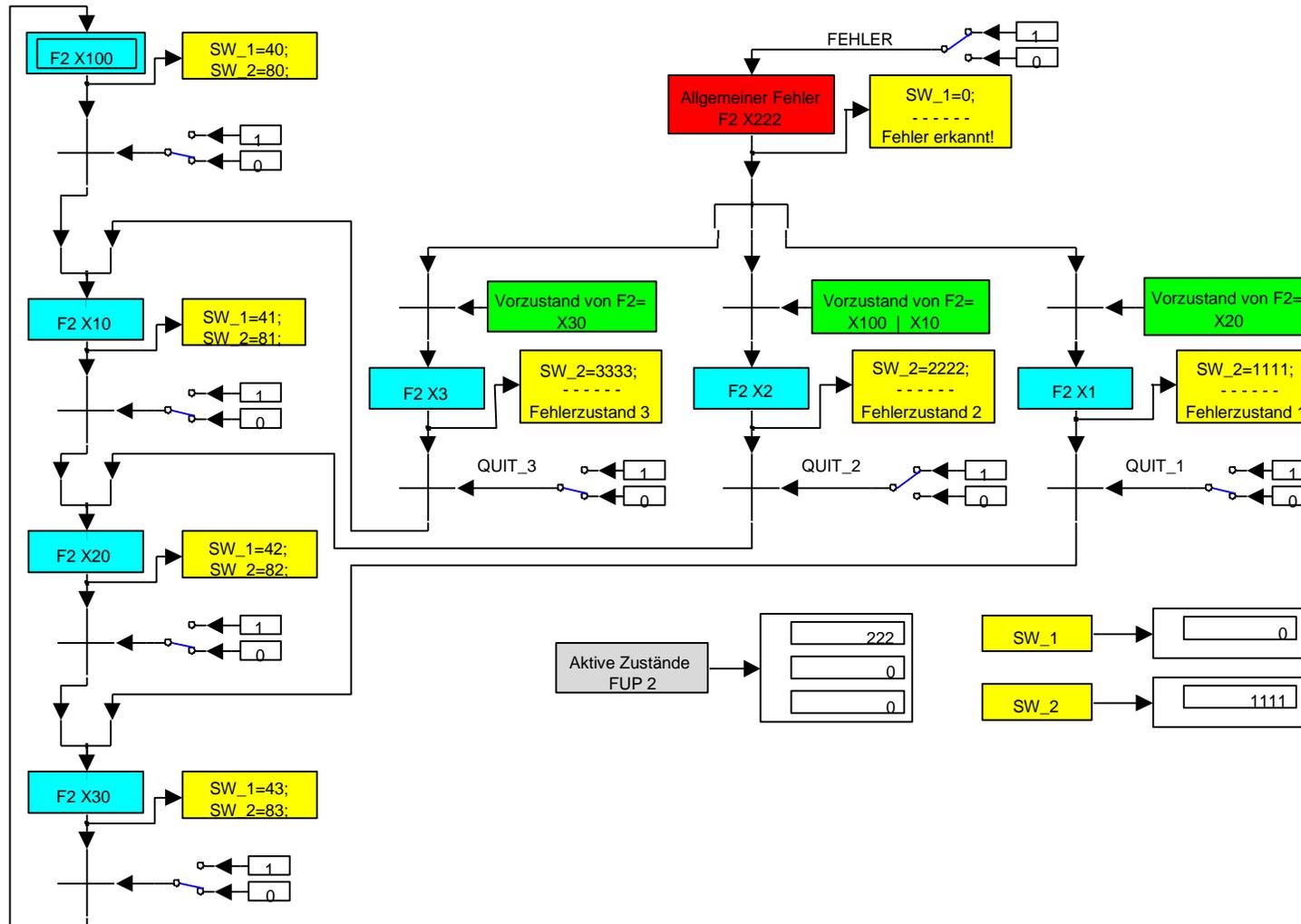


Bild 8 Verhalten bei „STOP“

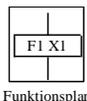
Bild 9 Verhalten bei „FEHLER“

Beispiel 7: Einfache verzweigte Fehlerbehandlung



Statischer Test einer verzweigten Fehlerbehandlung. Der zu realisierende Ablauf besteht aus einer Schleife mit den Zuständen X100, X10, X20 und X30. Ein fiktiver Fehler kann durch den Manual Switch FEHLER simuliert werden. Je nach Vorzustand im Ablauf kann als Reaktion auf den Fehler einer von drei Fehlerbehandlungszuständen X1, X2 oder X3 aktiviert werden. Nach fiktiver Beseitigung des Fehlers erfolgt durch einen zugeordneten Manual Switch QUIT_x der Übergang vom völlig getrennt aufstellbaren Fehlerfunktionsplan zum Normalbetriebsfunktionsplan. Die Übergänge (Verbindungen) sind für das einfache Beispiel direkt ausgeführt. Wie in Beispiel 5 und 6 können mit Vorteil der graphischen Trennung von Normal- und Fehlerablauf auch Konnektorblöcke für den Übergang zum Normalbetrieb eingesetzt werden. Der oder die Wiedereintrittspunkte in den Normalbetrieb sind entsprechend dem technologischen Ablauf frei wähl-

bar. Jeder Fehlerbehandlungsablauf (Zweig) muß mit **genau einem** Block FVZ beginnen (Ausnahme: keine Verzweigung - kein FVZ nötig), danach sind **ereignisabhängige** Verzweigungen möglich.



Funktionsplan

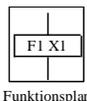


Kürzel	Blockname in Maske	Erläuterungen (Blockbeschreibung / Blockhilfe)
ZA	Anfangszustand	Anfangszustand bei Programmierung eines Funktionsplans (FUP).
		Dieser Block muß einmal pro Funktionsplan (FUP) als Anfangszustand verwendet werden. Mit Start der Simulation wird dieser Zustand aktiv geschaltet. Danach arbeitet dieser Block wie der allgemeine Zustandsblock (ZX) und kann Bestandteil zyklischer Abläufe sein. Funktionsplannummer (FUP-Nummer) muß eine natürliche Zahl zwischen 1 und 9 sein. Zustandsnummer muß eine natürliche Zahl zwischen 1 und 9999 sein. Dieser Bereich kann nach Rücksprache verändert werden. Die Verwendung gleicher Zustandsnummern in unterschiedlichen FUP's ist zulässig. Der Zustandsausgang kann direkt für logische Verknüpfungen (Boolsche Gleichungen) verwendet werden.
ZX	Allgemeiner Zustand	Allgemeiner Zustand in einem Funktionsplan (FUP).
		Dieser Block ist als Anfangszustand unzulässig. Funktionsplannummer (FUP-Nummer) muß eine natürliche Zahl zwischen 1 und 9 sein. Zustandsnummer muß eine natürliche Zahl zwischen 1 und 9999 sein. Dieser Bereich kann nach Rücksprache verändert werden. Die Verwendung gleicher Zustandsnummern in unterschiedlichen FUP's ist zulässig. Der Zustandsausgang kann direkt für logische Verknüpfungen (Boolsche Gleichungen) verwendet werden.
ZE	Allgemeiner Zustand mit Entkopplung	Allgemeiner Zustand in einem Funktionsplan (FUP) mit Entkopplung gegen arithmetische Schleifen.
		Der Block ZE muß jeweils <i>einmal</i> zur Entkopplung gegen arithmetische Schleifen verwendet werden, wenn Rückführschleifen nicht über den Anfangszustand ZA führen. Dieser Block ist als Anfangszustand unzulässig. Funktionsplannummer (FUP-Nummer) muß eine natürliche Zahl zwischen 1 und 9 sein. Zustandsnummer muß eine natürliche Zahl zwischen 1 und 9999 sein. Dieser Bereich kann nach Rücksprache verändert werden. Die Verwendung gleicher Zustandsnummern in unterschiedlichen FUP's ist zulässig. Der Zustandsausgang kann direkt für logische Verknüpfungen (Boolsche Gleichungen) verwendet werden.
ZF	Allgemeiner Fehlerzustand	Allgemeiner Fehlerzustand in einem Funktionsplan (FUP).
		Der Block ZF führt zur Deaktivierung aller bei seiner Aktivierung aktiven Zustände des betreffenden FUP. Er kann z.B. durch eine Sammel Fehlermeldung aktiviert werden. Als Voraussetzung für den Einsatz des FVZ-Blockes werden die Zustandsnummern der Fehlervorzustände bereitgestellt. Pro FUP kann z.Z. jeweils ein Block ZF eingesetzt werden. Dieser Block ist als Anfangszustand unzulässig. Funktionsplannummer (FUP-Nummer) muß eine natürliche Zahl zwischen 1 und 9 sein. Zustandsnummer muß eine natürliche Zahl zwischen 1 und 9999 sein. Dieser Bereich kann nach Rücksprache verändert werden. Die Verwendung gleicher Zustandsnummern in unterschiedlichen FUP's ist zulässig.

EX	Ereignisblock	Ereignisblock für Funktionsplan (FUP)
		Der obere Eingang und der Ausgang müssen an einen Zustandsblock (ZA, ZX, ZE, ZF) angeschlossen werden. Der rechte Eingang dient zum Anschluß eines Ereignissignals. Zur Erzeugung von Ereignissignalen können z.B. die in der Funktionsplan (FUP) Block-Bibliothek enthaltenen Blöcke Ereignisgenerator (EG_w, EG_s, EG_f, EGE_w, EGE_s, EGE_f), Zeitgeber (ZG) oder Fehlervorzustand (FVZ) verwendet werden. Logische Verknüpfungen von Ereignissen müssen vor Anschluß an den Ereignisblock (z.B. in der Prozeßanpassung) programmiert sein. Ausnahme: Verknüpfung von Fehlervorzuständen.
A2V	Zweifache Alternativverzweigung	Ermöglicht eine ODER-Verzweigung des Funktionsablaufes in Abhängigkeit von zwei unterschiedlichen Ereignissen.
		Der Eingang muß an einen Zustandsblock (ZA, ZX, ZE, ZF), die Ausgänge an jeweils einen Ereignisblock (EX) angeschlossen werden.
A3V	Dreifache Alternativverzweigung	Ermöglicht eine ODER-Verzweigung des Funktionsablaufes in Abhängigkeit von drei unterschiedlichen Ereignissen.
		Der Eingang muß an einen Zustandsblock (ZA, ZX, ZE, ZF), die Ausgänge an jeweils einen Ereignisblock (EX) angeschlossen werden.
A2Z	Zweifache Alternativzusammenführung	Ermöglicht eine ODER-Zusammenführung des Funktionsablaufes in Abhängigkeit von zwei unterschiedlichen Ereignissen.
		Die Eingänge müssen an jeweils einen Ereignisblock (EX), der Ausgang an einen Zustandsblock (ZA, ZX, ZE, ZF) angeschlossen werden.
A3Z	Dreifache Alternativzusammenführung	Ermöglicht eine ODER-Zusammenführung des Funktionsablaufes in Abhängigkeit von drei unterschiedlichen Ereignissen.
		Die Eingänge müssen jeweils an einen Ereignisblock (EX), der Ausgang an einen Zustandsblock (ZA, ZX, ZE, ZF) angeschlossen werden.
P2V	Zweifache Parallelverzweigung	Ermöglicht eine UND-Verzweigung des Funktionsablaufes in Abhängigkeit von einem Ereignis.
		Der Eingang muß an einen Ereignisblock (EX), die Ausgänge an jeweils einen Zustandsblock (ZX, ZX, ZE, ZF) angeschlossen werden.
P3V	Dreifache Parallelverzweigung	Ermöglicht eine UND-Verzweigung des Funktionsablaufes in Abhängigkeit von einem Ereignis.
		Der Eingang muß an einen Ereignisblock (EX), die Ausgänge an jeweils einen Zustandsblock (ZA, ZX, ZE, ZF) angeschlossen werden.

P2Z	Zweifache Parallel-zusammenführung	Ermöglicht eine UND-Zusammenführung des Funktionsablaufes in Abhängigkeit von zwei aktiven Zuständen.
		Die Eingänge müssen an jeweils einen Zustandsblock (ZX, ZE), der Ausgang an einen Ereignisblock (EX) angeschlossen werden.
P3Z	Dreifache Parallel-zusammenführung	Ermöglicht eine UND-Zusammenführung des Funktionsablaufes in Abhängigkeit von drei aktiven Zuständen.
		Die Eingänge müssen an jeweils einen Zustandsblock (ZX, ZE), der Ausgang an einen Ereignisblock (EX) angeschlossen werden.
KB	Kommentarblock	String-Wertevorgaben und/oder erläuternde Kommentare zum jeweiligen Zustand.
		Der Eingabevektor muß ein einzelner String sein. Er kann mehrere vollständige Gleichungen nach MATLAB-Definition enthalten. Jede Gleichung muß durch ein Semikolon abgeschlossen werden. Ein leerer String ist zulässig. Der Kommentar muß ebenfalls ein String sein. Ein leerer Kommentar ist zulässig. Zeilenwechsel ist mit "\n" möglich. Der Kommentarblock muß an den Ausgang eines Zustandsblockes (ZA, ZX, ZE, ZF) angeschlossen werden. Empfänger der String-Wertevorgaben ist ein FUP_Variablenblock (FUP_V) durch Eingabe des durch den String definierten Variablennamens. Für jede im Kommentarblock definierte Variable ist ein FUP_Variablenblock (FUP_V) als Empfänger im Prozeßmodell notwendig. Unterschiedliche Kommentarblöcke können Werte für den gleichen FUP_Variablenblock (FUP_V) vorgeben. Das gilt auch bei Programmierung mehrerer Funktionspläne.
FUP_V	FUP_Variablenblock	Empfang von einem Zustand (ZA, ZX, ZE, ZF) durch den Kommentarblock (KB) zugeordneten Wertevorgaben für eine Variable zur Einspeisung in ein Prozeßmodell bzw. eine Bewegungszelle.
		Der FUP_Variablenblock (FUP_V) stellt einen Quellblock für ein Prozeßmodell bzw. eine Bewegungszelle dar. Die Zuordnung zu den String-Wertevorgaben der Kommentarblöcke (KB) erfolgt über die dort definierten Variablennamen. Ein FUP_Variablenblock kann Wertevorgaben von mehreren Kommentarblöcken (KB) - auch unterschiedlicher FUP's - empfangen. Nach Start der Simulation sind zunächst die Anfangswerte des FUP_V Blockes wirksam.
FVZ	Fehlervorzustand	Abfrage und logische Verknüpfung von Zustandsnummern der <u>vor</u> Aktivierung von Block Allgemeiner Fehler (ZF) aktiven Blöcke zur Ereignisgenerierung bei verzweigter Fehlerbehandlung.
		Der Block Fehlervorzustand (FVZ) liefert ein binäres Ausgangssignal und stellt einen Quellblock für den Ereignisblock (EX) dar. Der Block FVZ kann mit einer oder mehreren, nach Boolescher Logik verknüpften, Zustandsnummern initialisiert werden. Ist diese logische Gleichung wahr, so liefert FVZ das Ausgangssignal „1“. Mit Deaktivierung des Blockes Allgemeiner Fehler (ZF) wird das Ausgangssignal zurückgesetzt, d.h. pro alternativem Fehlerpfad im Teilfunktionsplan zur Fehlerbehandlung ist <u>ein</u> FVZ Block zulässig. Die logische Gleichung darf keine Klammern enthalten, Zeilenwechsel mit "\n" ist zulässig. Entsprechend C-Konvention hat "&" (UND) Vorrang vor " " (ODER). „Nicht“ wird entsprechend MATLAB-Konvention mit „~“ eingegeben, „X“ vor der Zustandsnummer ist fakultativ.

ZG	Zeitgeber	Erzeugung von definiert um tx in ms zeitverschobenen Impulsen.
		Der Zeitgeberblock (ZG) liefert am Ausgang ex nach der Zeit tx einen Impuls von definierter Länge. Eingang En startet den Zeitgeber mit positiver Flanke, Eingang ZZ muß mit einer Rechteckfrequenz generiert vom Block Digital Clock (Sources-Bibliothek) gespeist werden.. Die Einstellung des Digital-Clock-Blocks muß unbedingt auf eine ms erfolgen. Bei Bedarf ist die Länge des am Ausgang ex zur Verfügung gestellten Impulses unter der Maske im Block Impulslänge veränderbar. Zwei ms dürfen jedoch nicht unterschritten werden.
EG_w	Ereignisgenerator wechselnd steigende und fallende Flanke	Generiert bei Erreichen des Schwellwertes einen Impuls definierter Länge.
		Der Ereignisgenerator wechselnd steigende und fallende Flanke (EG_w) überprüft das am Eingang angeschlossene Signal auf Erreichen des vorgegebenen Schwellwertes. Bei Eintreten des Ereignisses wird am Ausgang ein Impuls definierter Länge erzeugt. Bei Bedarf ist die Länge des Ausgangsimpulses unter der Maske im Block Impulslänge veränderbar. Zwei ms sollten im Normalfall jedoch nicht unterschritten werden. Erscheint das dennoch aus dynamischen Gründen notwendig, müssen Simulationsauflösung, Zeitgeber etc. entsprechend eingestellt werden.
EGE_w	EG_w erweitert	Generiert bei Erreichen des Schwellwertes einen Impuls definierter Länge <u>und</u> ein quasi statisches Ausgangssignal.
		Siehe EG_w. Ausgang 2 liefert das Ausgangssignal zusätzlich quasi statisch zur Ausführung von logischen Verknüpfungen. Über Eingang 2 kann der quasi statische Ausgang (z.B. durch die Bool'sche Ergebnisvariable der logischen Verknüpfung) zurückgesetzt werden.
EG_s	Ereignisgenerator steigende Flanke	Generiert bei Erreichen des Schwellwertes mit steigender Flanke einen Impuls definierter Länge.
		Der Ereignisgenerator steigende Flanke (EG_s) überprüft das am Eingang angeschlossene Signal auf Erreichen des vorgegebenen Schwellwertes mit steigender Flanke. Bei Eintreten des Ereignisses wird am Ausgang ein Impuls definierter Länge erzeugt. Bei Bedarf ist die Länge des Ausgangsimpulses unter der Maske im Block Impulslänge veränderbar. Zwei ms sollten im Normalfall jedoch nicht unterschritten werden. Erscheint das dennoch aus dynamischen Gründen notwendig, müssen Simulationsauflösung, Zeitgeber etc. entsprechend eingestellt werden.
EGE_s	EG_s erweitert	Generiert bei Erreichen des Schwellwertes mit steigender Flanke einen Impuls definierter Länge <u>und</u> ein quasi statisches Ausgangssignal.
		Siehe EG_s. Ausgang 2 liefert das Ausgangssignal zusätzlich quasi statisch zur Ausführung von logischen Verknüpfungen. Über Eingang 2 kann der quasi statische Ausgang (z.B. durch die Bool'sche Ergebnisvariable der logischen Verknüpfung) zurückgesetzt werden.



Funktionsplan



EG_f	Ereignisgenerator fallende Flanke	<p>Generiert bei Erreichen des Schwellwertes mit fallender Flanke einen Impuls definierter Länge.</p> <hr/> <p>Der Ereignisgenerator fallende Flanke (EG_f) überprüft das am Eingang angeschlossene Signal auf Erreichen des vorgegebenen Schwellwertes mit fallender Flanke. Bei Eintreten des Ereignisses wird am Ausgang ein Impuls definierter Länge erzeugt. Bei Bedarf ist die Länge des Ausgangsimpulses unter der Maske im Block Impulslänge veränderbar. Zwei ms sollten im Normalfall jedoch nicht unterschritten werden. Erscheint das dennoch aus dynamischen Gründen notwendig, müssen Simulationsauflösung, Zeitgeber etc. entsprechend eingestellt werden.</p>
EGE_f	EGE_f erweitert	<p>Generiert bei Erreichen des Schwellwertes mit fallender Flanke einen Impuls definierter Länge <u>und</u> ein quasi statisches Ausgangssignal.</p> <hr/> <p>Siehe EG_f. Ausgang 2 liefert das Ausgangssignal zusätzlich quasi statisch zur Ausführung von logischen Verknüpfungen. Über Eingang 2 kann der quasi statische Ausgang (z.B. durch die Bool'sche Ergebnisvariable der logischen Verknüpfung) zurückgesetzt werden.</p>
AZFx	Aktive Zustände FUP_x	<p>Quellblock zur Anzeige der Zustandsnummern aktiver Zustände eines ausgewählten Funktionsplanes (FUP) mittels Displayblock.</p> <hr/> <p>Pro programmiertem Funktionsplan (FUP) muß bei Bedarf je ein Block Aktive Zustände FUP_x (AZFx) zur Anzeige verwendet werden. Wegen eines Fehlers der Version SIMULINK-April/97 kann die Anzahl der angezeigten Zustände z.Z. nicht dynamisch verändert werden. Der Festeinstellwert beträgt 7. Eine Veränderung dieses Wertes kann nach Rücksprache erfolgen. Wegen des erwähnten Fehlers wird in der letzten Zelle des Displayblocks der Wert Null angezeigt.</p>
DA	Dekodierung Anzeige	<p>Binäre und summarische Überblicksanzeige über beliebige ausgewählte Zustände beliebiger Funktionspläne (FUP) und Ausgabe von Zustands- und Funktionsplannummer eines Einzelnen dieser Zustände.</p> <hr/> <p>Der Eingang Vek_ZA muß mit einem Vektor ausgewählter Zustandsausgänge (auch unterschiedlicher Funktionspläne) und der Eingang NrAkZu mit der Nummer eines Elementes des Vektors Vek_ZA (entspricht einem ausgewählten Zustand) gespeist werden. Der an einen Displayblock angeschlossene Ausgang ZA liefert eine binäre Zustandsanzeige (0 - inaktiv; 1 - aktiv). Über den Ausgang Anz_AkZu kann die Anzahl der gleichzeitig aktiven, im Vektor Vek_ZA zusammengefaßten, Zustände angezeigt werden. Die Ausgänge FN und BN ermöglichen die Anzeige von Funktionsplan- und Zustandsnummer des mit Eingang NrAkZu adressierten Elementes von Eingang Vek_ZA, wenn der Zustand aktiv ist!</p>

SIMULINK Blockbibliothek Funktionsplan - alle Rechte vorbehalten - Dr. Geitner ETI/TU Dresden