

TAP-Controller Modellierung mit Signalflussgraphen

JTAG TAP Controller Modeling Using Signal Flow Graph

Farouk Babba, Sebastian Sattler, Friedrich-Alexander-Universität, Lehrstuhl für zuverlässige Schaltungen und Systeme, 91052 Erlangen, farouk.babba@fau.de, sebastian.sattler@fau.de

Kurzfassung

JTAG ist die bekannte Methode für den standardisierten Testzugang zu einer Hardware. Sie dient dazu, integrierte Schaltungen mit ausschließlich 5 Pins seriell zu testen und zu debuggen, und befindet sich heutzutage in fast jeder integrierten Schaltung. Sie arbeitet im Normalbetrieb unabhängig von der Core-Logik und besteht aus verschiedenen Komponenten wie etwa Register, Gatter und Multiplexer. Die wichtigste Komponente ist der TAP-Controller. Dieser Automat steuert das System, er ist das Herzstück des JTAG. Die strukturtreue Modellierung dieses Automaten unterstützt die Absicherung seiner funktionalen und strukturellen Realisierung. Mit Hilfe von Signalflussgraphen kann das System zweifelsfrei und eindeutig wiedergegeben werden. Darüber hinaus kommen zusätzliche Parametern wie Zeit in Betracht. Der Beitrag stellt nun die strukturtreue Modellierung eines gerichteten Systems anhand von Signalflussgraphen vor. Dies wird sowohl für den flankengesteuerten als auch zustandsgesteuerten Betrieb beispielhaft gezeigt. Das gerichtete Verhalten kann damit in abstrakten Formeln abgebildet werden.

Abstract

JTAG is a known method for standardized test access to hardware. It is used to serially test and debug integrated circuits with only 5 pins. It is found in every integrated circuit and works in normal operation independent of the core logic. This system consists of various components such as registers, gates and multiplexers. The most important component is the TAP controller. This machine controls the system. It is the heart of the JTAG. The structurally faithful modeling of this machine supports the protection of its functional and structural realization. With the aid of signal flow graphs, the system can be reproduced without any doubt and in a uniquely way. In addition, additional parameters such as time are considered. The paper presents the structure-based modeling of a directional system using signal flow graphs. This is exemplarily shown for both the edge-controlled and the condition-controlled operation. The directed behavior can thus be mapped in abstract formulas.

1 Einführung und Motivation

Der TAP-Controller ist ein Zustandsautomat aus 16 Zuständen und 12 Ausgängen. Die Zustände werden dabei mit Hilfe eines Test Clock (TCK) synchronisiert [1]. Der Automat eines TAP-Controllers im IEEE Standard 1149.1 gibt nicht sämtliche Informationen über die Struktur dieses Modells wieder. Dieser Informationsmangel kann beim Design bzw. beim Test zu Schwierigkeiten zwischen Modell und Schaltung führen. Die Abbildung des Modells in der Struktur ist dadurch mehrdeutig.

Beim Testen können sich bei der Ausführung von Testpattern somit Inkonsistenzen ergeben. Die Zeit wird im vom Standard vorgegebenen Modell nicht berücksichtigt. Durch strukturtreue Modellierung jedoch kann das Ausgabeverhalten und die genaue Zuordnung der Funktionalität zur realen Struktur mit abgebildet werden und damit Strukturen für die Abbildung von Zuverlässigkeit besser nachvollzogen werden. Dafür wird die Struktur des Systems in einen Signalflussgraphen (SFG) entwickelt.

Der SFG zeigt, wie sich das System im Zustand stabilisieren lässt. Dies ist bei asynchronen Schaltungen von Bedeutung, da ein Synchronisationspunkt nicht mehr vorhanden ist. Die Robustheit der Schaltung wird

gesteigert gegen verschiedene Störeffekte, da nun direkt Stabilität gewährleistet werden kann. Wir beginnen mit einem D-Flip-Flop (D-FF), um das Muster (die Struktur) des Systems zu erkennen. Daraus wird dann der TAP-Controller strukturtreu modelliert und entwickelt.

2 Modellierung eines D-FF

2.1 Darstellung als Automatengraph

In diesem Abschnitt geht es um die Beschreibung des positiv flankengesteuerten D-Flip-Flop (D-FF) mit Hilfe von Katenation. Die Katenation ist definiert als die Anordnung (konkreter die serielle Verknüpfung) von zwei unabhängigen Elementen durch das Zeichen „.“. Eine nebenläufige Anordnung (konkreter die parallele Verknüpfung) wird als Konkatenation betrachtet und wird mit dem Symbol „+“ bezeichnet. Abb_1 zeigt den zugehörigen Automatengraph des D-FF, Abb_2 einen Ausschnitt aus dem Impulsdiagramm [2]. Die Knoten Q und \bar{Q} in Abb_1 symbolisieren die Zustände. Die gerichteten Kanten zwischen den Knoten beschreiben die Zustandsübergänge. Die angelegten Signale werden an den Kanten als Übergangsausdrücke aufgeführt. Um sicherzustellen, dass Belegungswechsel nur während der Taktflanke akzeptiert werden, muss das D-FF zum

Zeitpunkt des Taktwechsels zwei alternative Zustände annehmen können; einen, falls die Eingangsbedingung CK erfüllt ist, und einen zweiten falls die Bedingung nicht erfüllt ist (\overline{CK}). In diesem zweiten Zustand darf die Ausgabe nicht verändert werden. Der Zwischenzustand wird mit der 1-0-Flanke wieder verlassen.

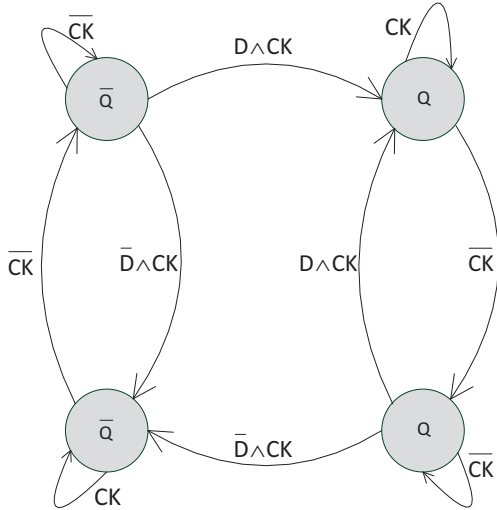


Abbildung 1: Automatengraph (AG)

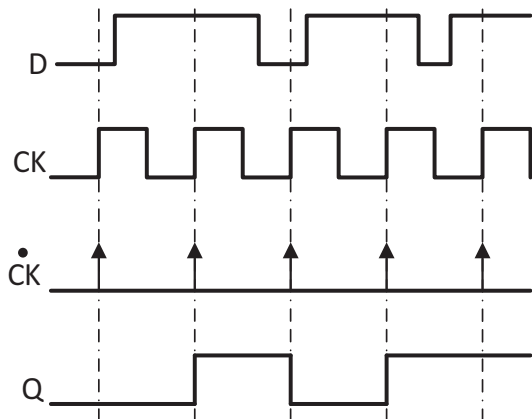


Abbildung 2: Impulsdiagramm (Ausschnitt)

Als erster Zustand wird der linke obere Zustand \overline{Q} betrachtet. Der Zustand wird mit \overline{CK} stabilisiert. Legt man das Signal D (High) an und schaltet anschließend den CK, erfolgt ein Zustandswechsel von \overline{Q} nach Q, wobei Q auch mit \overline{CK} stabil verbleibt (siehe auch Abb_2). Schaltet man den CK vor dem Anlegen des Signals D (Signal bleibt bei \overline{D} (Low)), erfolgt ein Zustandswechsel zu dem linken unteren Zustand \overline{Q} .

Symmetrisch liegt das gleiche Szenario noch einmal für den rechten unteren Zustand Q vor. Die mit \overline{CK} ausgehenden Kanten beschreiben das Schalten und Stabilisieren des CK von 1 nach 0.

Im nächste Schritt wird das flankengesteuerte D-FF in das

Modell eines zustandsgesteuerten gleichwertigen D-Latch überführt. Das Ziel der Transformation ist die Darstellung eines synchronen Systems in einem asynchronen ohne Verlust von Informationen. Die Umwandlung hat den Vorteil das System im Detail nachvollziehbar zu verstehen. Zunächst werden die Zustände betrachtet.

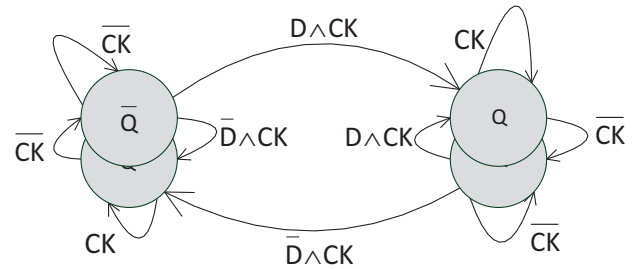


Abbildung 3: Automatengraph als Multi-Set

Die Zustände \overline{Q} und Q sollen in Abb_3 (im Sinne eines Multi-Set) übereinander gelegt werden [3]. Die weitere Analyse geschieht nun im Multi-Set. Eigenschleifen CK und \overline{CK} lassen sich beliebig wiederholen. Als Ausdruck (Katenation) geschrieben und als Graph in Abb_4 dargestellt, gilt:

$$CK \cdot CK \cdot \dots \cdot CK = CK \cdot CK^* = CK^+$$

$$\overline{CK} \cdot \overline{CK} \cdot \dots \cdot \overline{CK} = \overline{CK} \cdot \overline{CK}^* = \overline{CK}^+$$

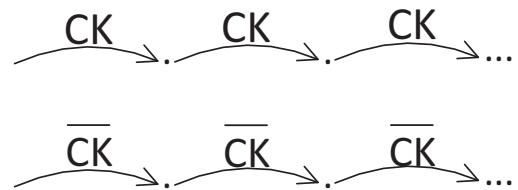


Abbildung 4: Wiederholung von Katenationen

Der Graph in Abb_3 als Signalflussgraph (SFG) besteht also aus Katenationen auf Kanten und Aussagen in Knoten. In der Sprache des Multi-Set liegen im Graph in Abb_4 verschiedene Katenationen aufeinander (sind angeordnet). Dies wird auch als Superposition bezeichnet. Der Graph in Abb_5 gibt dann die sichtbaren Katenationen (Anordnungen) wieder:

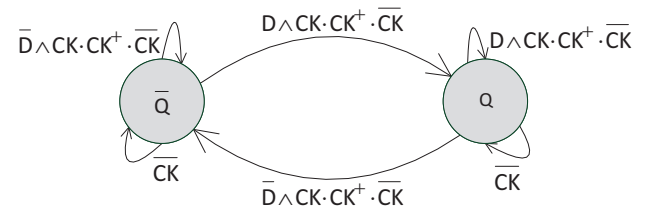


Abbildung 5: SFG des flankengesteuerten D-FF

Der Ausdruck $D \wedge CK \cdot CK^+ \cdot \overline{CK}$ an der von \overline{Q} ausgehenden Kante zwischen \overline{Q} und Q fasst z.B. die Katenationen der Kanten ab dem linken oberen Zustand \overline{Q} bis zu dem rechten unteren Zustand Q (in Abb_6) zusammen.

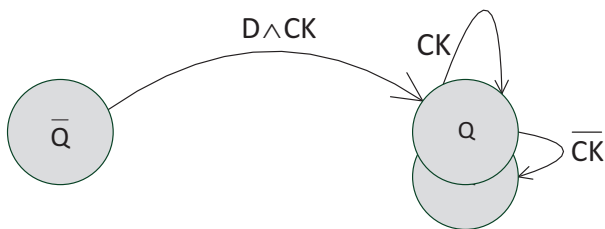


Abbildung 6: Flankenübergang von 0 nach 1

Der Ausdruck $D \wedge CK \cdot CK^+ \cdot \overline{CK}$ an der Eigenschleife des Zustandes Q fasst z.B. die Katenationen der Kanten beginnend mit dem rechten unteren Zustand Q bis zum rechten oberen gleichen Zustand und zurück (in Abb_7) zusammen.

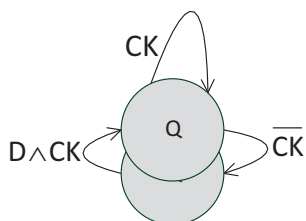


Abbildung 7: Hold (Schleife von 1 nach 1)

Die zweite Hälfte des Graphen in Abb_5 beginnend mit dem Zustand Q und dem Ausdruck $\overline{D} \wedge CK \cdot CK^+ \cdot \overline{CK}$ ist entsprechend der ersten Hälfte nun nachzuvollziehen.

Um nun den Ausdruck $D \wedge CK \cdot CK^+ \cdot \overline{CK}$ in Abb_7 an der Eigenschleife zu entwickeln (zu abstrahieren), geht man in die Darstellung als Katenation (Partial Order - PO) über. Auf diesem Abstraktionsniveau werden unäre, das sind einwertige Aussagen in Großbuchstaben, kateniert (angeordnet). Es werden die folgenden Regeln verwendet:

$$A \cdot \overline{A} = 1, \overline{\overline{A}} = (A, \overline{\overline{A}}),$$

$$(\overline{\overline{A}} = (A, \overline{\overline{A}}) = (\overline{\overline{A}}, \overline{\overline{A}}) = (\overline{\overline{A}}, \overline{A}) = (A, \overline{A}) = A) \Rightarrow (\overline{\overline{A}} = A)$$

In PO spielt die Stelligkeit (Wertigkeit) keine Rolle. Deswegen macht das Vertauschen der Stellen $(\overline{\overline{A}}, \overline{\overline{A}})$ keinen Unterschied. Im Allgemeinen verliert man falls man das positive Literal A konkretisiert hat (zur Existenz erhoben hat) sein Komplement \overline{A} . Mit Hilfe der Substitution nimmt man dann irgendeinen Buchstaben (z. B. das negative Literal \overline{A}) und verknüpft es mit dem Komplement von A. Das ergibt letztendlich das Substitut des Komplements von A als $\overline{\overline{A}}$ ($\overline{\overline{A}} = A$ Komplement). Durch diese Substitution ist $\overline{\overline{A}}$ ein existenter Buchstabe (negatives Literal), welcher das Komplement von A, das ist \overline{A} , darstellt [4]. Das Komplement ist der Rest eines konkretisierten Objekts. Es ist daher wichtig zwischen Negation und Komplement zu unterscheiden. Sie sind nur bei Vorhandensein der Dualität gleich zu interpretieren. Die Katenation $CK \cdot CK^+ \cdot \overline{CK}$ beschreibt den Übergang des Signals von Low auf High und stellt die nicht gerichtete Ableitung \dot{CK} dar, $CK \cdot CK^+ \cdot \overline{CK} = \dot{CK}$. Der Ausdruck $D \wedge CK \cdot CK^+ \cdot \overline{CK}$ lässt sich dann in $D \wedge \dot{CK}$ substituieren. Mit dem Anteil von \overline{D} wird genauso vorgegangen. Der resultierende neue Graph ist in Abb_8 dargestellt:

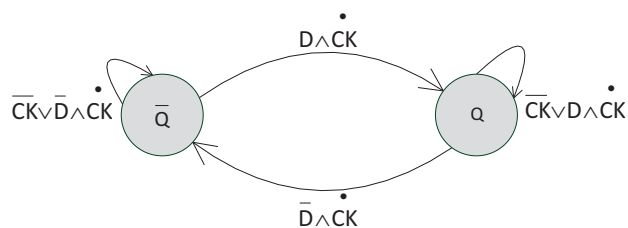


Abbildung 8: Sichtweise auf ein flankengesteuertes D-FF

Es befinden sich zwei ausgewiesene Katenationen an jedem Zustand, im Zustand Q zum Beispiel die Eigenschleifen \overline{CK} und $D \wedge \dot{CK}$. Diese Darstellung wird dann parallel komponiert (konkateniert) und es folgt der (aussagenlogische) Graph in Abb_9:

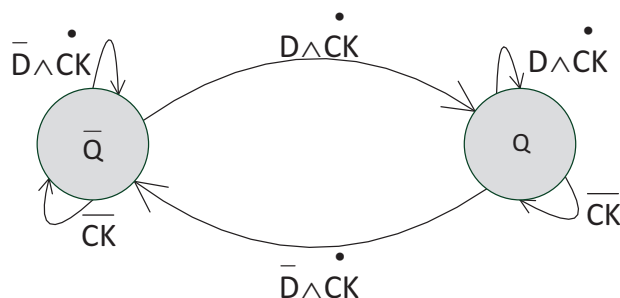


Abbildung 9: flankengesteuertes D-FF

Man erkennt nun die Struktur des zustandsgesteuerten D-Latch. Die Struktur des positiv flankengesteuerten transparenten D-FF kann also mit der Struktur des zustandsgesteuerten D-Latch gleichgesetzt (substituiert) werden. Die Beschriftung an den Kanten beinhaltet allerdings mehr Informationen, da ja eine Flankensteuerung beschrieben werden soll. Die Zeit ist abgebildet in der Transition \dot{CK} .

3 TAP-Controller

3.1 Automatenstruktur

Die Beschreibung des flankengesteuerten D-FF ist nun Ausgangspunkt, um den TAP-Controller eines JTAG Systems zu analysieren. Der Controller besteht aus zwei Eingängen nämlich TCK und TMS sowie verschiedenen Ausgängen, die den Automat in drei verschiedene Typen aufteilen (partitionieren). TCK steht für Test Clock und TMS für Test Mode Select. Die Struktur des Automaten mit Zustandsüberföhrungs-funktion δ und Ausgabe-funktionen μ_1, λ_1 und μ_2 ist in der Abb_10 als Blockschaltbild dargestellt. Der TAP-Controller ist vom Typ alt. Die Zustände werden über den TCK synchronisiert. Die Zustandsüberföhrungsfunktion hat die Eingänge X und Z^a , mit X für das TMS Signal, das aus einem Register bei TCK fallender Flanke zur Verfügung gestellt wird, und Z^a für den alten Zustand. Das Signal X geht hervor aus

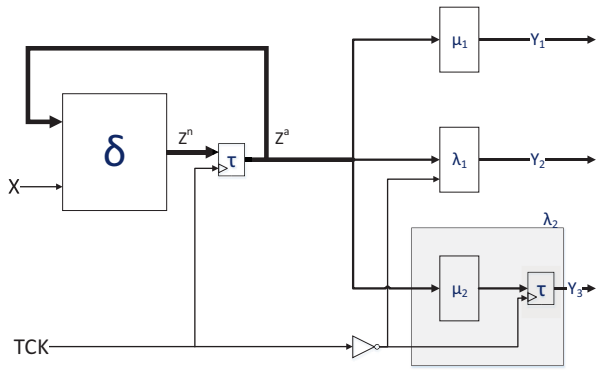


Abbildung 10: TAP Controller mit Ausgabefunktionen

$$X = \frac{TMS}{\dot{}}$$

TMS wird ab TCK negativer Flanke immer wieder für die Dauer eines Taktes zur Verfügung gestellt. Dieses Signal wird durch die Teilstruktur in Abb_11 erzeugt:

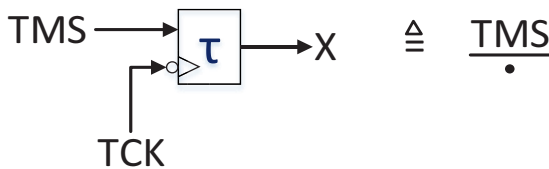


Abbildung 11: Zuführung des TMS Signals

3.2 Modellieren in Signalfussgraphen

Für den ersten Teil des Automaten mit der Zustandsüberföhrungsfunktion δ und dem Synchronisationspunkt τ wird in der Abb_12 das ausführliche SFG-Pattern des TAP-Controllers dargestellt. Die Zustände (allgemein Zustandsmengen) werden als Z und \bar{Z} bezeichnet. Der Zustandswechsel von \bar{Z} nach Z erfolgt mit $\frac{TMS}{\dot{}} \wedge TCK$ und von Z nach \bar{Z} mit $\frac{\overline{TMS}}{\dot{\phantom{\overline{TMS}}}} \wedge TCK$.

Das zugrundeliegende SFG-Pattern ist nochmals im Detail in Abb_13 gegeben. Die großen Kreise umfassen die Menge der Zustände, in denen die entsprechenden Kanten ein- bzw. ausgehen. Der Kreis oben rechts beinhaltet die Zustände, die mit der eingehenden Kante $TMS \cdot TCK$ geschaltet werden.

Die Zustände stabilisieren sich mit der Eigenschleife TCK und \overline{TCK} . Mit \overline{TCK} wird TCK wieder auf 0 geschaltet. Die Zustände sind Reset, Select-DR, Update-DR, Select-IR, Update-IR, Exit1-DR, Exit1-IR, Exit2-DR und Exit2-IR.

In Abb_13 kommen einige Zustände nebeneinander in einer graudunklen Ellipse zweimal vor. Das liegt an unterschiedlichen Ausgängen bei gleichem Zustand.

Die Zustände, die geschaltet werden mit eingehender Kante $\overline{TMS} \cdot TCK$, sind im Kreis unten links zu finden und sind Pause-DR, Pause-IR, Shift-DR, Shift-IR, Capture-DR, Capture-IR und Idle. In diesem SFG sind alle Informationen enthalten, die ermöglichen eine entsprechende Schaltung strukturtreu zu entwerfen. Zum einen ist dies das wiederkehrende TCK Muster der

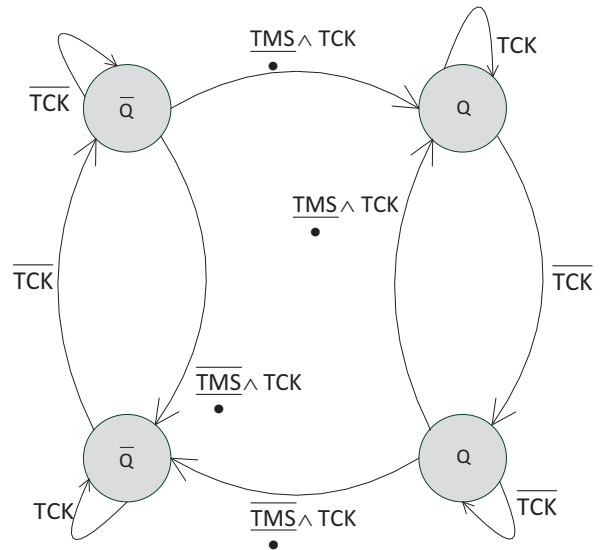


Abbildung 12: TAP Controller (Überföhrungsfunktion)

Stabilisierungspunkte des Systems, zum anderen ist dies die Ausgabe, die sich in den Zuständen wiederfindet. Aus dem SFG ist es möglich direkt die Struktur sowie das Impulsdiagramm zu extrahieren. Damit können Mehrdeutigkeiten und Fehler beim Test schneller festgestellt werden. Für eine gleichwertige Darstellung in der Struktur eines zustandsgesteuerten D-Latch legt man die oberen Kreise auf die unteren Kreise und erhält Abb_14. Damit wurde die Sichtweise auf den SFG ohne Verlust von Informationen verändert.

3.3 Darstellung der Ausgabefunktionen

Im diesem Abschnitt werden die Ausgänge des TAP-Controllers betrachtet. Der Automat lässt sich bzgl. seiner Ausgabefunktionen in drei Teilautomaten partitionieren, nämlich einen Moore Automaten mit dem Ausgang Y_1 , einen Mealy Automaten mit dem Ausgang Y_2 (wenn man \overline{TCK} als Eingang betrachtet) und einen ausgabeflankengesteuerten Moore Automaten mit dem Ausgang Y_3 .

3.3.1 Moore Automat

Der erste Automat ist in Abb_15 gegeben. Die Ausgabe ist negativ flankengesteuert und vom Zustand Z^a allein abhängig (Moore Automat). Die Ausgaben im TAP-Controller sind Select und UpdateDRState ([1] Seite 34). Die Ausgabe wird in folgenden Schritten nachvollzogen:

$$Y_1 = \mu_1(Z^a),$$

$$Z^a = Z^n \cdot \overline{TCK} \quad \text{und}$$

$$Z^n = \delta(Z^a, X) \quad \text{mit } X = TMS \cdot \overline{\dot{TCK}} = \frac{TMS}{\dot{}} \text{ bzw.}$$

$$Y_1 = \mu_1(\delta(Z^a, TMS \cdot \overline{\dot{TCK}}) \cdot \overline{TCK}).$$

Je nach axiomatischer Sichtweise ist die Schreibweise der Ausgabe verschieden. In Multi-Set Schreibweise (PO) werden Katenationen angeordnet, es ergibt sich

$$Y_1 = \mu_1(\delta(Z^a, TMS \cdot \overline{\dot{TCK}}) \cdot \overline{TCK}).$$

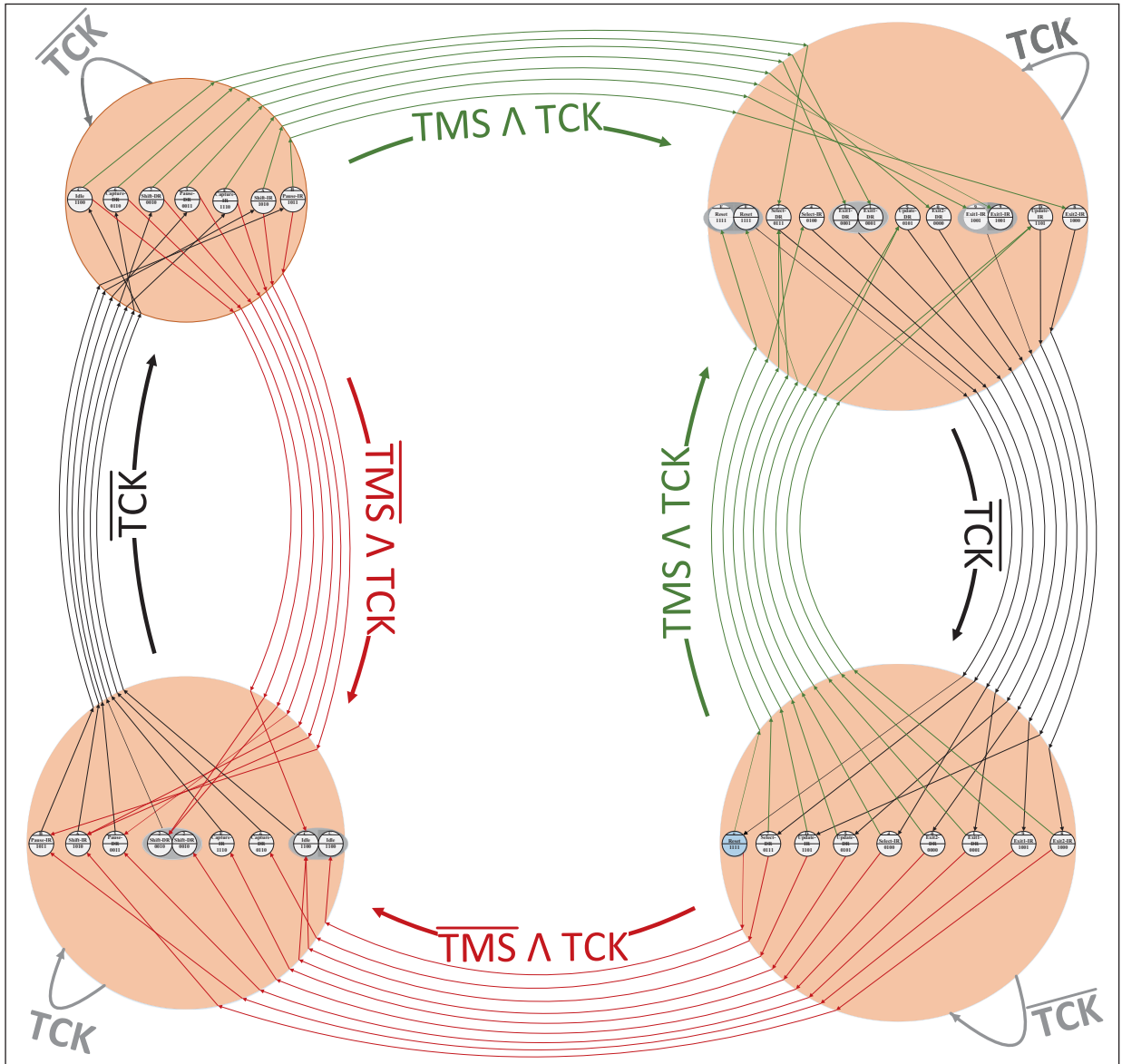


Abbildung 13: TAP Controller (State Diagramm mit Zustandssteuerung)

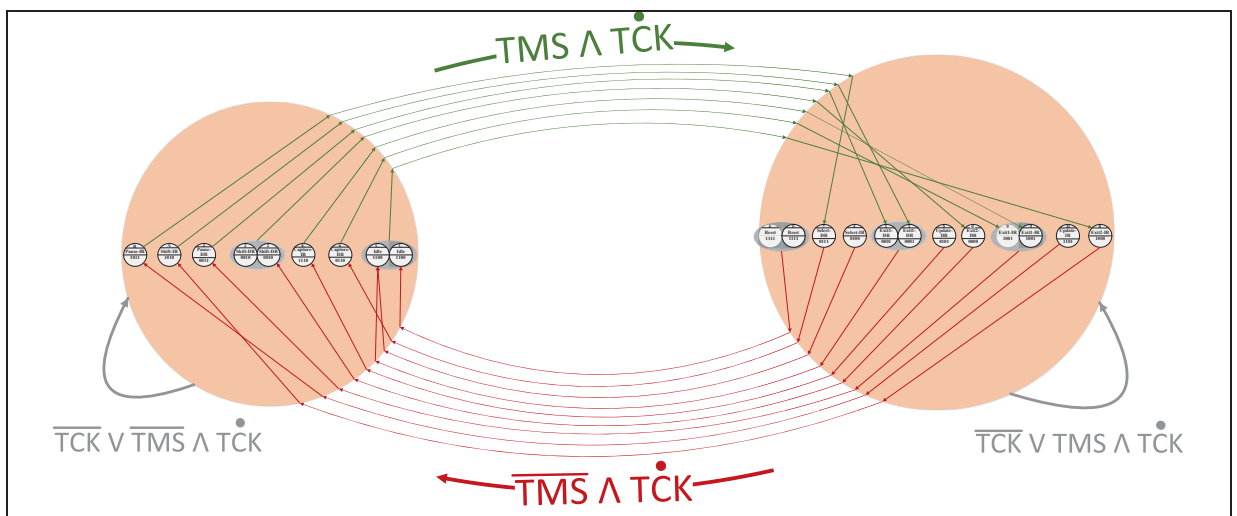


Abbildung 14: TAP Controller (State Diagramm)

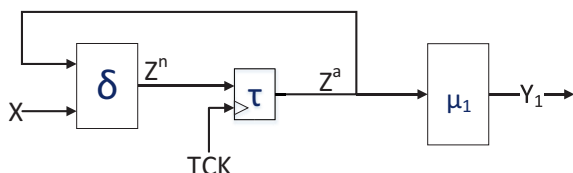


Abbildung 15: Moore Automat vom Typ alt

Eine andere Schreibweise ist die Kompositionsschreibweise [5]. Die Funktionen μ_1 , δ und τ werden als Funktoren bzw. Abbildungen betrachtet

$$Y_1 = \mu_1(\tau(\delta(\tau(\text{TMS}, \overline{\text{TCK}}), Z^a), \text{TCK})).$$

In einer konkreteren Sichtweise kann die Ausgabe auch in Wirkungsschreibweise angegeben werden [5]. Die Funktionen δ und τ sind dann Generatoren und die Eingänge TCK und TMS Variablen

$$Y_1 = (((\text{TMS}, \overline{\text{TCK}})\tau, Z^a)\delta, \text{TCK})\mu_1.$$

3.3.2 Mealy Automat

Der zweite Automat mit der zusätzlich vom Zustand TCK gesteuerten Ausgabe Y_2 ist in Abb_16 gegeben. Die Ausgabe hängt von Z^a und $\overline{\text{TCK}}$ ab.

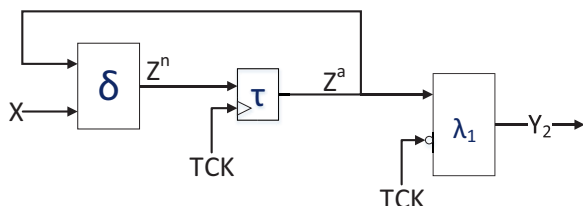


Abbildung 16: Mealy Automat vom Typ alt

Die Ausgaben im TAP-Controller sind UpdateIR, ClockIR, UpdateDR und ClockDR. Wie beim letzten Automat wird die Ausgabe als Formel entwickelt:

$$Y_2 = \lambda_1(Z^a, \overline{\text{TCK}}),$$

$$Z^a = Z^n \cdot \text{TCK} \quad \text{und}$$

$$Z^n = \delta(Z^a, X) \quad \text{mit } X = \text{TMS} \cdot \overline{\text{TCK}} = \frac{\text{TMS}}{\dot{\text{TCK}}} \text{ bzw.}$$

$$Y_2 = \lambda_1(\delta(Z^a, \text{TMS} \cdot \overline{\text{TCK}}) \cdot \text{TCK}, \overline{\text{TCK}}).$$

In Multi-Set Schreibweise ergibt sich

$$Y_2 = \lambda_1(\delta(Z^a, \text{TMS} \cdot \overline{\text{TCK}}) \cdot \text{TCK}, \overline{\text{TCK}}),$$

in Kompositionsschreibweise

$$Y_2 = \lambda_1(\tau(\delta(\tau(\text{TMS}, \overline{\text{TCK}}), Z^a), \text{TCK}), \overline{\text{TCK}})$$

und in Wirkungsschreibweise

$$Y_2 = (((\text{TMS}, \overline{\text{TCK}})\tau, Z^a)\delta, \text{TCK})\tau, \overline{\text{TCK}}\lambda_1.$$

3.3.3 Moore (Ausgabe flankengesteuert)

Der letzte Automat mit der Ausgabe Y_3 ist in der Abb_17 dargestellt. Die Ausgabe ist um eine halbe Periode vom TCK verzögert. Sie ist von Z^a und $\overline{\text{TCK}}$ abhängig (Ausgabe flankengesteuert). Die Struktur stellt sich wie ein Pipeline dar und ist gegen Störungen stabiler. Die Ausgaben im TAP-Controller sind Rest, Enable, ShiftIR, ShiftDR, CaptureIR und CaptureDR.

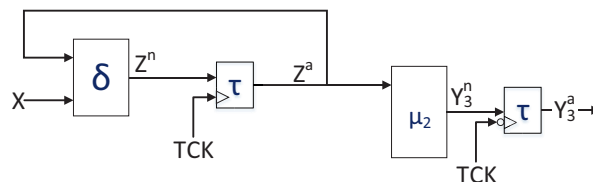


Abbildung 17: Moore (Ausgabe flankengesteuert)

Die Ausgabe lässt sich wie folgt entwickeln:

$$Y_3^a = Y_3^n \cdot \overline{\text{TCK}},$$

$$Y_3^n = \mu_2(Z^a),$$

$$Z^a = Z^n \cdot \text{TCK} \quad \text{und}$$

$$Z^n = \delta(Z^a, X) \quad \text{mit } X = \text{TMS} \cdot \overline{\text{TCK}} = \frac{\text{TMS}}{\dot{\text{TCK}}} \text{ bzw.}$$

$$Y_3 = \mu_2(\delta(Z^a, \text{TMS} \cdot \overline{\text{TCK}}) \cdot \text{TCK}) \cdot \overline{\text{TCK}}.$$

In Multi-Set Schreibweise ergibt sich

$$Y_3 = \mu_2(\delta(Z^a, \text{TMS} \cdot \overline{\text{TCK}}) \cdot \text{TCK}) \cdot \overline{\text{TCK}},$$

in Kompositionsschreibweise

$$Y_3 = \tau(\mu_2(\tau(\delta(\tau(\text{TMS}, \overline{\text{TCK}}), Z^a), \text{TCK})), \overline{\text{TCK}})$$

und in Wirkungsschreibweise

$$Y_3 = (((\text{TMS}, \overline{\text{TCK}})\tau, Z^a)\delta, \text{TCK})\tau, \overline{\text{TCK}}\mu_2, \overline{\text{TCK}}\tau.$$

4 Zusammenfassung und Ausblick

Die in diesem Beitrag behandelte Darstellung einer TAP-Controller Struktur anhand von Signalflussgraphen zeigt anschaulich die Möglichkeit auf, ein System strukturtreu, d.h. verhaltenstreu und gerichtet abzubilden. Man kann so vom Graph in die Struktur eineindeutig abbilden ohne Informationen über die Richtung zu verlieren.

Diese Modellierung hilft ein zuverlässigeres und robusteres System zu entwerfen, da die ingenieurhafte, typisch gerichtete Denkweise im Signalfluss nun in der Mathematik direkt abgebildet wird. Störungen wie Hazardfehler und geeignete gerichtete Gegenmaßnahmen können damit strukturtreu nachgebildet werden, komplexe Systeme lassen sich leichter verstehen und Fehler werden schneller verifiziert.

5 Literatur

- [1] IEEE Standard Association, IEEE Standard for Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1™-2013
- [2] Wuttke, H.-D; Henke, K, Schaltsysteme: Eine automatenorientierte Einführung, Pearson Studium 2003
- [3] Uygur G., Sattler S., A New Approach for Modeling Inconsistencies in Digital-Assisted Analog Design, Electronic Testing, vol. 32, no. 4, pp. 491–503
- [4] Goessel, M., Automatentheorie für Ingenieure, Akademie Verlag 1991
- [5] Ehrig, H., Pfender, M., Kategorien und Automaten, Walter de Gruyter 1972