

Der Logikanalysator

Der Logikanalysator (Logic Analyzer) ist im wesentlichen ein digitaler Ablaufspeicher, der dazu dient, das zeitliche Verhalten digitaler Signale zu erfassen und darzustellen. Eine gewisse Zahl digitaler Signale wird daraufhin überwacht, ob bestimmte, zuvor eingestellte Belegungen auftreten (Triggerbedingungen). Ist dies der Fall, werden die Signalbelegungen in gewissen Abständen abgetastet und gespeichert. Die Zeitpunkte der Abtastung und Speicherung können durch einen von außen gelieferten Takt bestimmt (State Analysis) oder intern vorgewählt werden (Timing Analysis). Der Speicherinhalt wird (üblicherweise durch einen Mikroprozessor) zur (Bildschirm-) Darstellung aufbereitet. Abb. 1 zeigt ein typisches Blockschaltbild.

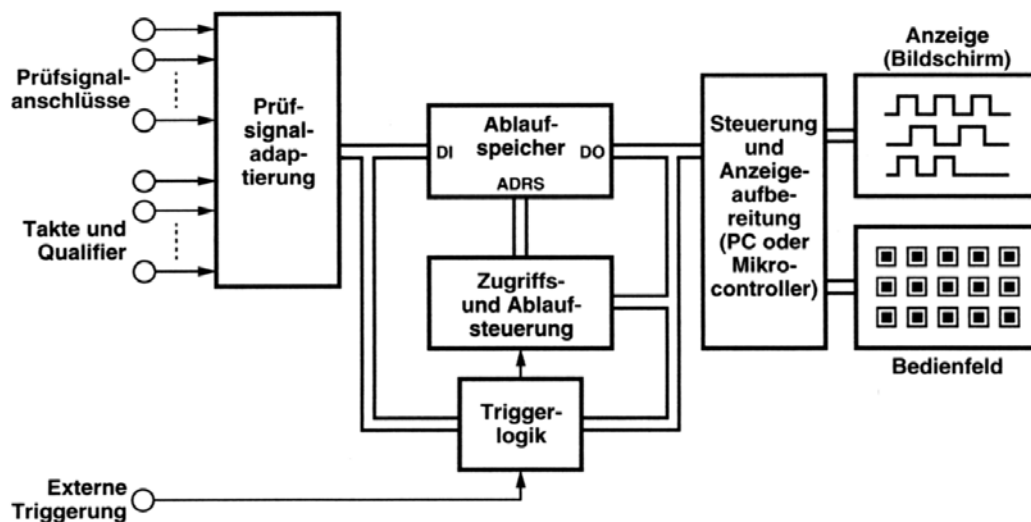


Abb. 1 Blockschaltbild eines Logikanalysators

1. Überblick über die Funktionseinheiten

Prüf-signal-adaptierung

Der Logikanalysator erfaßt binäre Signale. Für jedes Prüf-signal ist ein Kanal (Channel) vorgesehen. Typische Kanalzahlen sind 8, 16, 24, 32 usw. Im Gegensatz zum digitalen Speicheroszilloskop (DSO) hat aber jeder Kanal nur 1 Bit Auflösung; es werden nur Einsen oder Nullen erfaßt und gespeichert.

Die allgemeinen technischen Anforderungen¹⁾ sind unter der Bedingung zu erfüllen, daß im Gegensatz zu anderen Meßgeräten (z. B. Oszilloskopen) eine wesentlich größere Anzahl von Signalen adaptiert werden muß.

Die Schaltmittel der Signalbewertung sind üblicherweise in vom eigentlichen Prüfgerät abgesetzten Anschlußkästen (Probe Pods) untergebracht. Zur eigentlichen Adaptierung im Prüfling (DUT, Zielsystem) dienen kurze Zwischenkabel, die mit Steckkontakten oder Klemmprüfspitzen verbunden sind. Die Anschlußkästen werden mit dem eigentlichen Prüfgerät über Flach- oder Rundkabel verbunden.

1): (1) keine Beeinträchtigung des Prüflings, (2) keine Einkopplung von Störungen in die Prüfleitungen, (3) keine Signalverfälschungen auf dem Wege vom Prüfling zum Ablaufspeicher.

Hinweise:

1. Die praktische Brauchbarkeit eines Logikanalysators wird in starkem Maße dadurch bestimmt, wie einfach (bzw. wie umständlich und zeitaufwendig) eine bestimmte DUT adaptiert werden kann.
2. Die Verbindung zwischen Probe Pod und Prüfgerät ist oft mit Flachbandkabeln ausgeführt und mechanisch nicht immer besonders robust.
3. Manche Logikanalysatoren haben jeweils ein Probe Pod für einen Teil der Kanäle (z. B. ist für 4 oder 8 Kanäle je ein Probe Pod vorgesehen).
4. Zu verschiedenen Logikanalysatoren gibt es spezielle Vorschalteneinrichtungen (Preprozessoren) zur schnellen Adaptierung an bestimmte Zielsysteme (z. B. an verschiedene Interfaces und Mikroprozessoren).

Ablaufspeicher mit Zugriffs- und Ablaufsteuerung

In der einfachsten Ausführung ist der Ablaufspeicher (Acquisition Memory) ein RAM mit Adreßzähler, in den die binären Prüfsignale gemäß einem bestimmten Taktraster eingetragen werden. Die Speichertiefe ist dann ein Maß dafür, wieviele Einsen und Nullen je Kanal gespeichert werden können. Typische Speichertiefen liegen zwischen 512 bis hin zu 8M und mehr. Preisgünstigere Logikanalysatoren haben üblicherweise Speichertiefen zwischen 1k und 16k. Bei manchen Geräten sind Speichertiefe und Kanalzahl wechselseitig voneinander abhängig (je mehr Kanäle, um so geringer die Speichertiefe je Kanal).

Triggerlogik

Die Triggerlogik entspricht in ihrer Wirkung der Triggerung beim digitalen Speicheroszilloskop (DSO). Sie bestimmt, wann die Aufzeichnung beginnt (Posttrigger) bzw. wann eine laufende Aufzeichnung angehalten wird (Pretrigger). Üblicherweise läßt sich vorgeben, welcher Anteil der Speichertiefe für Signalverläufe vor und welcher für Signalverläufe nach dem Triggerereignis reserviert wird (Midtrigger). Wie beim DSO entscheidet die Flexibilität des Triggersystems wesentlich über die Brauchbarkeit des Gerätes bei schwierigen Fehlersuchproblemen.

Steuerung, Bedienung, Anzeige

Diese Funktionen werden praktisch ausnahmslos von Mikroprozessoren wahrgenommen. Neuere preisgünstige Geräte sind auf einen PC angewiesen. Der eigentliche Analysator ist ein Kästchen, das über eine gängige Schnittstelle (heutzutage meist USB) mit dem PC verbunden wird. Manche Modelle der oberen Preisklassen haben komplette PCs eingebaut. Bei Stand-alone-Geräten (mit eigenen Bedien- und Anzeigemitteln) gibt es beträchtliche Unterschiede in der Art der Anzeige (LCD-Bildschirm oder Bildröhre), in der Anzeige-Auflösung, in der Gestaltung des Bedienfeldes und im Bedienkomfort. Schnittstellen zum Verkoppeln mit anderen Geräten und mit PCs gehören zum Stand der Technik.

2. Signaldarstellungen

Die aufgezeichneten Signale können üblicherweise als Impulsdiagramm (Abb. 2) oder in Tabellenform (Abb. 3) dargestellt werden. Eine Tabellendarstellung kann man meist zwischen verschiedenen Darstellungsweisen umschalten (Tabelle 1).

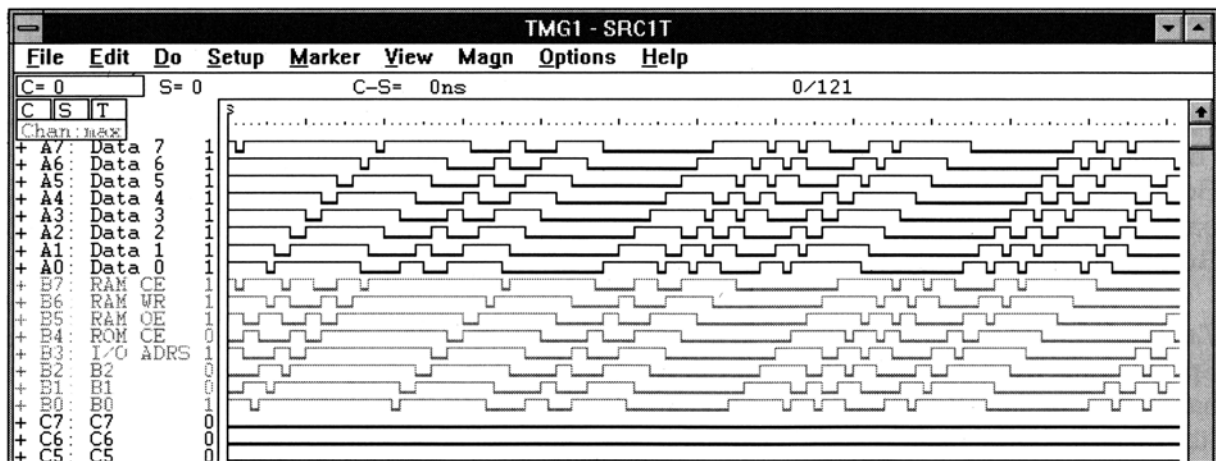
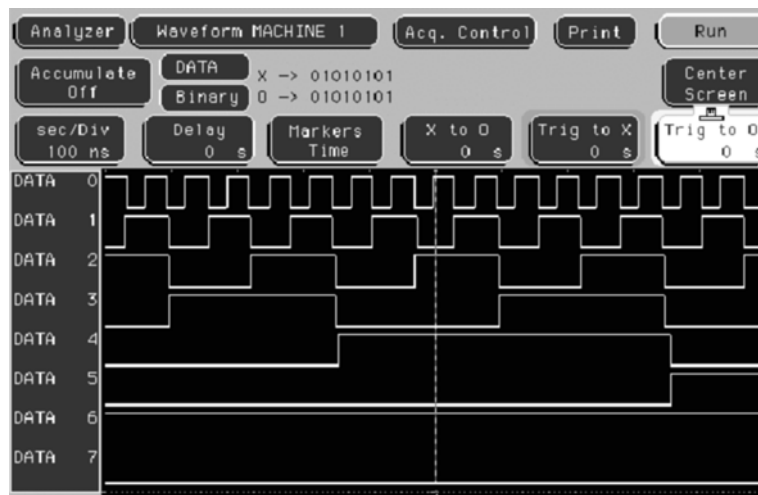


Abb. 2 Beispiele der Impulsdigrammdarstellung (oben HP/Agilent, unten DLI)

	A	B	C	D
Loc	+Bin	+Hex	+Hex	+Hex
-CS----	11111111	E9	00	00
1	01111111	69	00	00
2	11111111	D3	00	00
3	11111111	D2	00	00
4	11111111	E7	00	00
5	11111110	A5	00	00
6	11111101	EF	00	00
7	11111101	4B	00	00
8	11111011	97	00	00
9	11111011	97	00	00
10	11110111	AF	00	00
11	11110111	2F	00	00
12	11101111	5F	00	00
13	11101111	5F	00	00
14	11011111	BF	00	00

Abb. 3 Beispiel einer Tabellendarstellung (DLI)

Art der Darstellung	Symbole	eine Tabellenspalte betrifft
binär	0 und 1 oder L und H; manchmal zusätzliche Sonderzeichen	1 Kanal
ASCII	alle Zeichen des ASCII-Zeichensatzes	bis zu 8 Kanäle
oktal	0...7	bis zu 3 Kanäle
hexadezimal	0...9, A..F	bis zu 4 Kanäle
dezimal	0...9	typischerweise werden 4, 8 oder 16 Kanäle zu einer mehrstelligen Zahlenangabe zusammengefaßt

Tabelle 1 Darstellungsweisen der Tabellendarstellung

Mnemonicische Anzeige

Diese Anzeigeform ist beim Prüfen von Mikroprozessorsystemen oder von standardisierten Interfaces (z. B. SCSI) von Bedeutung. Hierbei werden die in den Kanälen aufgezeichneten Bitmuster gemäß der Assemblersprache des Mikroprozessorsystems oder gemäß den Begriffsbildungen des Interfacestandards in mnemonicische Bezeichnungen umgeschlüsselt und entsprechend dargestellt, beispielsweise eine bestimmte Datenbusbelegung eines Mikroprozessors als ADD-Befehl oder eine SCSI-Interfacebelegung als RESELECTION-Busphase.

3. Aufzeichnungsverfahren

Zeitanalyse (Timing Analysis, asynchrone Abtastung)

Die Signale werden mit einem im Analysator gebildeten Takt abgetastet und gespeichert. In dieser Betriebsart verhält sich der Logikanalysator praktisch wie ein digitales Speicheroszilloskop mit einer Amplituden-Auflösung von einem Bit je Kanal.

Flankengesteuerte Abtastung (Transitional Sampling)

Dies ist ein Aufzeichnungsverfahren der Zeitanalyse, das den Speicher besser ausnutzt. Bei hohen Abtastraten ist herkömmlicherweise der Speicher in kurzer Zeit gefüllt. Hingegen werden beim Transitional Sampling die zu erfassenden Signale auf Änderungen hin überwacht. Hat sich etwas geändert, wird die geänderte Belegung zusammen mit einer Zeitangabe (der abgelaufenen Zeit seit der letzten Änderung) abgespeichert. Besonders vorteilhaft ist das Verfahren beim Erfassen stoßweiser Aktivitäten (Bursts), deren Folge immer wieder von Zeiten der Inaktivität unterbrochen wird. Eine solche Betriebsweise ist für moderne Systeme charakteristisch. Beispiel: ein Prozessor mit eingebautem Cache, der nur gelegentlich den Bus belegt, dann aber meistens eine schnelle Folge von Buszugriffen ausführt.

Zustandsanalyse (State Analysis, synchrone Abtastung)

Die Zeitpunkte der Abtastung und Speicherung werden durch einen Takt bestimmt, der vom Prüfling selbst geliefert wird. Die Logikanalysatoren haben üblicherweise besondere Eingänge für externe Takte und für Takterlaubnisignale (Clock Qualifiers). Die Aufzeichnung erfolgt dann, wenn die jeweils spezifizierte Flanke (Low-High oder High-Low) des Taktsignals auftritt und die zugehörigen Qualifier-Signale gültig (valid) sind (vgl. Abb. 10). Der Vorteil: Es werden nur die Vorgänge aufgezeichnet, die wirklich von Interesse sind.

Die Probleme:

- wir müssen die Zeitverhältnisse der Signale untereinander (einschließlich Takt und Qualifier) einigermaßen kennen – und darauf vertrauen, daß die tatsächlichen Signalverläufe den Annahmen einigermaßen entsprechen. Ist das nicht der Fall, zeichnet der Analysator Unsinn auf),
- was zwischendrin passiert, bekommen wir nicht mit (und das ist nicht selten am Fehlverhalten der DUT schuld).

Zweifachanalyse

Es gibt Geräte, die Signalverläufe gleichzeitig im Sinne der Zeitanalyse und im Sinne der Zustandsanalyse abtasten und speichern. Der Vorteil: Einsparung von Meßzeit. Hat man auf Grundlage der Zustandsanalyse einen Fehler erkannt, so kann man in den Aufzeichnungen der Zeitanalyse nachsehen, ob etwas Verdächtiges – z. B. ein Glitch – zu erkennen ist. Mit einem herkömmlichen Gerät müßte man hingegen die Messung in der Zeitanalyse-Betriebsart wiederholen. Damit verringert sich natürlich die Wahrscheinlichkeit, dieselben Fehleranzeigen unter denselben Bedingungen nochmals zu erwischen.

4. Grundlagen der Erfassung binärer Signale

Die entscheidende Anforderung: ein digitales Prüfmittel muß das zu prüfende Signal genau so bewerten wie die zu prüfende Schaltung:

- ein Signal, das die Gatter und Flipflops der DUT als logische 1 sehen, darf das Prüfmittel nicht als logische 0 bewerten und umgekehrt,
- ein Datensignal, das in Bezug auf einen Takt in der DUT als gültig anerkannt und beispielsweise als logische 1 gespeichert wird, muß in Bezug auf denselben Takt auch vom Prüfmittel als gültig erkannt und als logische 1 gespeichert werden,
- ein Impuls, der in der DUT eine Wirkung auslöst, muß auch vom Prüfmittel als Impuls erkannt werden,
- ein Impuls, der (z. B. weil er zu schmal oder weil seine Amplitude zu niedrig ist) in der DUT keine Wirkung veranlaßt, sollte vom Prüfmittel nicht als gültiger Impuls erkannt werden.

Digitale Prüfmittel sind ihrerseits synchrone (taktgesteuerte) Schaltwerke (Abb. 4). Die Problemstellen der Signalerfassung liegen (1) in der Signalbewertung (d. h. in der Entscheidung, ob das Signal als Low oder als High anzusehen ist) und (2) in der Synchronisation bzw. Eintaktierung (d. h. in der taktgesteuerten Übernahme in die Erfassungshardware).

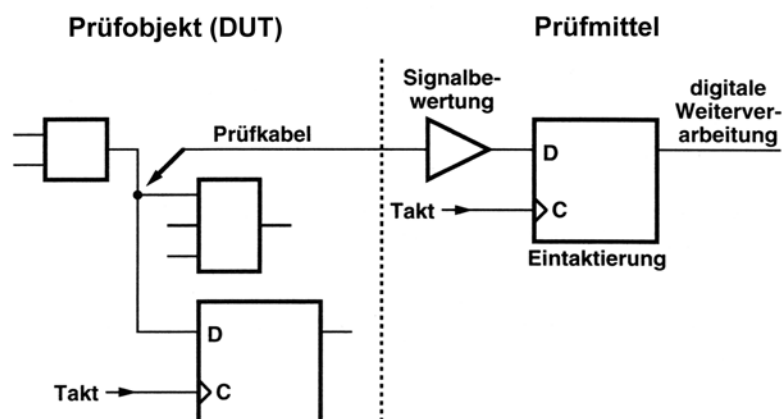


Abb. 4 Die Erfassung binärer Signale mit einem Prüfmittel, das als synchrones Schaltwerk ausgebildet ist

Signalbewertung

Abb. 5 veranschaulicht die wesentlichen Problemstellen.

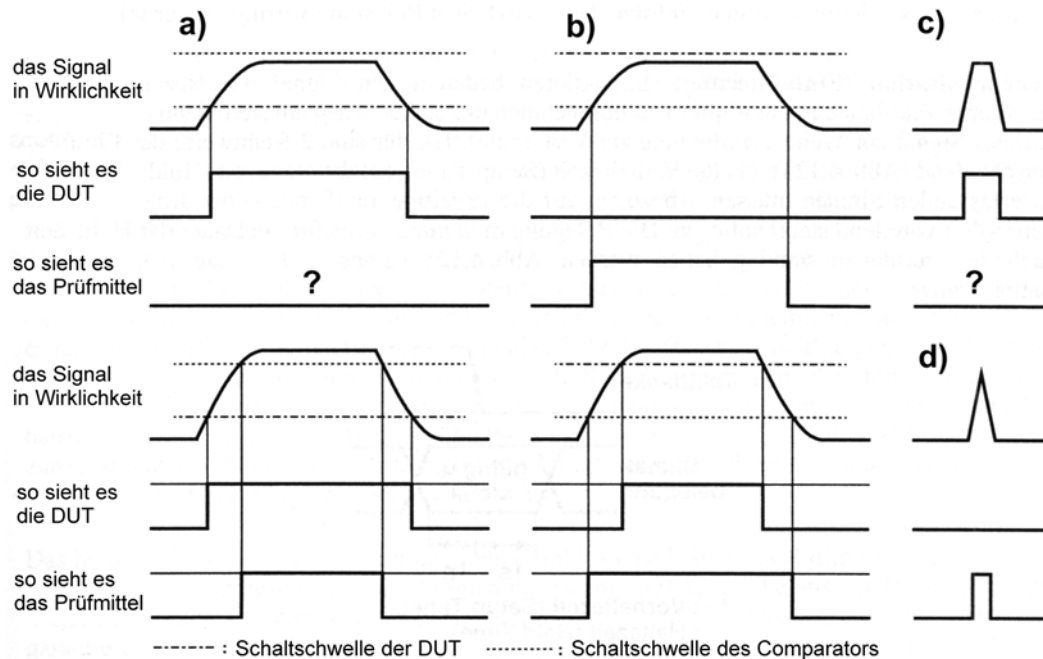


Abb. 5 Problemstellen der Signalbewertung

- die Schaltschwelle des Comparators ist höher als jene der DUT-Schaltkreise: das Prüfmittel sieht Signale, die in der DUT wirken, entweder gar nicht, oder es sieht Impulse, die kürzer sind als in Wirklichkeit,
- die Schaltschwelle des Comparators ist niedriger als jene der DUT-Schaltkreise: das Prüfmittel erfasst entweder Signale, die in der DUT gar nicht wirksam sind, oder es sieht Impulse, die länger sind als in Wirklichkeit,
- die Antwortzeit (Response Time) des Comparators ist zu lang: das Prüfmittel kann kürzere Impulse – die in der DUT wirksam sind – gar nicht erfassen,
- der Comparator ist zu empfindlich; seine Antwortzeit ist kurz, sein Schwellwert aber zu niedrig: Signalformen, die in der DUT gar nicht als Impulse wirken, werden als richtige Impulse erfasst.

Hinzu kommt die Belastung des zu prüfenden Signals durch Prüflleitung und Comparator. Dies wirkt sich vor allem als zusätzliche kapazitive Last aus (Amplituden werden vermindert, Flanken verschliffen) und kann zwei Effekte haben:

- nach der Adaptierung des Prüfmittels hat sich das Fehlerbild wahrnehmbar geändert (oft funktioniert gar nichts mehr),
- nach Adaptierung des Prüfmittels ist der Fehler verschwunden (der typische Fehlerverdacht: Störimpulse, deren Amplitude infolge der zusätzlichen Belastung verringert wurde).

Synchronisation (Eintaktierung)

Eintaktieren bedeutet, ein Signal mit Bezug auf eine bestimmte Taktflanke in ein Flipflop zu übernehmen, so daß an dessen Ausgang ein taktsynchrones Signal zur Weiterverarbeitung zur Verfügung steht. Hierfür sind zwei Kennwerte des Flipflops entscheidend (Abb. 6): (1) die Vorhaltezeit (Setup Time), (2) die Haltezeit (Hold Time). Die zu erfassenden Signale müssen – bezogen auf die jeweilige Taktflanke – mindestens um die Setup-Zeit voreilend stabil anliegen. Die Belegung muß mindestens für die Dauer der Hold-Zeit nach der Taktflanke stabil gehalten werden. Abb. 7 veranschaulicht die verschiedenen Betriebsfälle, Tabelle 2 gibt einen Überblick über die Setup- und Haltezeiten verbreiteter Logikbaureihen.

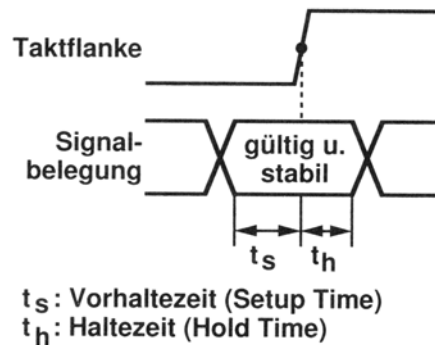


Abb. 6 Kritische Zeiten beim Eintaktieren

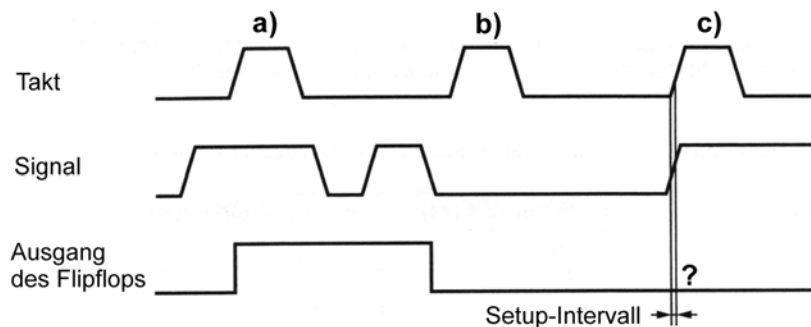


Abb. 7 Betriebsfälle bei der Eintaktierung

- alles o. k. Das Signal ist im Setup-Hold-Intervall stabil High und wird deshalb auch als High eintaktiert.
- auch o. k. Das Signal hat gerade rechtzeitig vor Beginn des Setup-Hold-Intervalls auf Low geschaltet und wird deshalb auch als Low eintaktiert.
- das Signal schaltet innerhalb des Setup-Hold-Intervalls um. Mit welchem Logikpegel es eintaktiert wird, ist demzufolge ungewiß.

Hinweis:

Was geschieht, wenn sich die Eingangsbelegung eines Flipflops im Setup-Hold-Intervall einer schaltenden Taktflanke ändert, hängt ganz vom Zufall ab; das Flipflop wird entweder seinen Zustand ändern oder den bisherigen beibehalten. Metastabile Zustände können wir ausschließen (die Prüfmittel haben entsprechende Auswerteschaltungen).

Das heißt in der Praxis: wir müssen die Setup-Hold-Spezifikation des Prüfmittels genau beachten. Das betrifft im besonderen alle Formen des Bezugs auf Takte, die aus der DUT selbst entnommen werden

(Zustandsanalyse). Nur Signale, die bei der Erfassung garantiert nicht innerhalb des Setup-Hold-Intervalls umschalten, werden auch verlässlich aufgezeichnet. Ansonsten sehen wir Signalverläufe, die es in Wirklichkeit gar nicht gibt. Es könnte beispielsweise vorkommen, daß ein Flipflop in der DUT schaltet, das mit demselben Datensignal und demselben Takt angesteuerte Flipflop im Prüfmittel aber nicht (oder umgekehrt). Also: überlegen, ob die zu prüfenden Signale die Setup-Hold-Spezifikation einhalten. Im Zweifelsfall: die Annahme überprüfen (z. B. mittels Oszilloskop).

Baureihe	Setup-Zeit [ns]	Hold-Zeit [ns]	Baureihe	Setup-Zeit [ns]	Hold-Zeit [ns]
74	20	5	74ABT	1,8...2,5	0,7...1,3
74S	3	2	74HC/HCT	20	5
74AS	2	0	74AC/ACT	3,5	0
74F	3	1	74LV	16	3
74LS	20	5	74LVC	4	2
74ALS	15	0	74ALVC	1,1	1,5
74BCT	3	1	74LVT/LVZ	2,7	0

Tabelle 2 Setup- und Haltezeiten verbreiteter Logikbaureihen

Zeitversatz zwischen verschiedenen Signalen (Skew)

Werden mehrere Signale gleichzeitig erfaßt, so sind die Verzögerungszeiten der Signalbewertung nicht genau gleich. Ebenso verhalten sich die einzelnen Eintaktierungs-Flipflops im Setup-Hold-Intervall unterschiedlich. Der Zeitversatz des Prüfmittels addiert sich zum Zeitversatz der DUT (z. B. zwischen den einzelnen Leitungen eines Datenbus). Das heißt: (1) das Prüfmittel sollte einen möglichst geringen Skew haben, (2) bei der Untersuchung, ob die Setup-Hold-Anforderungen erfüllt werden, ist der ungünstigste Fall des Gesamt-Skew anzusetzen.

Störungen in den Prüfsignalwegen

Vor allem die Wege von der Adaptierung in der DUT zum Probe Pod sind anfällig gegen Störungen (Ground Shift, gegenseitiges Übersprechen, Einstrahlung von außen). Abhilfe: für jeden Kanal oder für Gruppen weniger Kanäle sind eigene Masseleitungen vorgesehen (die in der DUT jeweils in der Nähe der betreffenden Signale adaptiert werden müssen).

Hinweis:

Der Masseanschluß des Probe Pods sollte in der DUT in der Nähe der erfaßten Signale adaptiert werden. Wer ihn über Krokodilklemme und 1 m Laborkabel an der Gehäuse-Masse adaptiert (oder gar nicht anschließt!), sollte sich nicht wundern, wenn er Mist mißt.

Ungenauigkeiten der Abtastung

Wird asynchron, also mit einem vom Prüfmittel erzeugten Takt abgetastet (Zeitanalyse), so ist damit zu rechnen, daß wir (genauer: das Prüfgerät) die Signale anders sehen als die DUT (Abb. 8).

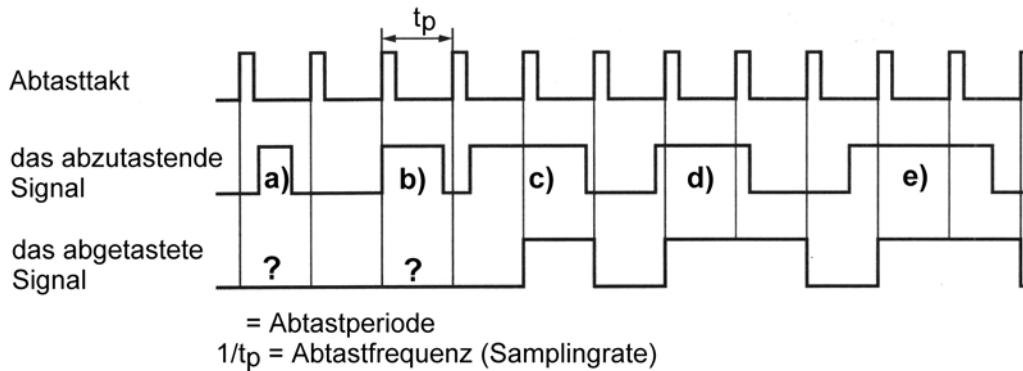


Abb. 8 Ungenauigkeiten der (asynchronen) Abtastung

- a) ein Impuls, der kürzer ist als die Abtastperiode, kann verlorengehen. Hinweis: Daß wir nichts mitbekommen, was zwischen den Abtastungen passiert, ist ein Problem aller Abtastverfahren.
- b) der längste Impuls, der verlorengehen kann (also gar nicht erfaßt wird), ist etwa 1 Abtastperiode (t_p) lang. Er geht dann verloren, wenn seine Flanken mit zwei aufeinanderfolgenden Setup-Hold-Intervallen zusammenfallen, das erfassende Flipflop aber nicht schaltet. Damit ein Impuls sicher erfaßt werden kann, muß seine Breite mindestens $t_p + 2 \cdot \text{Setup-Hold-Intervall}$ betragen (um ganz sicherzugehen: Mindestimpulsbreite $> 2 t_p$).
- c) der Impuls sieht kürzer aus, als er ist. Wenn seine Rückflanke unmittelbar auf ein Setup-Hold-Intervall folgt, kann er am Anfang um nahezu eine Abtastperiode (t_p) verkürzt werden.
- d) der Impuls sieht länger aus, als er ist. Wenn seine Vorderflanke unmittelbar auf ein Setup-Hold-Intervall folgt, kann er am Ende um nahezu eine Abtastperiode (t_p) verlängert werden.
- e) infolge der Synchronisation mit dem Abtasttakt erscheinen alle erfaßten Impulse mehr oder weniger zeitverschoben. Die maximale Zeitverschiebung ist näherungsweise eine Abtastperiode (t_p); die (statistisch gesehen) mittlere Zeitverschiebung entspricht $t_p/2$.

Glitcherkennung

Die Glitcherkennung ist vorgesehen, um Ereignisse zwischen den Abtastzeitpunkten zu erkennen. Digitale Prüfgeräte werten üblicherweise jede Signaländerung als Glitch, die zwischen zwei Abtastzeitpunkten wenigstens zwei Flanken aufweist (Abb. 9). Glitches werden registriert und angezeigt, es können aber weder die genaue zeitliche Lage noch der genaue Signalverlauf erkannt werden.

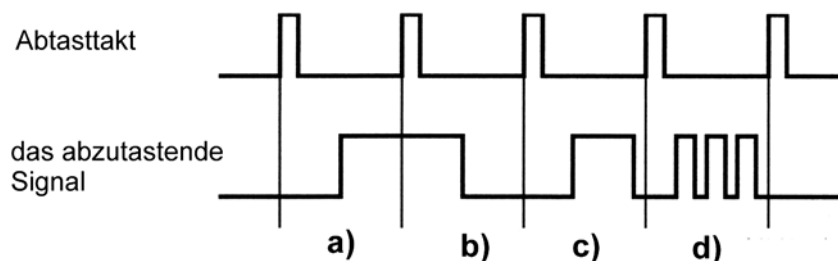


Abb. 9 Glitcherkennung

- a), b) kein Glitch, da jeweils nur eine Signalflanke in der Abtastperiode,
- c), d) Glitch, da jeweils mehr als eine Signalflanke in der Abtastperiode. In beiden Fällen zeigt ein entsprechender Logikanalysator einen Glitch an. Es ist aber nicht erkennbar, wie er wirklich aussieht. Die Glitches c) und d) können in der DUT ganz verschiedene Wirkungen haben!

Zeiten, in denen nichts aufgezeichnet wird

Der Ablaufspeicher kann meist nur einen Ausschnitt aus dem Signalverlauf aufnehmen. In den Zeiten, in denen er zwecks Darstellung ausgelesen wird, kann gar nichts aufgezeichnet werden. Das sind ähnliche Verhältnisse wie beim Messen mit dem digitalen Speicheroszilloskop.

5. Triggerung

Die Triggerung von Logikanalysatoren beruht im wesentlichen auf dem Vergleich der erfaßten Prüf-signale mit eingestellten Mustern (Trigger Patterns). Daneben ist üblicherweise eine externe Triggerung möglich.

Praxisbeispiel

Um die Nutzung der verschiedenen Triggervorkehrungen zu veranschaulichen, beziehen wir uns auf ein einfaches Beispiel (Abb. 10). Es handelt sich um eine übliche Zusammenschaltung von Mikrocontroller und Speichersubsystem. Wir nehmen an, daß alle 20 Signale des dargestellten Mikrocontroller-Interfaces an Dateneingänge des Logikanalysators angeschlossen sind.

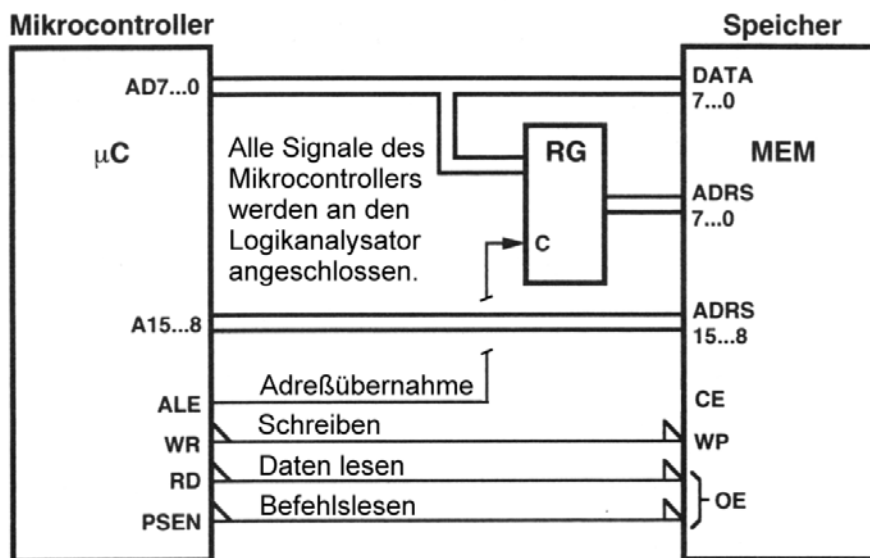


Abb. 10 Praxisbeispiel zum Erklären der Triggervorkehrungen

Bitmustervergleich

Das Trigger-Ereignis wird ausgelöst, wenn Signalbelegung und Triggerbedingung in allen Bitpositionen übereinstimmen. Trigger-Bitmuster werden üblicherweise in dreiwertiger Form vorgegeben. Vorgabe = 0: das Signal muß mit logisch 0 (Low) belegt sein, Vorgabe = 1: das Signal muß mit logisch 1 (High) belegt sein, Vorgabe = X: das Signal wird nicht berücksichtigt; es ist gleichgültig, ob es 0 oder 1 ist (Don't-Care-Bedingung).

Beispiel 1:

Uns interessiert, ob auf die Adresse 2015H Zugriffe (gleich welcher Art) stattfinden. Vergleichsbedingungen: (1) AD7...0 und A15...8 gemäß Adreßangabe, (2) ALE = 1, (3) WR, RD, PSEN = X (um sie vom Vergleich auszuschließen).

Beispiel 2:

Wir wollen sehen, was passiert, wenn ein Programmstück von Adresse 3F01H an abgearbeitet wird. Mit den angeschlossenen 20 Signalen geht das aber nicht so ohne weiteres (zum Ausweg siehe Seite 13).

Eine Alternative: Wir schließen die Adreßeingänge des Speichers an den Analysator an (Vergleichsbedingung: Adreßangabe, ALE = 0, PSEN = 0 (kennzeichnet Zugriff zum Programmspeicher), WR und RD = 1).

Mehrere Bitmuster

Höherentwickelte Analysatoren ermöglichen es, mehrere Bitmuster gleichzeitig vorzugeben. Diese werden parallel mit den Signalbelegungen verglichen. Die Vergleichsaussagen werden disjunktiv zusammengefaßt. Damit entspricht die Triggerbedingung einer Booleschen Gleichung in disjunktiver Normalform (ODER-Verknüpfung von UND-Verknüpfungen).

Beispiel 3:

Wir haben den Verdacht, daß bei Zugriffen auf Adresse 023FAH oder auf Adresse 23FFH etwas schiefeht. An einem entsprechend ausgerüsteten Analysator könnten wir beide Bedingungen einstellen.

Bereichsvorgaben

Oft sind Abläufe von Interesse, die sich in bestimmten Bereichen von Adressen, Datenwerten, Steuer-codes usw. abspielen. Was einzelne Signale angeht, so kann man sie durch eine Don't-Care-Vorgabe (X) von der Auswertung ausschließen (vgl. Beispiel 2); die Triggerbedingung wäre dann im gesamten möglichen Bereich der Signalbelegungen erfüllbar. Sinngemäß lassen sich Bereiche erfassen, indem man beispielsweise Daten- oder Adreßbits mit X kennzeichnet (Tabelle 3).

Signalbelegungen	Triggerbeispiele				
	1001	100X	10XX	1X01	1X0X
0000					
0001					
0010					
0011					
0100					
0101					
0110					
0111					
1000		3	3		3
1001	3	3	3	3	3
1010			3		
1011			3		
1100					3
1101				3	3
1110					
1111					

Tabelle 3 Bereichsangaben durch Don't Cares (X-Bedingungen)

Wir betrachten hier ein Beispiel mit 4 Signalen. Diese können insgesamt 16 Belegungen annehmen (linke Spalte). Die folgenden Spalten enthalten Beispiele für Triggerbedingungen, wobei jeweils durch Pfeile angegeben ist, welche Belegungen die jeweilige Bedingung erfüllen.

Allgemein:

n Signale können 2^n mögliche Belegungen annehmen. Eine Triggerbedingung, die kein einziges X enthält, wählt davon eine einzige Belegung aus; sie ist nur dann erfüllt, wenn genau diese Belegung eintrifft. Ein X führt dazu, daß die Triggerbedingung für 2 Belegungen erfüllt ist, bei 2 X-Angaben ist sie für 4 Belegungen erfüllt usw., also allgemein bei k Don't Cares für 2^k Belegungen. Betrachten wir die Signale als Darstellungen natürlicher Binärzahlen (z. B. als Adressen), so geben k Don't Cares in den niedrigstwertigen Stellen einen Bereich der Länge 2^k vor, der an einer integralen Grenze beginnt (das heißt, mit einem Zahlenwert, dessen k niedrigstwertigen Bits Nullen sind. Das heißt: Mit Don't Cares in den niedrigstwertigen Stellen kann man nur auf Bereiche triggern, die an integralen Grenzen beginnen und deren Länge einer Zweierpotenz entspricht.

Damit kann man schon etwas anfangen. Allerdings liegen nicht alle Bereiche, auf die wir triggern müssen, so günstig.

Beispiel 4:

Wir wollen prüfen, ob auf den Bereich 221H...229H Zugriffe erfolgen. Dies läßt sich mit Don't Cares allein nicht einstellen. Praxistip: Auf 022XH triggern, also die niedrigsten 4 Adreßbits vom Vergleich ausschließen. Dann heißt es eben, die Aufzeichnungen darauf hin zu durchmustern, ob die betreffenden Werte enthalten sind. (Manchmal ist das eine Komfort-Funktion des Analysators.) Wir können aber auch Pech haben: beispielsweise dann, wenn die Adressen 22AH, 22BH usw. von einer Programmschleife belegt sind, die immer wieder aufgerufen wird. Es treten dann so viele Trigger-Ereignisse auf (und stopfen den Ablaufspeicher regelrecht zu), daß die wenigen Zugriffe, auf die es uns ankommt, oft gar nicht in der Aufzeichnung zu finden sind.

Manche Geräte haben deshalb besondere Triggermöglichkeiten, wobei Signale als Binärzahlen interpretiert werden. Man kann dann auf Bereiche „von...bis“ oder „außerhalb von...bis“ triggern.

Zeitbewertung

Einfachere Analysatoren lösen Trigger-Ereignisse stets dann aus, wenn die zusammengefaßte Triggerbedingung im Setup-Hold-Intervall der jeweiligen Taktflanke erfüllt ist. Hierbei kann es aber vorkommen, daß wir auf Ereignisse triggern, die uns gar nicht interessieren, oder welche verpassen, auf die wir eigentlich gewartet haben. Deshalb ermöglichen höher entwickelte Geräte eine zeitliche Bewertung der Trigger-Ereignisse (Tabelle 4).

Zeitbewertung	zusätzliche Angaben	Bedingung für Auslösung des Trigger-Ereignisses
Breite (Width)	Zeitangabe, Wahl „innerhalb/außerhalb“	die Zeit, in der die Triggerbedingung erfüllt ist, ist entweder kürzer oder länger als die Zeitvorgabe
Zeitbereich (Range)	Zeitintervall (2 Zeitangaben), Wahl „innerhalb/außerhalb“	die Zeit, in der die Triggerbedingung erfüllt ist, liegt entweder in den Grenzen des Intervalls oder sie ist kürzer bzw. länger als das Intervall
Zeitüberschreitung (Time Out)	Zeitangabe	die Triggerbedingung tritt innerhalb der Zeitvorgabe nicht auf

Tabelle 4 Zeitbewertung von Triggerbedingungen (Auswahl)

Sequenzbewertung (Mehrebenen-Triggerung)

Hiermit können zeitlich aufeinanderfolgende Trigger-Ereignisse erfaßt werden.

Beispiel 5:

Wir wollen triggern, wenn auf Adresse 1AA0H ein Schreibzugriff mit der Datenbelegung 4FH erfolgt. Mit den bisher beschriebenen Triggervorkehrungen gelingt das nicht, da Adresse und Daten nacheinander über dieselben Leitungen übertragen werden. Hierfür wäre eine Triggervorkehrung nötig, die folgendes leistet:

1. warte auf Adresse = 1AA0H UND AEN = 1,
2. ist dies erfüllt, triggere auf AD7...0 = 4FH UND WR = 0.

Auch unser *Beispiel 2* läßt sich so im Rahmen der an den Analysator angeschlossenen 20 Signale erledigen:

1. warte auf Adresse = 3F01H UND AEN = 1,
2. ist dies erfüllt, triggere auf PSEN = 0.

Manche Logikanalysatoren haben eine entsprechende Mehrebenen-Triggerung. Das heißt, nach Auftreten der ersten Bedingung wird gewartet, ob die zweite eintrifft usw. Hierbei können wiederum Zeitbedingungen ähnlich Tabelle 4 vorgegeben werden. *Hinweis:* Wichtig ist die Zeit, die das Gerät braucht, um von einer Ebene zur nächsten überzugehen. Nur wirklich hochwertige Typen schaffen das – bei maximaler Abtastrate – innerhalb eines Taktzyklus.

Reagieren auf Trigger-Ereignisse

Die einfachsten Reaktionen bestehen darin, die Signalerfassung zu starten oder anzuhalten. Höher entwickelte Analysatoren kann man hingegen regelrecht programmieren. Hierfür gibt es an höhere Programmiersprachen angelehnte Sprachkonstrukte, die sich über entsprechende Bildschirm-Menüs ausfüllen lassen, beispielsweise „Trigger on... Followed by...“ (für die oben skizzierte einfache Sequenzbewertung) oder „Trigger if... else do...“ usw.

Zustandstriggerung (State Triggering)

Das Triggern bei der Zustandsanalyse unterscheidet sich grundsätzlich nicht von dem der Zeitanalyse. Es wird lediglich die Auswertung der Triggerbedingung auf den Takt bezogen, der von der DUT geliefert wird. Des weiteren sind bei manchen Geräten in dieser Betriebsart bestimmte Triggerfunktionen nicht nutzbar (z. B. die Zeitbewertung und die Sequenzbewertung).

Takt-Qualifier bei der Zustandsanalyse

Bei einfachen Analysatoren sind Qualifier Signale, die mit dem Takt (aus der DUT) kombinatorisch verknüpft werden (beispielsweise müssen bei einer UND-Verknüpfung alle Qualifier = 1 sein, damit der Takt wirken kann). Bessere Geräte bieten flexiblere Verknüpfungsmöglichkeiten unter Einschluß von Don't Cares. Durch Einbeziehen von Qualifiern kann man oft den Ablaufspeicher besser ausnutzen (weil wir nicht bei *jedem* Takt eine Signalbelegung speichern).

Hinweise:

1. Erfüllte Triggerbedingung, spezifizierte Taktflanke und erfüllte Qualifier müssen zeitlich zusammentreffen – sonst passiert gar nichts.
2. Takt und Qualifier werden kombinatorisch verknüpft; es gibt also zwischen diesen Signalen keine Setup- und Haltezeiten. *Aber:* Der „eigentliche“ Takt, der im Analysator die Daten abtastet – auf dessen Flanke also die Setup- und Haltezeiten zu beziehen sind –, entsteht erst, wenn beides (Taktsignal & erfüllte Qualifier-Bedingung) wirksam ist. Zudem darf es nicht vorkommen, daß der Qualifier den Takt abschneidet und dabei Impulse entstehen, die kürzer sind als die Mindest-Taktimpulsbreite (Abb. 11).

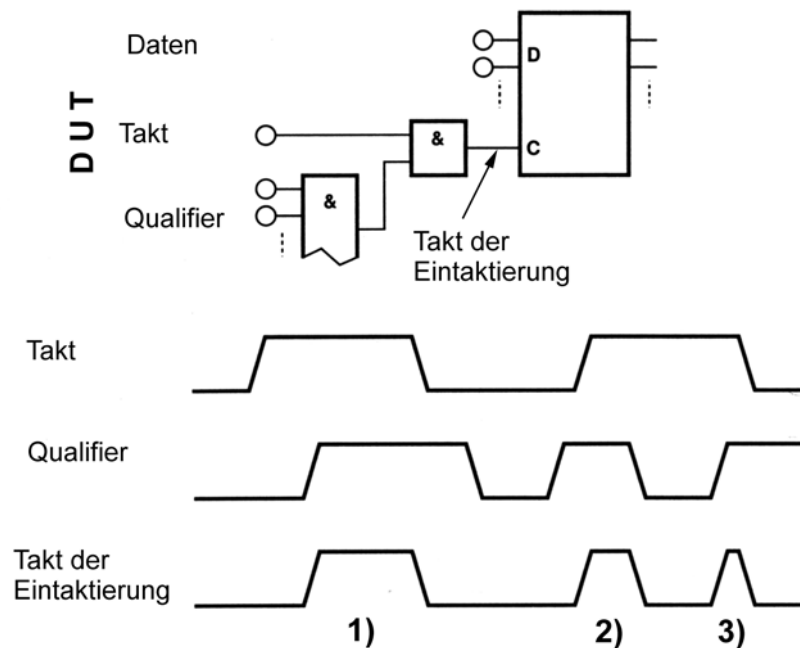


Abb. 11 Takt und Qualifier bei der Zustandsanalyse (oben: Prinzipschaltung, unten: was nicht vorkommen sollte)

- 1) Qualifier kommt später als Taktflanke; somit wird verspätet eintaktiert. Es kann sein, daß Daten-signale schon umgeschaltet haben bzw. während des Setup-Hold-Intervalls der Eintaktierung schalten. Hinweis: Viele moderne Schaltungen sind im Grunde flankengesteuerte State Machines. Diese haben die Eigenheit, daß unmittelbar nach der Taktflanke nahezu alle Signale sofort schalten. Man würde hier also nicht die „eingeschwungene“ Signalbelegung vor, sondern das „wilde“ Schalten nach dem Zustandswechsel aufzeichnen, vielleicht sogar schon die eingeschwungenen Belegungen vor dem nächsten Zustandswechsel (was weniger auffällt, da die Impulsdiagramme meist schöner – also nicht auf den ersten Blick verdächtig – aussehen).
- 2) Takt und Qualifier überlappen sich sowenig, daß nur ein kurzer Eintaktierungs-Impuls entstehen kann. Gefahr: es wird entweder gar nichts oder Unsinn aufgezeichnet (manche Geräte können, mit unzulänglichen Impulsen angesteuert, wie ein PC abstürzen).
- 3) durch das Schalten des Qualifier bei aktivem Takt entstehen weitere Impulse der Eintaktierung. An sich sollte erst wieder bei einer erneuten Low-High-Flanke des Taktes eintaktiert werden. Enthält der Analysator keine entsprechenden Vorkehrungen, messen wir Mist.

Speichertiefe und Triggervorkehrungen

Als Alternative zu hochentwickelten Triggervorkehrungen liegt es nahe, den Ablaufspeicher großzügig auszubauen. Der Grundgedanke: wir speichern zunächst einmal alles, was überhaupt zu erfassen ist, und durchsuchen dann die gesamte Aufzeichnung mittels Software nach dem, was uns interessiert. Eine solche Lösung ist auch nicht billig, hat aber einen Vorteil: Wir können sowohl Durchmusterungs- als auch Triggervorkehrungen nur dann erfolgreich nutzen, wenn wir wissen, was wir wollen. In der Praxis spielt sich das üblicherweise so ab, daß man zunächst eine bestimmte Hypothese hat, die man entweder beweisen oder widerlegen will. Demgemäß formuliert man seine Such- oder Triggerbedingungen. Nun kann man eine einmal getätigte Aufzeichnung immer wieder durchsuchen. Hingegen bleibt bei einer erfolglosen Aufzeichnung infolge einer „danebengegangenen“ Trigger-Hypothese nichts anderes übrig, als den Prüfablauf zu wiederholen – und dabei ist es oft Glückssache, daß genau dieselbe Fehlersituationen nochmals auftritt.