

Analizador lógico

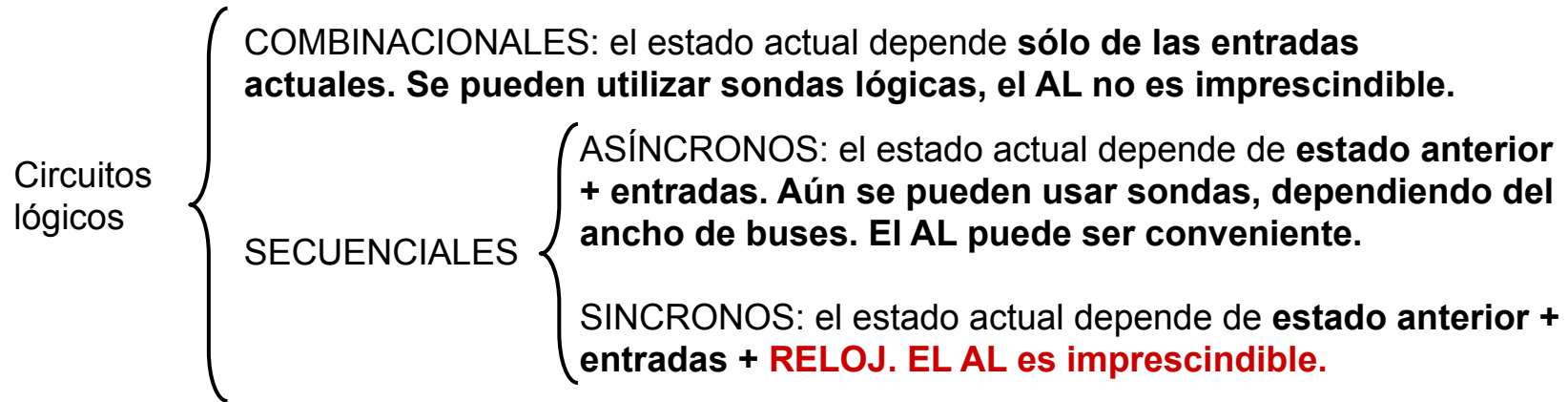
Carlos A. Zerbini

Cátedra Medidas Electrónicas 2

Universidad Tecnológica Nacional - Facultad Regional Córdoba

Año lectivo 2020

Generalidades

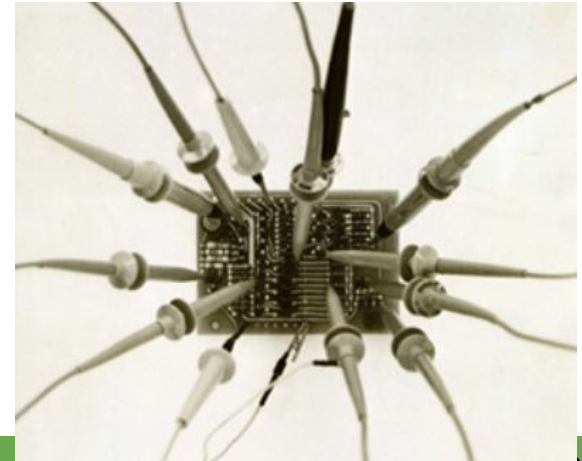


Se requiere una herramienta para analizar **circuitos secuenciales síncronos (y por extensión los demás), abarcando desde el software hasta el hardware.**

Puntas lógicas (repaso)



- especialmente adecuadas para circuitos combinacionales o secuenciales **asíncronos**
- Suelen detectar niveles de un par de familias lógicas
- En general, soportan **sólo un canal digital**



Generalidades

El primer analizador lógico apareció en 1967, como un proyecto personal del Ing. Gary Gordon en HP. Seis años antes, Gary aplicaba analizadores lógicos como interno en el lab de osciloscopios, para resolver problemas de deriva cuando se cambiaba la base de tiempo de los osciloscopios digitales.

Los analizadores lógicos **evolucionaron a partir de los OADs**, y casi simultáneamente con los primeros microprocesadores disponibles. Los primeros analizadores lógicos se centraban en operaciones muy parecidas a los osciloscopios para debuggear y testear **hardware**. Luego evolucionaron, abarcando el monitoreo de la **actividad de las señales de un microprocesador y en el debugging de software**.

Un analizador lógico sólo mide señales digitales, no analógicas! Puede capturar **muchas señales digitales simultáneamente** y mostrar sus - muchas veces complejas - relaciones en el tiempo. **Algunos analizadores cruzan ligeramente el límite y se adentran en el dominio del OAD, detectando glitches y violaciones de tiempos de setup/hold**. Pero estas funciones se centran en debuggear señales elusivas e intermitentes, y hasta algunos permiten relacionar código fuente con problemas de hardware específicos.

El DSO es el instrumento ideal cuando se necesita:

- Medir características analógicas de **no más de cuatro señales a la vez**.
- Medir características **eminentemente analógicas**, tales como tiempos de subida/bajada, fase, potencia, corriente, amplitud, y relaciones entre flancos.
- Determinar la **estabilidad de una señal (jitter)**.
- Medir **márgenes de tiempo**, tales como tiempos de setup/hold y retardos de propagación.
- Determinar **estados transitorios**, tales como señales deformadas o “redondeadas”, glitches, o estados meta-estables (estados estables por cortos periodos)
- Observar **pequeñas excursiones de tensión** e una señal.
- Trabajar con **alta precisión en la medición de intervalos de tiempo**.

Generalmente, el OAD es el instrumento a utilizar cuando se necesita alta resolución vertical o de tensión.

Un analizador lógico es el instrumento ideal cuando se necesita:

- Debuggear y verificar la operación de **sistemas digitales**.
- Seguir, observar y correlacionar **muchas señales digitales simultáneamente**.
- Detectar y analizar **relaciones en tiempo** entre muchas señales digitales.
- Realizar un **seguimiento de la evolución de operaciones digitales**.
- Observar las señales de un sistema **de la misma manera en que las ve el hardware**.
- Disparar en base a un patrón de “altos” y “bajos” sobre varias líneas digitales, y observar el resultado.



HP 5000A (1973)



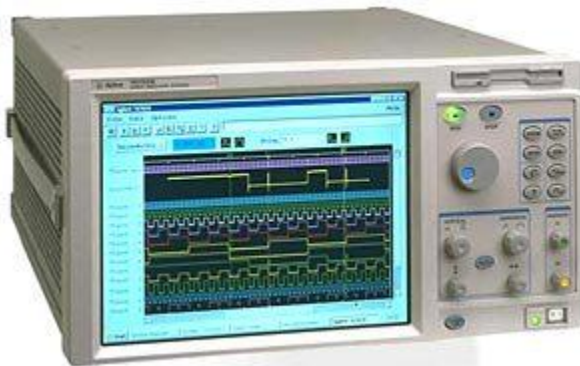
Serie 1600 (80s)



HP 1601L (1974)



http://hpmemoryproject.org/wb_pages/wall_b_page_12.htm
<https://testequipment.center/HP/Logic-Analyzers/Logic-Analyzers>
<https://www.tek.com/logic-analyzer/tla6400>
<https://hackaday.com/2009/03/06/tools-saleae-logic-logic-analyzer/>



Product specifications of Agilent 16702A (From

Alternative 16900 Series

Instrument Type Mainframe

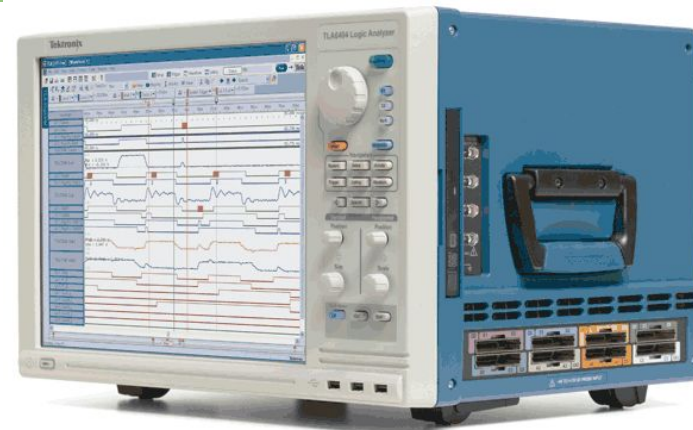
Max. Channels Number 340

Max. State Speed 100 MHz

Max. Timing Speed 4 GHz

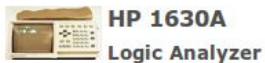
Status Discontinued by Manufacturer

- 5 slots
- Multi-window interface
- Setup Assistant
- Max State Clock 100 MHz
- On-chip emulation link-debugge
- Multiple time-correlated views
- Real-time view
- 800x600 LCD display
- Accepts 5 measurement modules and 2 emulation modules



Characteristic	TLA6401	TLA6402	TLA6403	TLA6404
Channels	34	68	102	136
High-speed timing	25 GHz (40 ps) with 128 Kb record length			
Maximum timing sample rate (Half/Full channel)	3.2 GHz / 1.6 GHz			
Maximum State clock rate	333 MHz (standard) 667 MHz (with Option 1T)			
Maximum State data rate	667 Mb/s (standard) 1333 Mb/s (with Option 1T)			
Maximum record length	2 Mb (standard) 4 Mb (Option 1S) 8 Mb (Option 2S) 16 Mb (Option 3S) 32 Mb (Option 4S) 64 Mb (Option 5S)			

Generalidades



HP 1630A
Logic Analyzer

Algunos modelos clásicos



HP 1630D
43 Channel 100 MHz Logic Analyzer



HP 1630G
65 Channel 100 MHz Logic Analyzer



HP 1631D
200 Ms/s Logic Analyzer / 50 MHz D



HP 1650A
Logic Analyzer



HP 1650B
80 Channel Logic Analyzer



HP 1651A
32 Channel Logic Analyzer



HP 1651B
32 Channel Logic Analyzer



LA1034 LogicPort Specifications and Characteristics

Sampled channels: 34

Timing mode sample rate: 1KHz to 500MHz (uses internal clock)

State mode sample rate: 0 to 200MHz (clock provided by circuit under test)

Sample buffer: 34 x 2048 samples

Maximum sample compression: 2⁴³ to 1 (sample rates to 200MHz)

Trigger sequencer: 250MHz max, 4ns minimum pulse width

Trigger capability:

Edges - Rising, Falling, Either (multiple channels, any combination)

Patterns - True, False, Entered, Exited (across any / all channels)

Bus Value - Equal, Not Equal, Less Than, Greater Than, In Range, Not in Range

Occurrence Count - Range of 1 to 1,048,576 Consecutive or Cumulative

Duration - Equal, Less Than, Greater Than, Within Range (samples or time)

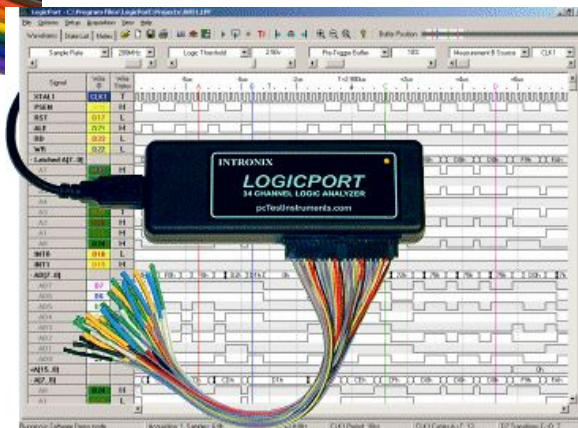
Input impedance: 200K Ohms, paralleled by <5pF

Frequency counter range: 300MHz with 10Hz resolution, 4 channels

Threshold range: adjustable +6 to -6 volts with 50mv resolution

Las de bajo costo:

- SALEAE
- Logicport
- Redpitaya, AD Discovery,...
- **FPGAs: Signaltaps, Chipscope**



Interpreted Protocols (included):

Asynchronous Serial (RS232, RS422 and RS485 to 25Mbps)

Synchronous Serial (SPI, PS2 and proprietary formats)

Inter-Integrated-Circuit (I2C)

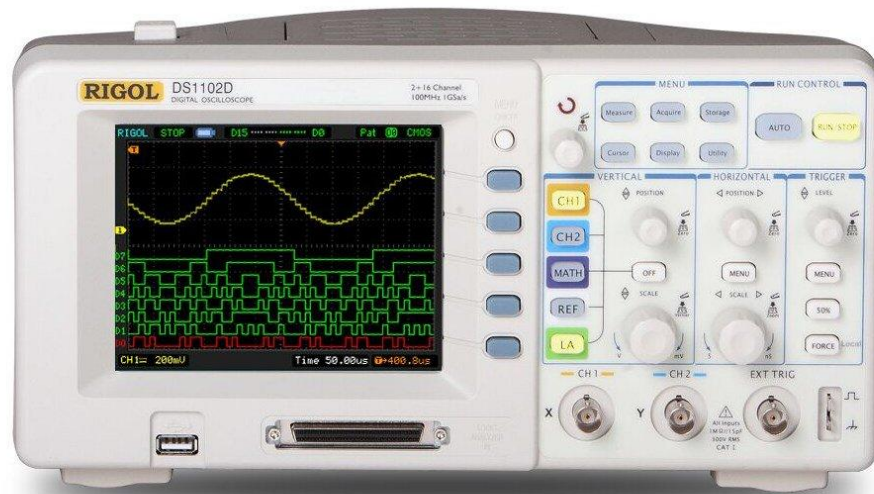
CAN 2.0A and 2.0B

Synchronous Parallel (Quad/multi-IO SPI and proprietary formats)

1-Wire (Dallas/Maxim devices/iButton)

ISO7816-3 (Smart Card and SIM devices)

Generalidades



Osciloscopio de señal mixta (MSO)

Rigol DS1102D

Catalog #: DS1102D

★★★★★ 4.5 of 5

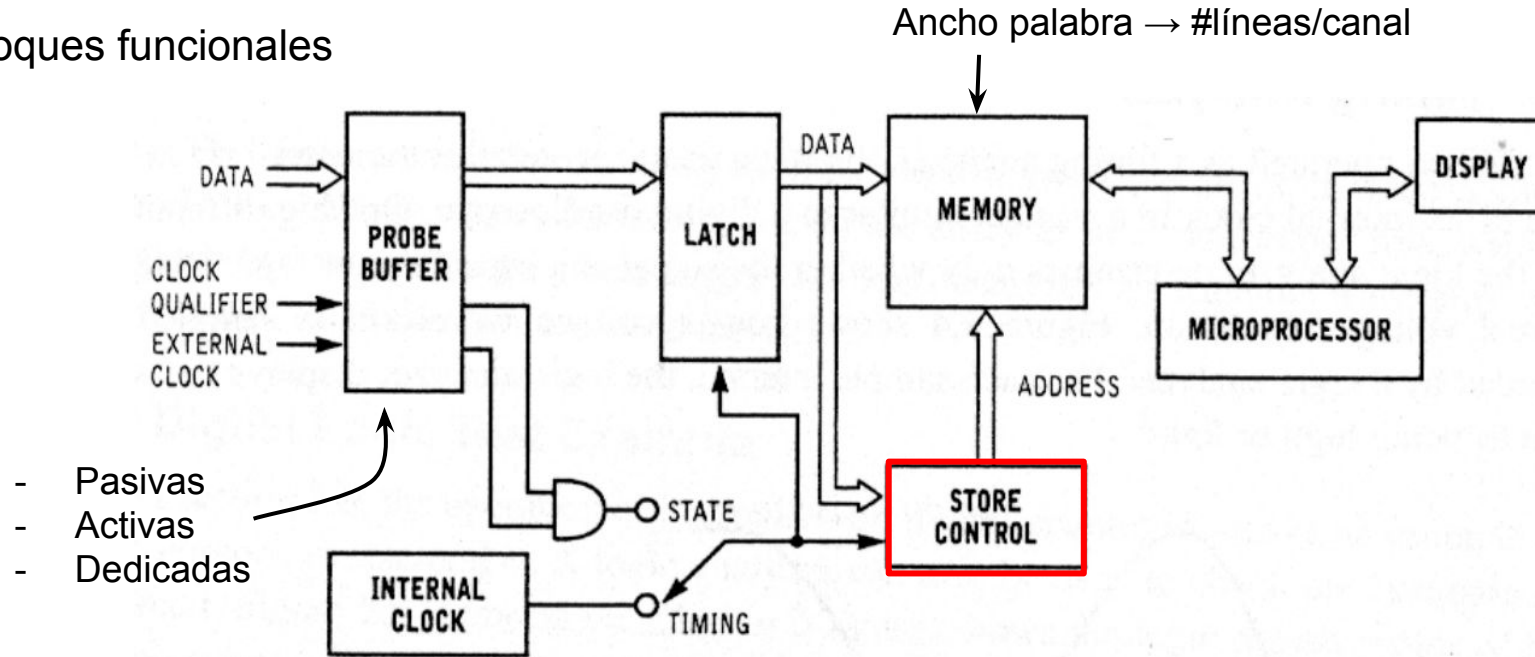
[Read Reviews \(6\)](#) | [Write Reviews](#)

100 MHz Mixed Signal Oscilloscope with 2 analog channels and 16 digital channels plus USB storage and connectivity and 1 GSa/sec sampling

Features

- 1 GSa/s maximum real-time sample rate, 25 GSa/s maximum equivalent-time sample rate
- 1Mpts long Memory Depth
- 100 MHz, 50 MHz Bandwidths
- Ultra compact design: 303mm x 154mm x 133mm (WxHxD)
- 64 k TFT color LCD, bright and vivid waveform display
- Advanced trigger modes: Edge, Video, Pulse Width, Slope, Alternate
- Adjustable bigger sensitivity: Effectively filter noise from bigger signal and avoid false trigger
- 20 automatic measurements

Bloques funcionales



- 1) **Sincronismo:** utiliza **clock y qualifier** para generar **pulsos de escritura calificados**
- 2) **Disparo/comparadores:** compara el dato en el latch con los registros de trigger cargados por el usuario. Cuando ambos coinciden, se envía una señal a los **secuenciadores y retardadores**.
- 3) **Secuenciadores/retardadores:** ante una señal de disparo, ejecuta el **modo de disparo** configurado.

Medios para mejor aprovechamiento de la memoria del AL

Debido a la alta velocidad requerida, la memoria de un AL es limitada (por ej. 34 canales x 2048 bits en el LA1034). A fin de aprovecharla para capturar los eventos de interés, se pueden utilizar las siguientes configuraciones generales:

- 1) **Qualifier (calificador)** → circuito **combinacional**, se usa una señal como **máscara** de otra señal. Permite eliminar partes del flujo de programa que no son de interés. **Se utiliza principalmente para enmascarar ciclos de clock que no son de interés durante análisis de estados (lo veremos luego)**
- 2) **Muestreo transicional (“compresión”)** → utiliza un **detector de transiciones** que le indica al DMA **si debe o no almacenar al dato actual**.
 - Caso especial → **detección de glitches**
- 3) **Modos de disparo** → más elaborados que en un OAD, existen varios **modos de disparo**, diferentes según el tipo de análisis a realizar (temporal o de estados). Estos modos se verán luego.

Modos de uso: conceptos

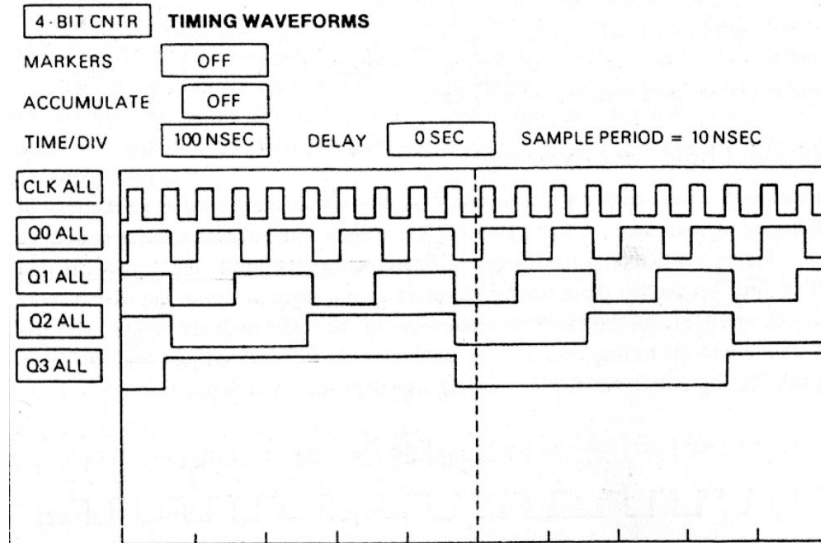
1) Análisis temporal (o modo asíncrono)

- Utiliza un **reloj interno**, comúnmente a frecuencia mayor que la del circuito medido
- Las **líneas** se capturan en sincronismo con un clock interno, **asíncrono al circuito medido**
- La visualización es **similar a un osciloscopio aunque sólo muestra dos niveles!**
- Se utiliza para depurar problemas **a nivel de hardware.**

2) Análisis de estado (o modo síncrono)

- Utiliza un **reloj externo**, proveniente del sistema a medir.
- Los **datos** se capturan en sincronismo con el circuito ensayado.
- La visualización es bajo la forma de **palabras digitales** (binario, comp. 2, hexa, decimal, etc.)
- Se utiliza para depurar problemas **a nivel de software.**

Modos de uso: comparación

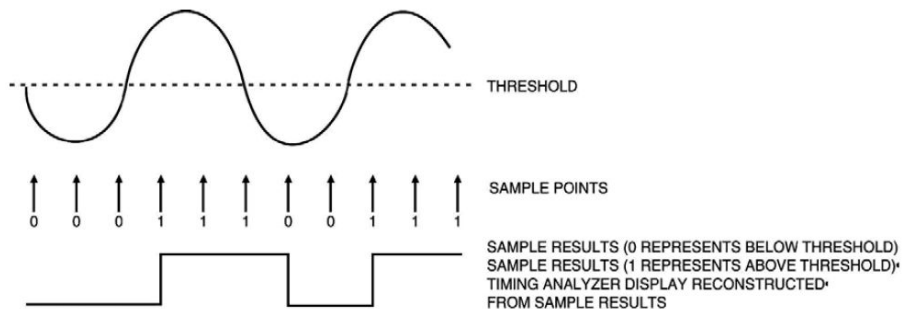


Análisis temporal

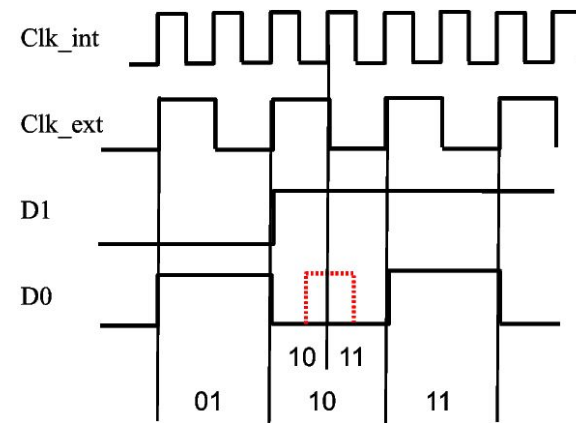
4-BIT CNTR		STATE LISTING	
MARKERS		<input type="button" value="OFF"/>	
LABEL	>	<input type="button" value="Q"/>	
BASE	>	<input type="button" value="BIN"/>	
+0000		0000	
+0001		0001	
+0002		0010	
+0003		0011	
+0004		0100	
+0005		0101	
+0006		0110	
+0007		0111	
+0008		1000	
+0009		1001	
+0010		1010	
+0011		1011	
+0012		1100	
+0013		1101	
+0014		1110	

Análisis de estado

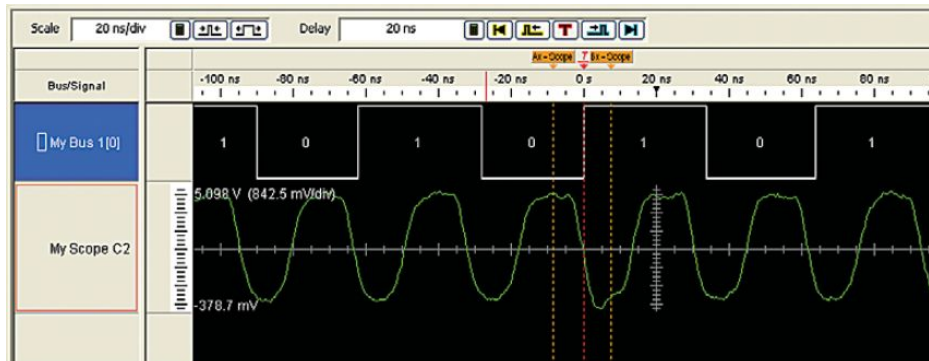
Análisis temporal



Ejemplo de muestreo en análisis temporal

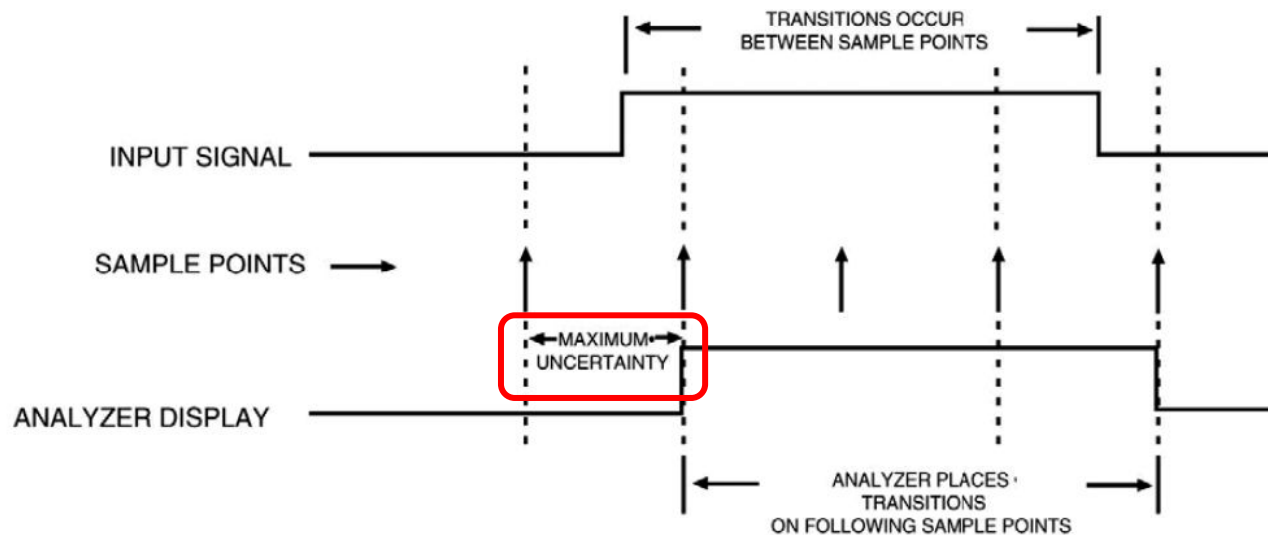


Ejemplo de detección de una transición indeseada



Una señal de clock en el análisis temporal, y en un osciloscopio

Análisis temporal



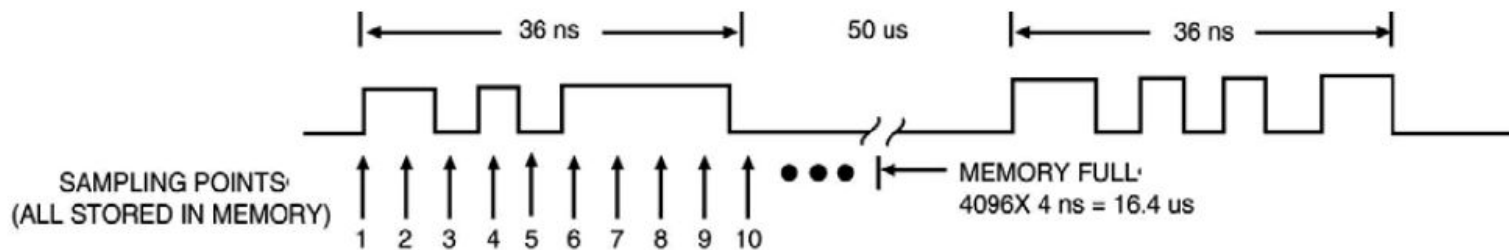
La incertidumbre en análisis temporal depende de:

- **Velocidad** de los circuitos del AL
- **Tamaño o profundidad** de memoria

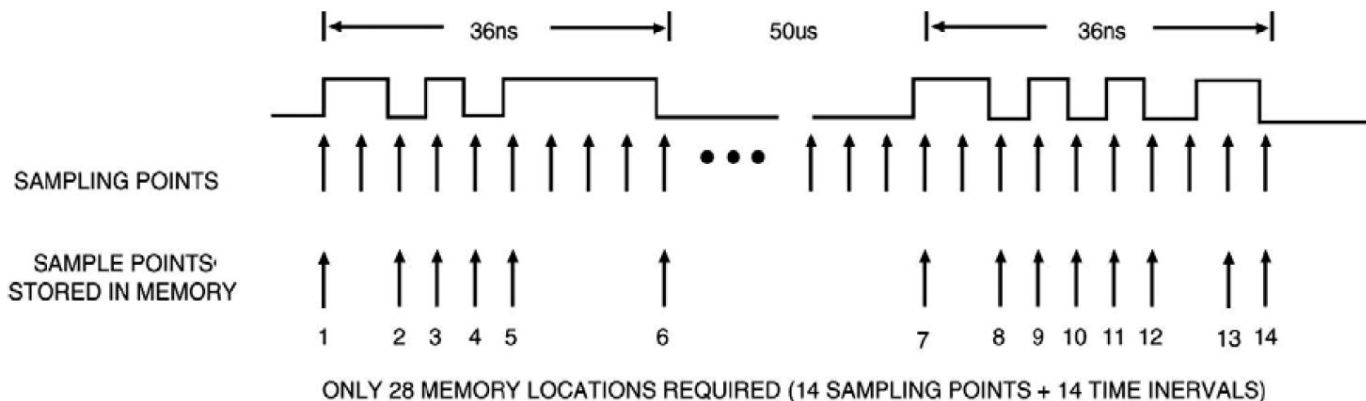
$$\text{incertidumbre máxima} = t_{\text{clk}} = 1 / f_{\text{CLK}}$$

$$T_{\text{MEDIDO}} = n \times t_{\text{clk}} = n / f_{\text{CLK}}$$

Análisis temporal

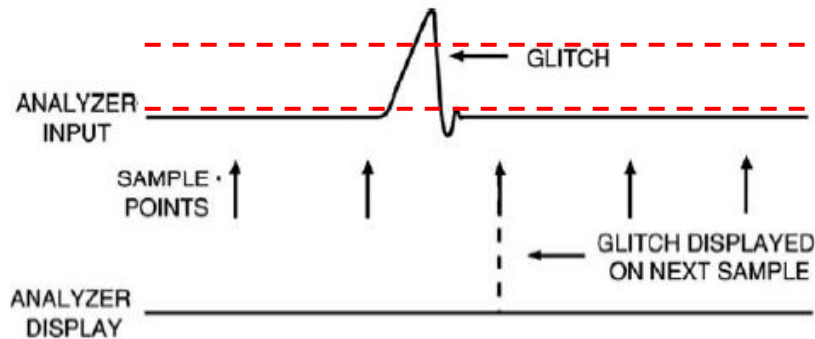


Muestreo normal



Muestreo transicional ("compresión")

Análisis temporal



Transiciones rápidas de tensión (glitches)

Existen dos tipos, uno de ellos detectable por el AL y el otro no:

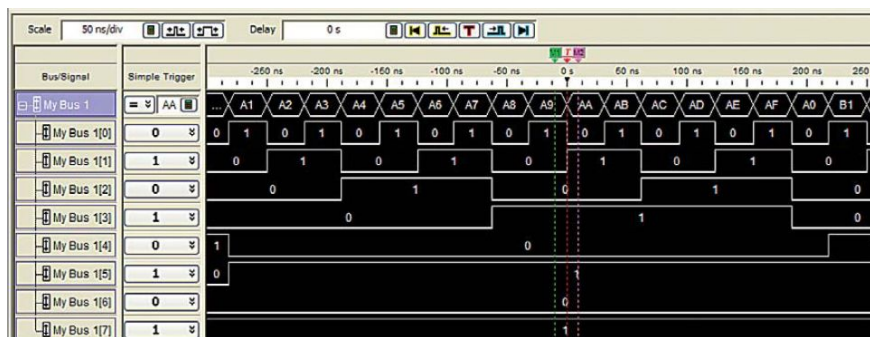
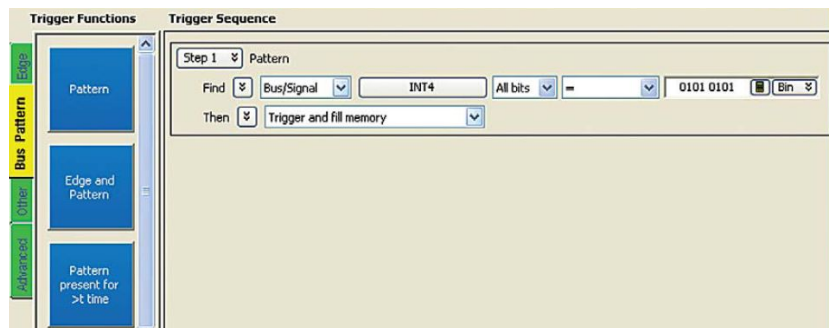
- Transiciones que cruzan los umbrales **más de una vez entre muestras**. Se detectan mediante el modo “glitch detect” **hasta un ancho mínimo**.
- Transiciones que **no cruzan los umbrales lógicos, pero salen de los niveles máximos o mínimos tolerados**. **NO los detecta el EL**, sino que se utiliza un osciloscopio de memoria profunda (DMO)

El AL sólo ve lo que cruza ambos umbrales, es ciego a todo lo demás. El AL muestra estos cruces como una transición ideal entre dos estados, y no permite observar la FORMA DE ONDA REAL.

Análisis temporal

Eventos de disparo (trigger)

- **Pattern trigger:** genera el disparo cuando se da un cierto patrón en las líneas observadas



- **Edge trigger:** genera el disparo cuando las líneas observadas transicionan $0 \rightarrow 1$ ó $1 \rightarrow 0$

Análisis de estados

Step 1 ▾ Advanced If/Then

▾ If ▾ Bus/Signal ▾ ADDR All bits ▾ = ▾ FFF0 3187 ▾ Hex ▾

occurs ▾ 1 ▾ - ▾ + ▾ eventually ▾

Then ▾ Trigger and fill memory ▾

▾ with ▾ Default Storage ▾

Configuración de trigger

Sample Number	ADDR	DATA	STAT
	= ▾ FFF0 34D8 ▾	= ▾ XXXX XXXX ▾	= ▾ XX XXXX ▾
-8	0000 41B0	0241 23D7	03 23D7
-7	0000 41B1	1F41 23D7	0B 23D7
-6	0000 41B2	FB41 23D7	13 23D7
-5	0000 41B3	5241 23D7	0B 23D7
-1	FFF0 3187	5541 03E7	0B 03E7
3	FFF0 34DB	A641 03E7	0B 03E7
7	FFF0 34DF	7841 03E7	0B 03E7
11	FFF0 34E3	E841 03E7	0B 03E7
15	FFF0 34E7	2941 03E7	0B 03E7
19	FFF0 6A0F	F441 03E7	0B 03E7

Presentación de secuencia de estados

Sample Number	Software Address	MPC821/860 PowerQUICC Inverse Assembler
-29	FFF0 3190	addi r11,r11,0x0001
-25	FFF0 3194	addis r12,r0,0x0000
-21	FFF0 3198	stw r11,0x41b0(r12)
-20	0000 41B0	mem write 0x00
-19	0000 41B1	mem write 0x00
-18	0000 41B2	mem write 0x0b
-17	0000 41B3	mem write 0x6c
-13	FFF0 319C	addis r12,r0,0x0000
-9	FFF0 31A0	lwz r3,0x41b0(r12)
-8	0000 41B0	mem read 0x00
-7	0000 41B1	mem read 0x00
-6	0000 41B2	mem read 0x0b
-5	0000 41B3	mem read 0x6c
-1	FFF0 31A4	bl update_display
3	FFF0 31B4	mispr r0,d8
7	FFF0 31B8	or r11,r1,r1
11	FFF0 31BC	stwu r1,0xffe8(r1)
15	FFF0 31C0	bl .text+4A08
19	FFF0 6A08	stw r28,0xffff0(r11)
23	FFF0 6A0C	stw r29,0xffff4(r11)
27	FFF0 6A10	stw r30,0xffff8(r11)

Visualización mediante
des-ensamblador dedicado

Análisis de estados


Modos de disparo (trigger)

Si nosotros no configuramos una secuencia de disparo, el AL comienza a almacenar sin control desde que se “arma”.

A diferencia del análisis temporal, en este caso estamos interesados en **capturar ciertas secciones del programa**. Por ejemplo, podríamos querer capturar una situación de error que se da SÓLO cuando se lee la dirección 0x0001 LUEGO de la dirección 0x0000. **Para eso disponemos de los modos de disparo, implementados mediante la combinación de comparadores, secuenciadores y retardadores.**

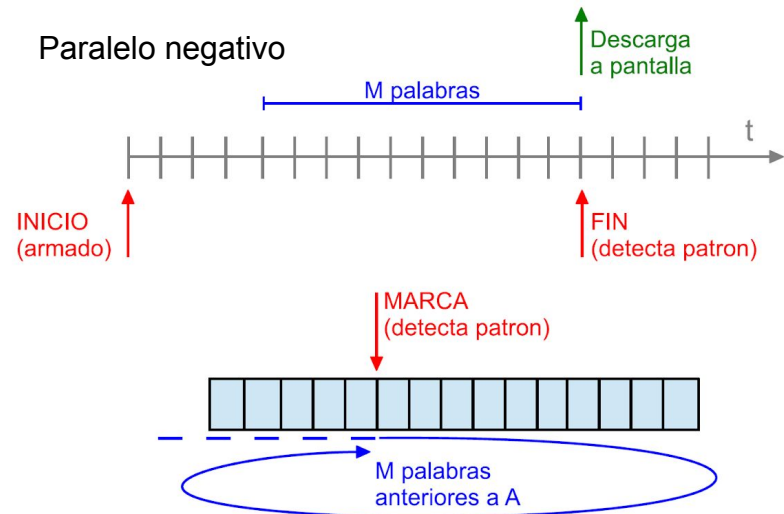
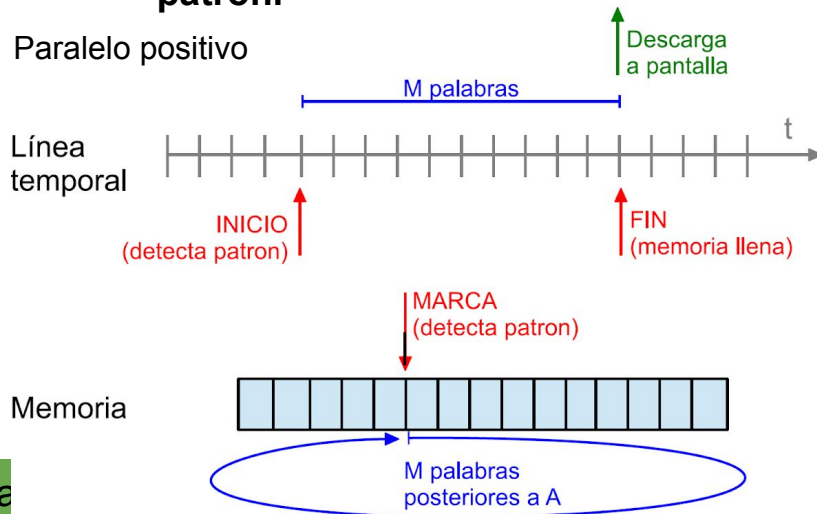
En el analizador de estados, las líneas SE AGRUPAN en UNIDADES LÓGICAS A, B, etc. definidas por el usuario. Sobre ellas, el usuario define **eventos de disparo**, y combinando varios de estos patrones SECUENCIALMENTE se obtienen los **modos de disparo**. **Se pueden realizar combinaciones**, por ej. Unir dos puertos, disparar un puerto con otro, etc.

Eventos de disparo comunes

- 
- Igual al patrón / distinto / mayor / menor / \geq / \leq
 - Dentro de cierto rango / fuera de él
 - Igual al patrón durante cierto #ciclos **consecutivos**
 - Igual al patrón un #ciclos no necesariamente consecutivos
 - Combinación de patrones mediante AND/OR y derivados

Análisis de estados

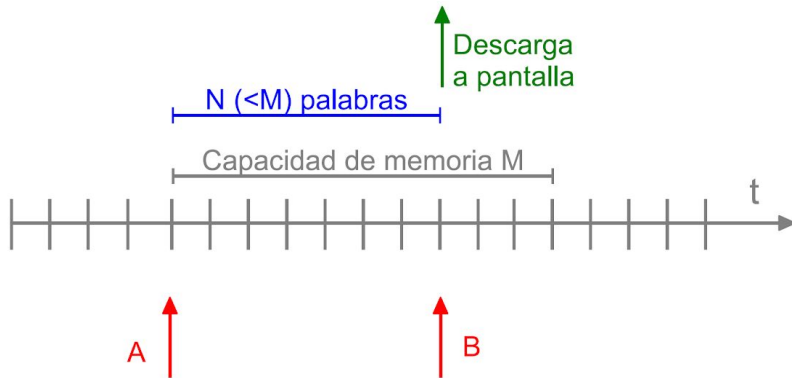
- 1) **Disparo PARALELO:** las palabras de disparo se agrupan obteniendo un patrón digital utilizada para **comenzar o finalizar** la adquisición de datos. Esto define **dos modos diferentes**:
- a) **Paralelo positivo:** la adquisición **comienza** con el patrón y **finaliza** cuando se llena la memoria de tamaño **M**. La memoria contiene las **M palabras posteriores al patrón**.
 - b) **Paralelo negativo:** la adquisición comienza con el “armado” del instrumento, y finaliza cuando se encuentra al patrón. La memoria contiene las **últimas M palabras previas a producirse el patrón**.



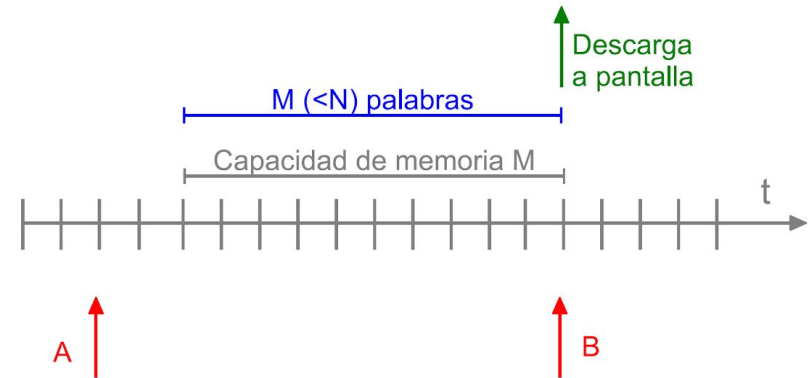
Análisis de estados

2) **Disparo SERIE:** se utilizan **dos palabras de disparo para inicio y fin de la captura respectivamente. Se pueden dar dos casos:**

- a) Distancia entre palabras $N < \text{capacidad de memoria } M \rightarrow$ se capturan N palabras
- b) Distancia entre palabras $N > \text{capacidad de memoria } M \rightarrow$ se capturan **las últimas M palabras**



Distancia $B-A < M$



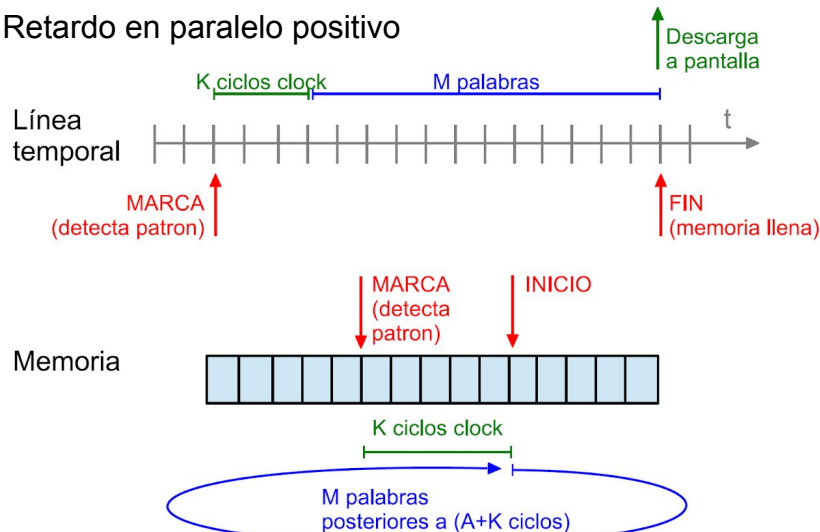
Distancia $B-A > M$

Análisis de estados

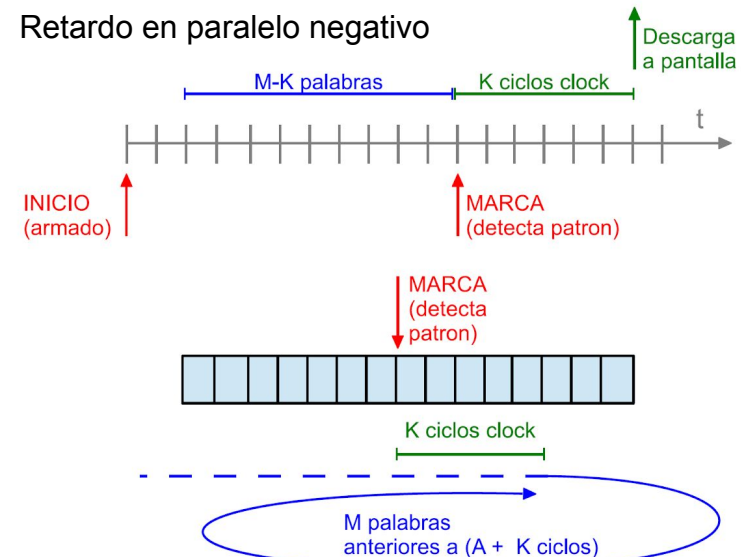
3) **Retardos:** implementa una **demora sobre la condición de trigger**. Para disparo paralelo, esto sería equivalente a **demorar la ocurrencia de la palabra de trigger**. La demora puede ser:

- En ciclos de reloj:** es una demora fija, sin importar lo que ocurra con los datos digitales
- En ocurrencias de palabras de disparo:** el retardo consiste en un **determinado número de ocurrencias de la palabra de disparo**. Es decir que no es un tiempo fijo, sino que depende de la frecuencia de ocurrencia de la palabra de disparo

Retardo en paralelo positivo



Retardo en paralelo negativo



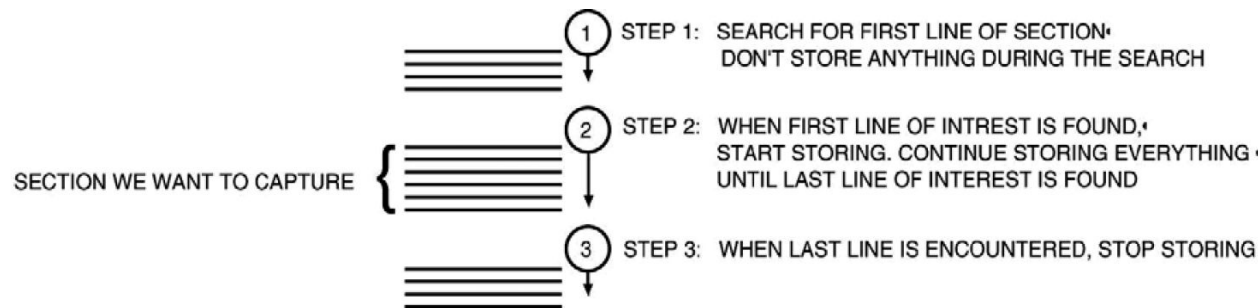
Análisis de estados

- **Niveles de secuencia y almacenamiento selectivo:** este método, definido a nivel de software, permite capturar datos sólo durante **ciertas sub-rutinas de un bucle** (por ejemplo un cálculo repetitivo que falla). A su vez, este método ayuda a **utilizar la memoria eficientemente para la zona de interés (almacenamiento selectivo)**
- **Funciones pre-definidas de trigger:** Librería de secuencias de eventos de uso común, pre-definidas. Por ejemplo “Find Nth occurrence of an edge” y “Find event ‘n’ times”.
- **Interpretadores de protocolos:** Combinación pre-definida de todas las técnicas anteriores y orientada a testear un protocolo particular. Por ej, I²C, SPI, CAN, etc..

```

1 find xxxx
else on xxxx go to level x 2
  then find xxxx
else on xxxx go to level x 3
  trigger on xxxx

```



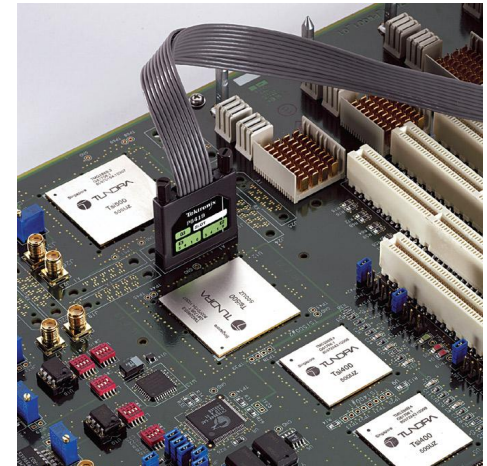
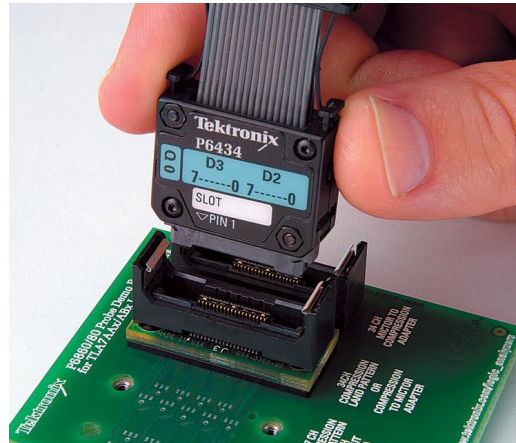
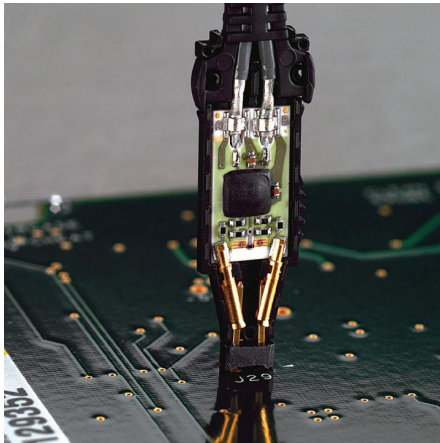
Puntas de prueba

- Son muy importantes ya que producen la **interfaz eléctrica** con el circuito a probar. Pueden ser de distintos tipos: individuales, de alta densidad (16/32/64/128 canales), de compresión, etc..
- Es muy importante la **carga** que presenta la punta al DUT, representada por una resistencia y un capacitor (por ej. el Intronix presenta 200 K // 5 pF)

Tipos de puntas

1. **Pasivas:** permiten **tomar** información pero **no inyectan estímulos**. Su objetivo es proveer **separación eléctrica, rechazo de ruido/perturbación, y asegurar correcta temporización datos/clock/calificador**
2. **Activas:** permiten **inyectar un estímulo**, por ejemplo cuando el sistema espera un acknowledge para continuar.
3. **Dedicadas:** específicas a un sistema en particular basado en cierto procesador. Incluyen toda la interfaz necesaria (address/data, control), protocolos de lectura/escritura, lenguaje de programación, etc. Se usan por ej. en **líneas de producción**

Puntas de prueba



Bibliografía

Feeling comfortable with logic analyzers, Agilent AN1337, 2006

Logic analyzer fundamentals, Tektronix Primer, 2010

The data domain transition, HP Memory Project,
[Online:] http://hpmemoryproject.org/wb_pages/wall_b_page_12.htm

V. D. Martin, *A Logic Analyzer Tutorial*, Nuts & Volts Magazine, Sept. 2007.
[Online:] http://www.nutsvolts.com/magazine/article/a_logic_analyzer_tutorial_part_1