

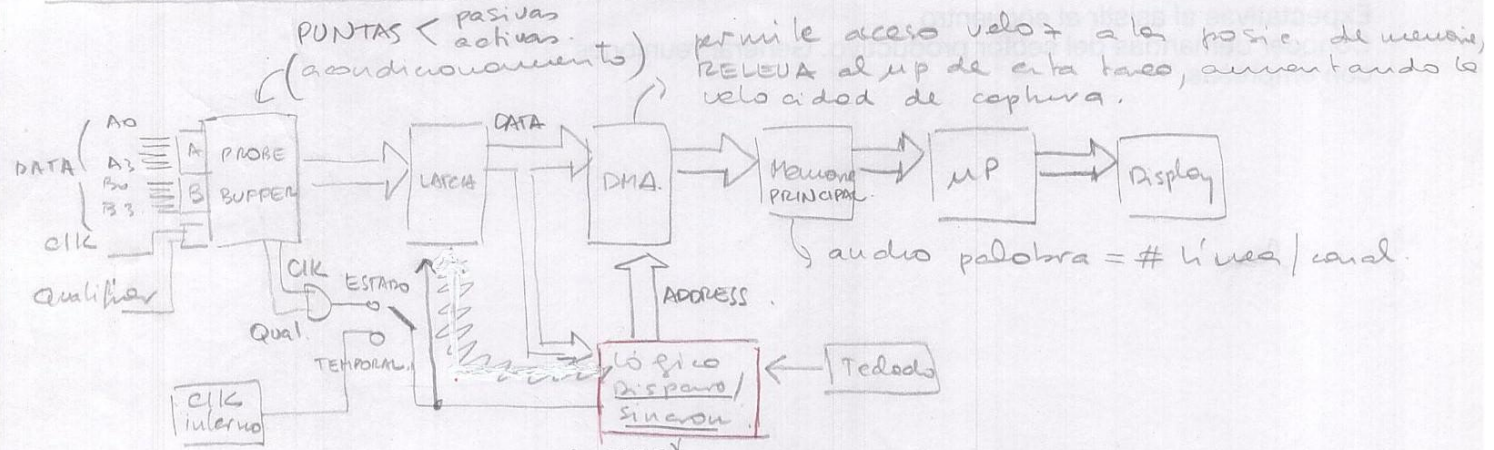
# ANALIZADOR DE ESTADOS LÓGICOS

1

- Circuitos LÓGICOS:
- COMBINACIONALES: el estado actual depende solo de las ENTRADAS ACTUALES. Se usan SONDAS pata por pata.
  - SECUENCIALES:
    - ASINCRONOS → el estado sí a COMBINAC. de ESTADO ANTERIOR + ENTRADAS. El estado ANTERIOR INTERESA.
    - SINCRONOS → depende de todo lo anterior MAS UN RELOJ..

Necesito una herramienta p/ auditar sistemas SINCRONOS, que son los MAS COMPLEJOS. Deben poder chequear DESDE EL SOFT HASTA EL HARD. Incluso si sirve p/ SINCRONOS, también puede cubrir los otros.

## Audizador de estados lógicos



"Sincronismo" → clock + QUALIFIER → LATCH

"disparo" (lógico + secuencial) → MODOS de disparo. Mem. principal.

("STORE" (arma)) genera pulsos de escritura cualificados p/ control de acceso a memoria

1) SINCRONISMO: utiliza el CLOCK y el QUALIFIER p/ generar PULSOS DE ESCRITURA CUALIFICADOS (o "PONDERADOS").



2) DISPARO / COMPARADORES: compara la latches de datos de entrada con la latch de trigger, cuyo contenido lo cargó el micro. Cuando AMBOS VALORES COINCIDEN, se envía una señal a los SECUENCIADORES X RETARDADORES.

### 3) SECUENCIADO / RETARDADORES

Procesa la SEÑAL DE DISPARO obtenida junto con los MODOS DE COMANDO y dirige la operación de la demás sub-sistema, proporcionando flexibilidad p/ implementar los MODOS DE DISPARO que veremos luego.

(Fernández)

- Señal que comanda el latch  $\rightarrow$  puede provenir de DOS FUENTES.

#### 1) ANÁLISIS DE ESTADO (operac. SINCRÓNICA) $\rightarrow$ usa RELOJ EXTERNO

La idea es CAPTURAR el DATO LÓGICO EN SINCRONISMO con el CIRCUITO QUE SE ESTÁ ENSAYANDO.

#### 2) ANÁLISIS TEMPORAL (operación ASINCRÓNICA) $\rightarrow$ se desea monitorear

las señales lógicas en una base MAS CONTINUA, INDEPENDIENTE del reloj del sistema que se ensaya  $\Rightarrow$  usa un RELOJ INTERNO de alta frecuencia, que lo CONTROLA.

- Entrada CALIFICADA DE RELOJ  $\rightarrow$  los sistemas digitales suelen

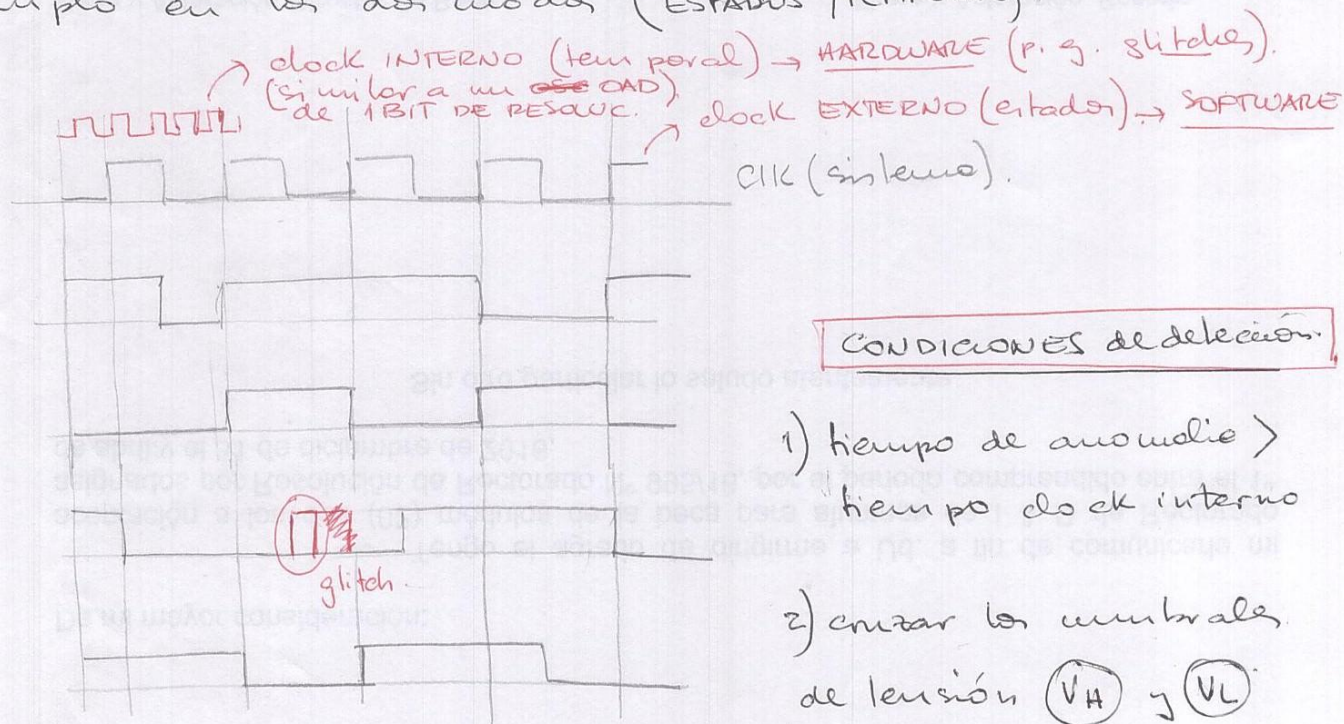
usar clocks muy veloces ( $> 50 \text{ MHz}$ ), peor aún si tenemos análisis temporal donde el clock es MAS RAPIDO q' el del sistema. Así, es fácil LLEVAR la memoria del instrumento. Uso



de la forma p/ controlar esto es el CALIFICADOR DE CLOCK, <sup>(3)</sup> que hace q' el audiotador IGNORE ~~SE~~ CIERTOS CICLOS DE RELOJ. Un ej. típico es ciclo de lectura.

- CONTROL DE ALMACENAMIENTO → puede PROGRAMARSE para ALMACENAR DATOS con CIERTOS REQUERIMIENTOS (p. ej. un PATRON LOGICO). p/ CONSERVAR MEMORIA. (esto ya sería DISPARO?)

Ejemplo en la dos modos (ESTADOS / TEMPORAL):



#### CONDICIONES de detección.

- 1) tiempo de anomalía > tiempo clock interno
- 2) cruzar los umbrales de tensión  $V_H$  y  $V_L$ .

#### RESOLUCION TEMPORAL

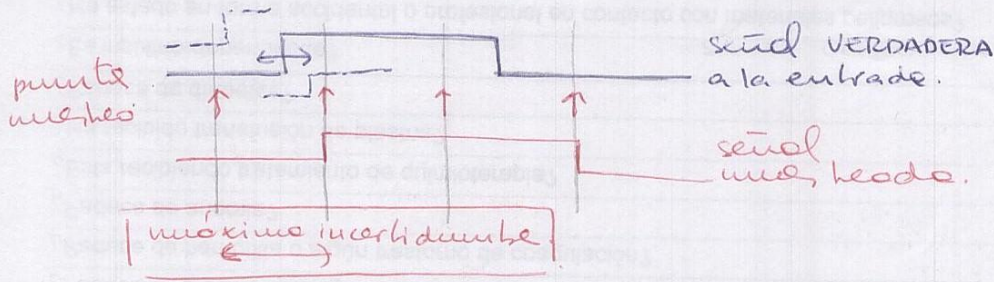
El audiotador NO PUEDE SABER lo que ocurre ENTRE FLANCOS DE CLOCK. p/ MEJORAR RESOLUCION (↓ incertidumbre), Fek debe SUBIR:

PERO hay dos limitaciones a Fek < velocidad circuitos internos.

TAMAÑO DE MEMORIA: representa un TIEMPO DE MEDICION  $T$ .

$$T_{MEDIDO} = N \times T_{CLK} = \frac{N}{f_{CLK}} \quad \left( \begin{array}{l} \text{Suponiendo} \\ \text{QUALIFIER SIEMPRE} \end{array} \right)$$

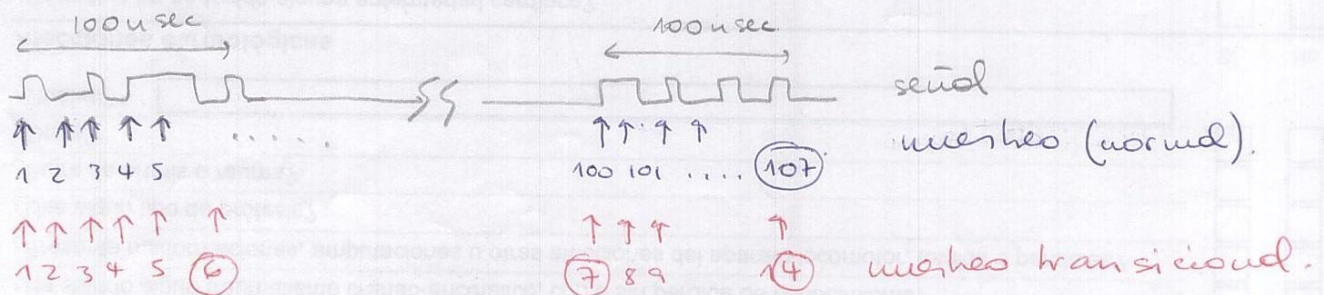




## Muestras TRANSICIONAL

- Es una herramienta para utilizar memoria MAS EFICIENTEMENTE. La data se almacena SOLO CUANDO LA SEÑAL CAMBIA DE ESTADO.
- p/ señales con ~~MUCHAS TRANS~~ TRANSICIONES FRECUENTES, el ahorro ~~ES~~ NO ES IMPORTANTE. Para señales con MUCHAS MUESTRAS ENTRE TRANSICIONES (BURSTS) el ahorro es SIGNIFICATIVO.
- Se obtienen dos  $\rightarrow$  mayor  $(T_M)$  con igual  $T_{CLK}$   
 BENEFICIOS  $\rightarrow$  igual  $(T_M)$  con MENOR  $T_{CLK}$  ( $\uparrow f_{CLK} \rightarrow$  mejor RESOLUC.).

## Ejemplo:



## Ejemplo:

$T_{mcd} = 50 \mu s$ , resoluc. deseado =  $10 \mu s$ .

$$\begin{cases} \text{muestras normal} \rightarrow \frac{50 \mu s}{10 \mu s} = 5000 = (N) \text{ posic. memoria,} \\ \text{muestras TRANSICIONAL} \rightarrow N = (14) \end{cases}$$



⊕ PERO necesito DOS POSICIONES por muestra:

⑤

↳ info de ESTADO (1 bit/cañal)

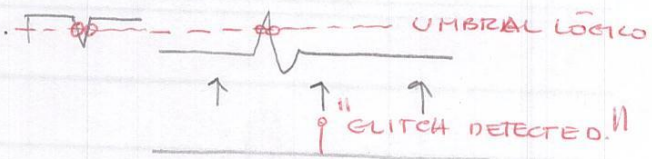
↳ info TEMPORAL (N bits), según sea T<sub>MED</sub>.

### TRANSICIONES RAPIDAS DE TENSION (glitches)

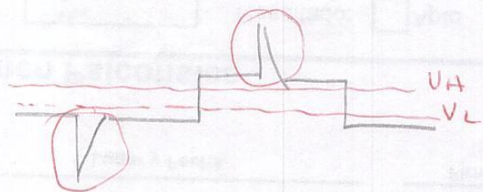
(crosstalk  
acopl. inductivo  
~~and~~  
violaciones timing.  
etc.)

Tengo 2 opciones para verlas:

1) Modo GLITCH DETECT → transiciones q' cruzan umbrales,  
(en AEL)  
MAS DE UNA VEZ ENTRE MUESTRAS. (oto!). Detecto un

MINIMO ANCHO DE GLITCH. 

2) OSCILOSCOPIOS DE MEMORIA PROFUNDA → glitches q' NO  
(DPO).  
TRANSICIONAN ESTADOS pero EXCEDEN los niveles MAXIMOS  
de la compuertas (perjudicio). (NO) los detecta con muestreo  
transicional:



Almacenan MUCHA INFORMACION de UNA LINEA (son osciloscopios).  
en forma ASINCRONA.

(A veces el sistema de selección de glitch me permite mover  
VENTANAS p/ detectar el caso ②, pero NO me muestra la FORMA).  
(niveles logicos)

### VIOLACIONES TIEMPOS SETUP/HOLD (TEK)



# EJEMPLO DE ENSAYO de un CONTADOR (apunte Rabinovich)

- Si estamos interesados en RELACIONES DE TIEMPO ENTRE SEÑALES, hacemos ANÁLISIS TEMPORAL. Si queremos por ej. verificar una FSM, hacemos ANÁLISIS DE ESTADOS.
- (NO) se muestra el CLK ya q' SIEMPRE ESTARÍA EN CERO ante el flanco.



## SECUENCIA DE CARGADO

5988

5968-5560

4632

White

Elephant

Int. inst.

- Hasta el momento yo ~~no puedo~~ puedo controlar el almacenamiento sólo en forma "combinacional", mediante el codificador. Pero fuera de ello no puedo controlar los datos que se graban ni el espacio ocupado de memoria. Para eso necesito un control más avanzado:

- ( - Lógico de DISPARO (Latches y comparadores).
- SECUENCIADORES de estado.
- Lógico de RETARDO -

En este contexto el TRIGGER NO se refiere a un NIVEL o PENDIENTE, sino a UNA PALABRA LÓGICA que introduce mediante un teclado.

### Latches de disparo y comparadores

Compara la palabra de entrada con el contenido de latches determinados x el usuario. Cuando ambas palabras coinciden, se envía una señal a los SECUENCIADORES y RETARDADORES.

### Secuenciadores de estado y lógico de retardo

PROCESA la SEÑAL de disparo anterior JUNTO con los MODOS de comando p/ controlar los otros módulos y así IMPLEMENTAR los MODOS DE DISPARO.



## Modos de disparo

8

- Si nosotros no le decimos al analizador, un det. gatillado, comienza a almacenar sin control desde el máximo ciclo de clock.
- La SECUENCIA expande el concepto de gatillado en UN PATRÓN y permite buscar OTROS EVENTOS O PATRONES en una SECUENCIA PARTIC

Por ejemplo, podríamos querer copiar una situación de error que se da SOLO cuando se lee la dirección 000001

LUEGO de la dirección 0X0000.

Eventos de gatillado como:

- IGUAL al patrón / no igual / mayor / menor /  $\geq$  /  $\leq$
- en CIERTO RANGO de valores / FUERA DEL RANGO.
- ~~en~~ patrón PRESENTE UN CIERTO TIEMPO.
- # de cuenta del patrón
- combinación con LÓGICA AND / OR.

- Recordamos que la línea de datos se AGRUPAN en "UNIDADES LÓGICAS" A, B, etc. de 16 bits x el número. El número define PATRONES DE DISPARO para estas unidades lógicas, o CANALES LÓGICOS, incluso COMBINANDOLAS (por ej. combinando DATA + ADDRESS en cierto patrón).

La COMBINACIÓN de estos patrones define los MODOS DE DISPARO.

### 1) DISPARO PARALELO

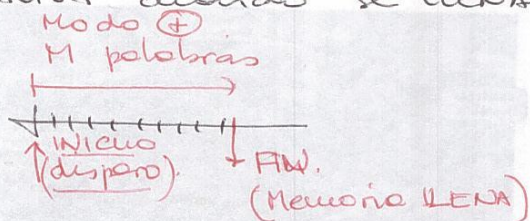
Se agrupan los patrones de disparo de cada canal en forma ASOCIATIVA. Se obtiene una ÚNICA PALABRA DIGITAL, utilizada para COMENZAR o FINALIZAR la adquisición de datos.

Esto conduce a 2 MODOS DIFERENTES:

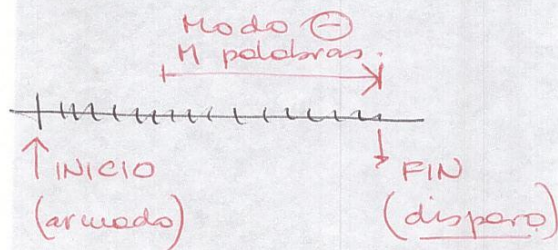
- paralelo POSITIVO
- paralelo NEGATIVO.



1a) Paralelo positivo: la adquisición COMIENZA con el patrón y FINALIZA cuando se LLENA LA MEMORIA de capacidad (M).



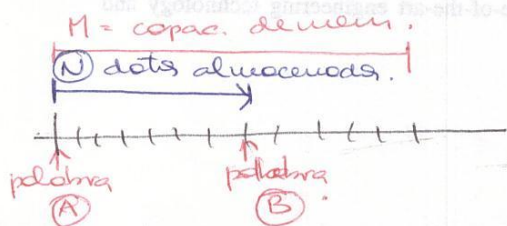
1b) Paralelo NEGATIVO: la adquisición COMIENZA con el "armado" del instrumento, y FINALIZA cuando se encuentra el PATRON. Suponiendo  $g'$  pararon) de M adquisiciones, la memoria contendrá el PATRON MAS las (M-1) palabras ANTERIORES.



## 2) Disparo SERIE

En este caso se utilizan DOS TRIGGERS, uno de inicio y otro de FIN. (Se puede disparar un canal mediante un patrón definido para otro canal). Se dan 2 casos:

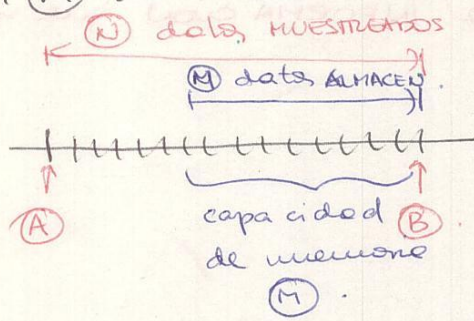
2a) distancia (N) entre palabras < cap. de memoria (M): se copiarán (N) palabras, donde  $N < M$ .



(Rabinovich) → también se pueden aplicar controles IF/THEN/ELSE a partir del gatillado, y en base a esto hacer un "gato", start/reset de un temporiz., incrementar un contador, llevar memoria, etc.



2b) dist.  $(N)$  entre palabras > capac. de memoria  $(M)$ : se almacenan los  $(M)$  datos anteriores a la palabra de FIN:

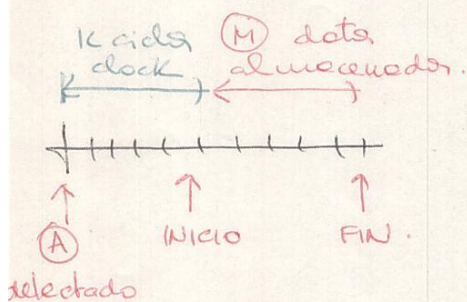


### RETARDOS

El disparo puede ser DEMORADO a partir de la ocurrencia DE LA PALABRA CONFIGURADA. El retardo puede ser en CANTIDAD DE CLOCKS o # DE ~~PALAB~~ OCURRENCIAS DE LA PALABRA DE TRIGGER.

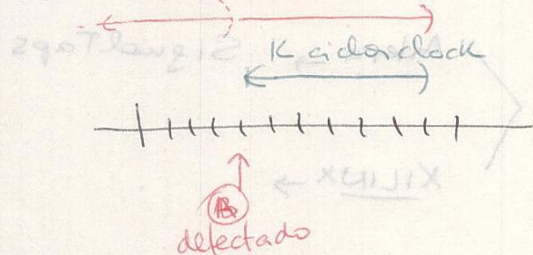
(a) Retardo de ciclos de clock: se aplica retardo de  $(K)$  ciclos (de clock INT/EXT) medidos A PARTIR DE LA PALABRA DE TRIGGER.

Es una demora en el INICIO o al FIN de la adquisición.



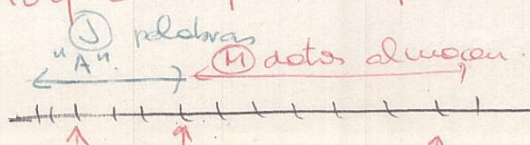
Caso PARALELO POSITIVO.

(II ⊕)  $(M)$  datos almacenados.



Caso PARALELO NEGATIVO.

b) Retardo de PALABRAS DE DISPARO: retardo de  $(J)$  ocurrencias de la palabra de trigger con respecto al INICIO o FIN de adquisición. Muy útil para depuración de PROCESOS ITERATIVOS (bucles).

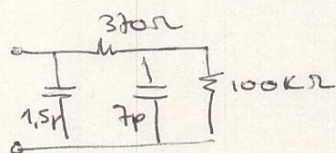




## Puntas de prueba para auditar lógico

Proveen la INTERFAZ ELECTRICA entre el circuito bajo prueba y el analizador. Por el elevado # de canales, se suelen agrupar en FICHAS de 16/32/64/128 canales.

- La punta AFECTA en cuanto medido al circuito medido, X lo que es importante la CARGA que presenta la punta. generalmente se adopta  $\sim 100K\Omega$  y  $8pF$ .



↓  
afecta los TIEMPOS DE SUBIDA observados, en especial a ALTA VELOC.

Puntas { Pasivas  
Activas  
Dedicadas.

1) Pasivas: permiten ~~se~~ TOMAR información pero NO INyectan ESTIMULOS. Su objetivo es PROVEER SEPARACION ELECTRICA, rechazo de ruido / perturbacion, y asegurar correcto TEMPORIZADO entre DATOS / SINCRONISMO / CALIFICADORES.

Permiten SELECCIONAR LOGICA que se mide (CMOS, TTL,

2) Activas: permiten TOMAR E INJECTAR información, por ejemplo si el sistema espera alguna respuesta p/ continuar.

3) Dedicadas: son mas especificas al ensayo de algun sistema en particular basados en un det. procesador. Conocen toda su interfaz, protocolos de lectura/escritura, lenguaje, etc. Se usan para...