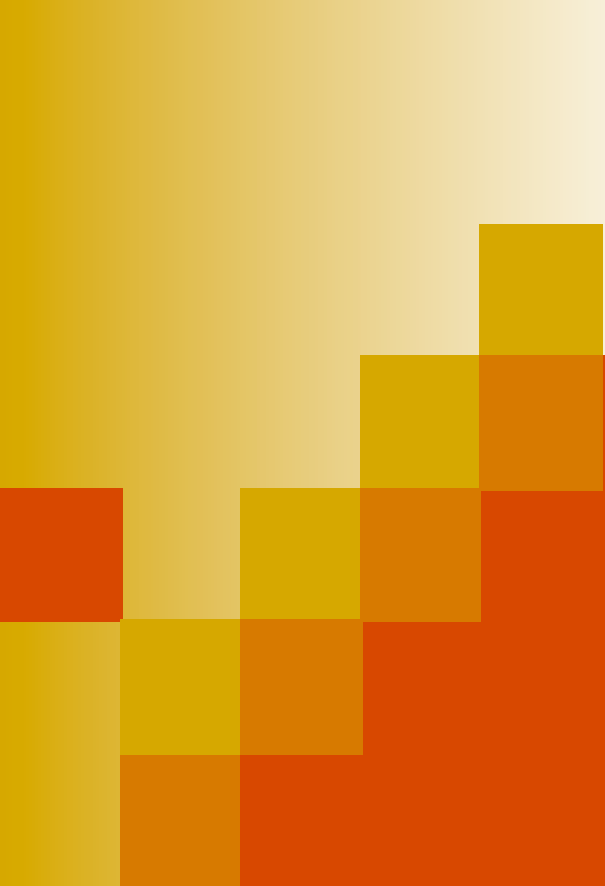




Sistemas de Entrada/Salida

Diseño de Sistemas Embebidos

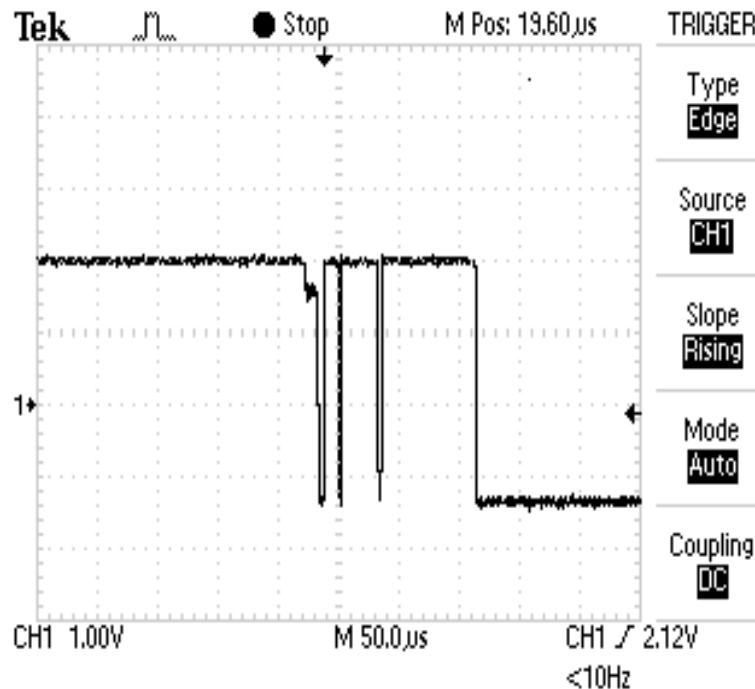
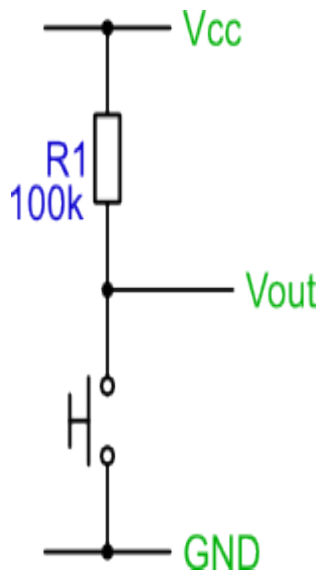
Eduardo Daniel Cohen

dcohen@herrera.unt.edu.ar

<http://www.microprocesadores.unt.edu.ar/sistemasembebidos>

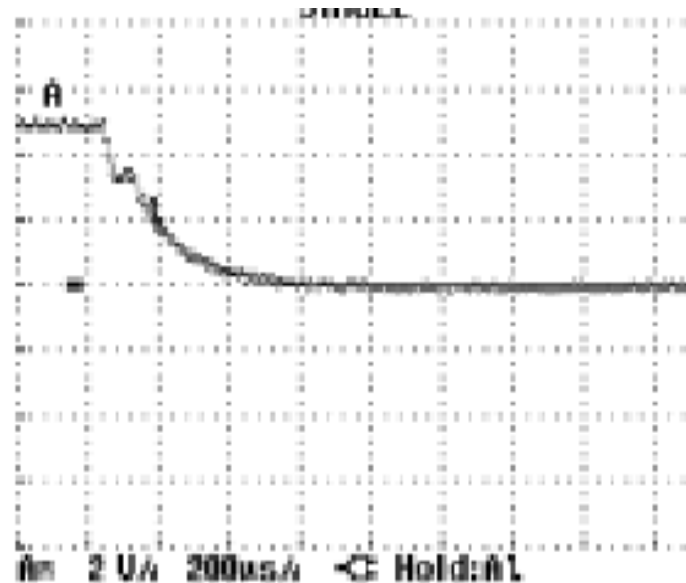
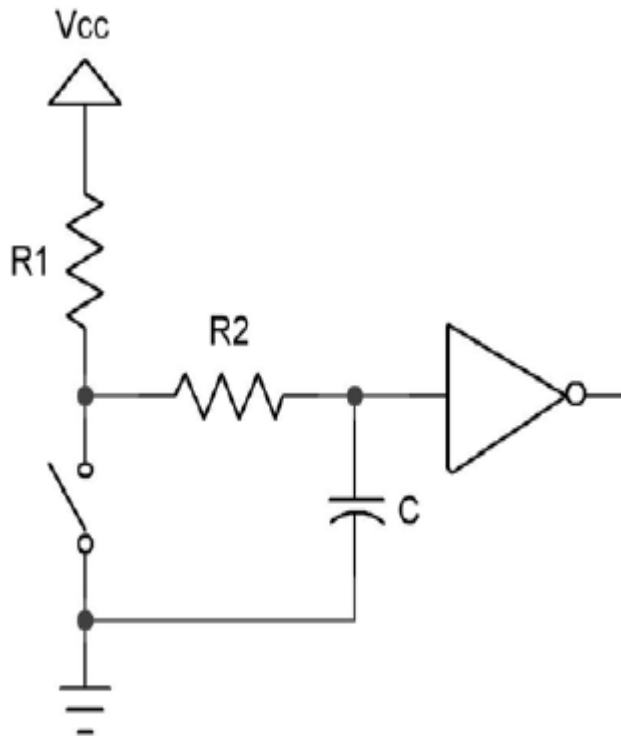
Llaves y pulsadores

- Al oprimir o soltar → Rebotes.
- Riesgo → interpretar más de una pulsación.
- ¿Soluciones? Debounce = quitar rebotes.



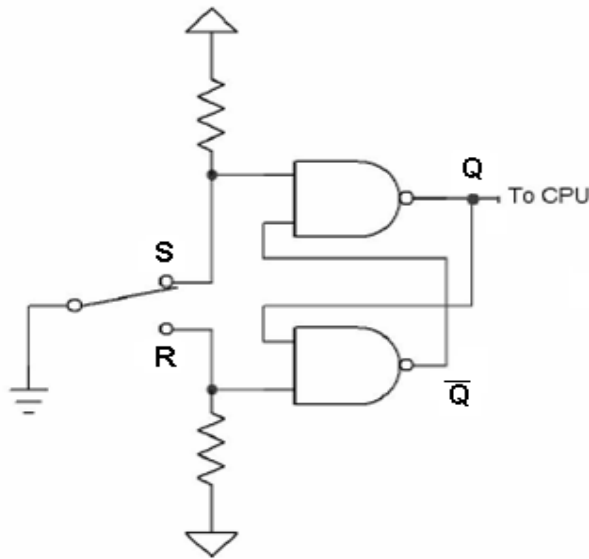
Solución Hw 1: Filtro Pasabajo

- El capacitor filtra las oscilaciones



Solucion Hw 2: Flip-Flop

- Dos compuertas NAND realimentadas.
- Se eliminan las oscilaciones.



S	R	Q_{t-1}	Q	$\neg Q$
0	0	0	1	1
0	0	1	1	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

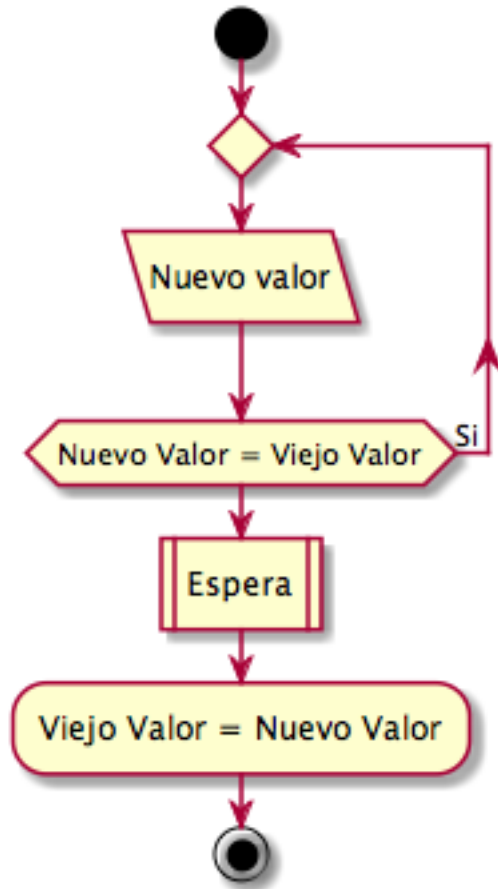
• ¿Ocurren las filas en amarillo?

Solución por Software

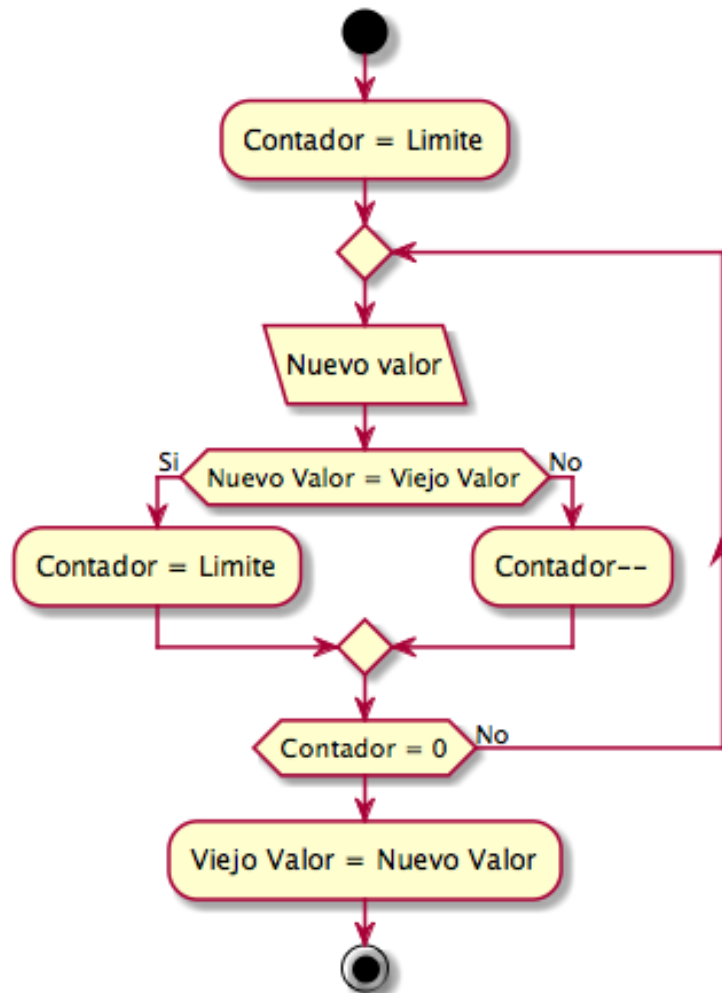
¡Idea!:

Esperar a que cese el rebote

Si viene un ruido
¿Se interpreta cómo Nv?



Solución mejorada



Teclados

- Con múltiples contactos (ej. un teclado)
 - Problema: disponibilidad de puertos.
 - Escasos en un microcontrolador.
- Solución: combinar hardware y software.

Teclado 8x8 con dos puertos

1. $i=0$
2. $S(i)=0, S(j)=1:j \neq i$
3. Si $E \neq 0xFF$
 - decode column.
 - Fin
4. $i=i+1$
5. Si $i=8 \rightarrow$ error.
6. Ir a 2.

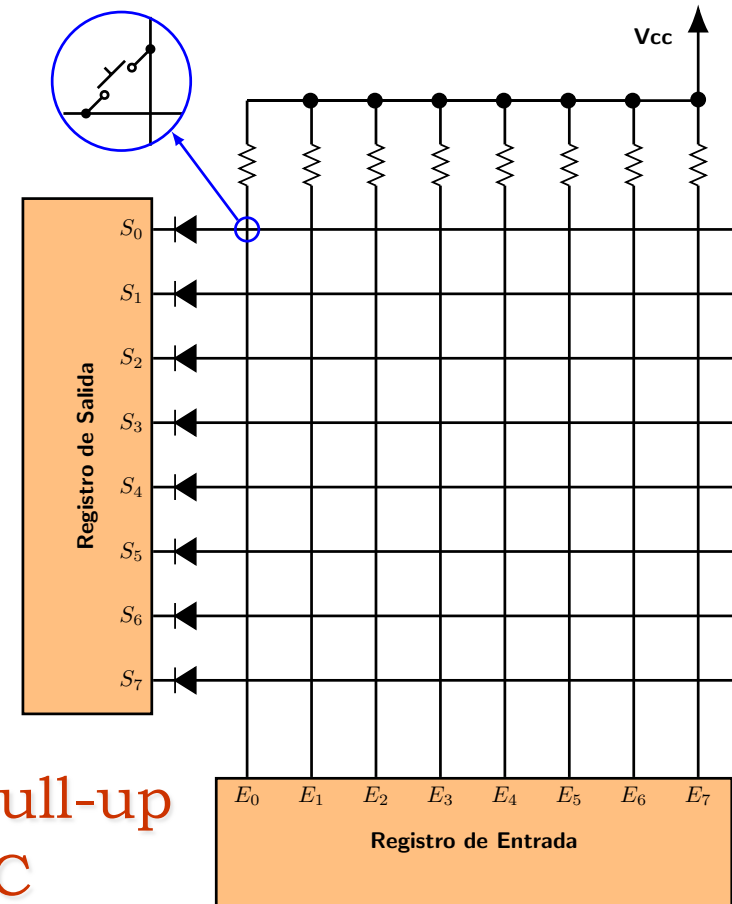
Column: primer bit=0

¿Para qué los diodos?

Configurar E como entrada con pull-up

Elimina las resistencias y VCC

Configurar S como input tri-state salvo $S(i)=0$, sin diodos



Casos Especiales.

1. ¿Qué pasa si se presionan 2 teclas a la vez?
 2. ¿Y si se mantiene presionada una tecla mucho tiempo?
- Definir **Políticas** – Se debe prever todos los casos.
 1. No tomar ninguna tecla hasta que no se oprima más que una.
 2. Obligar a soltar antes de tomar otra tecla.
 3. Tomar como repetición a un cierto ritmo.
 - **Políticas** según qué convenga más a la aplicación.

Ej. Políticas a Implementar.

1. La tecla se toma sólo una vez.
 - ❑ Para tomar otra debe soltar todas.
2. Si se oprime más de una tecla.
 - a. No tomar ninguna.
 - b. O tomar la más prioritaria.

La tecla se toma solo una vez



- Para tomar una nueva tecla se obliga a liberar todas antes
 - Se espera que se presione una tecla
 - Una vez que se determina que tecla se pulso
 - Se espera que a se suelten todas las teclas

Salto a la Acción

- Cada Tecla puede disparar una acción determinada.
- Se implementa mediante una tabla de saltos
 - Accediendo según el número de tecla ya decodificada.
- Estructura de Datos:
 1. 'TABLA' direccion inicial de la tabla de saltos
 2. NRO direccion M del número de tecla (0-63).
- Algoritmo:
 1. Destino = (TABLA + (NRO) x N)
 2. Saltar a Dirección a que apunta puntero en Destino.
 3. FIN
- Es la implementación de un SWITCH en C.

Tabla de saltos en C

```
void fa(void) {...}
```

```
void fb(void) {...}
```

```
void main(void) {  
    typedef void (*funcion) (void) ;  
  
    const funcion tabla[] =  
        { fa, fb, ...};  
  
    tabla[indice] ();  
}
```