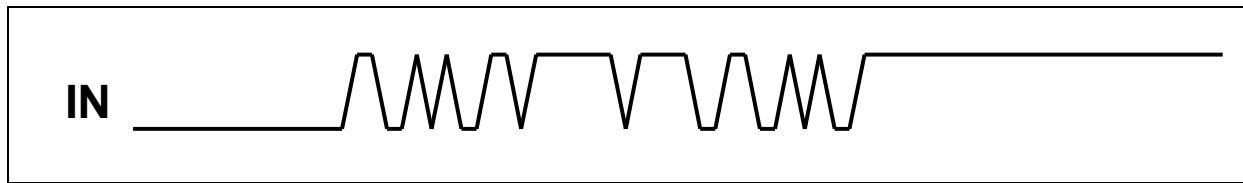
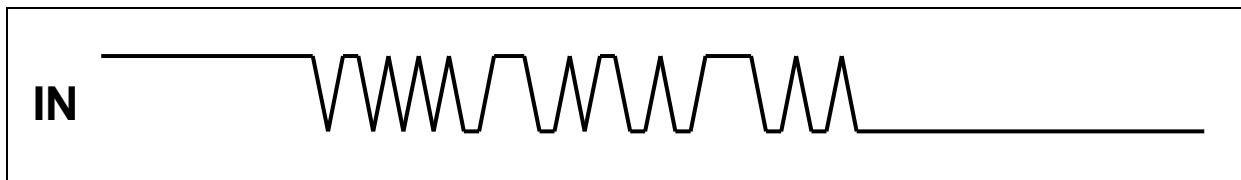


10. DEBAUNSIranJE TASTATURE

- Tasteri i prekidači ne uspostavljaju i ne raskidaju kontakt momentalno.
- Tipičan mehanički prekidač pravi nekoliko prelaza, odnosno odskakivanja (“*bounce*”) u toku nekoliko milisekundi dok ne promeni stanje, usled efekata izazvanih mehaničkim dizajnom.
- Tipični vremenski dijagrami odskakivanja pri pritiskanju tastera

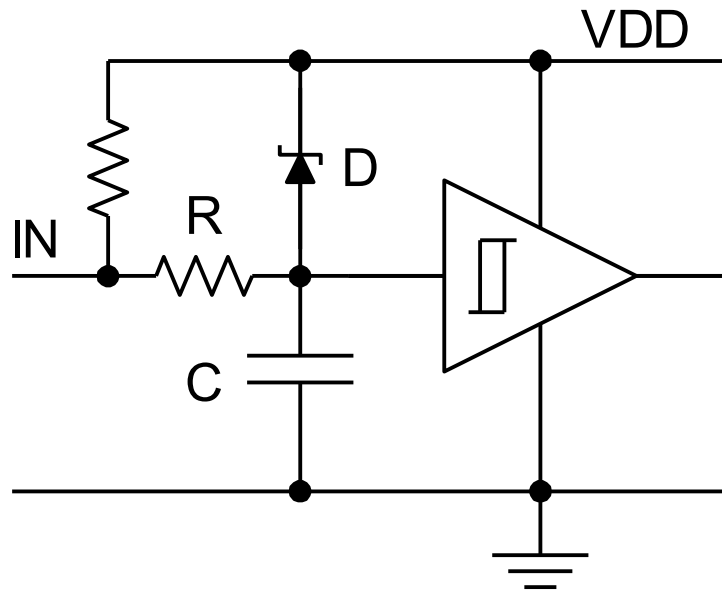


- Tipični vremenski dijagrami odskakivanja pri otpuštanju tastera



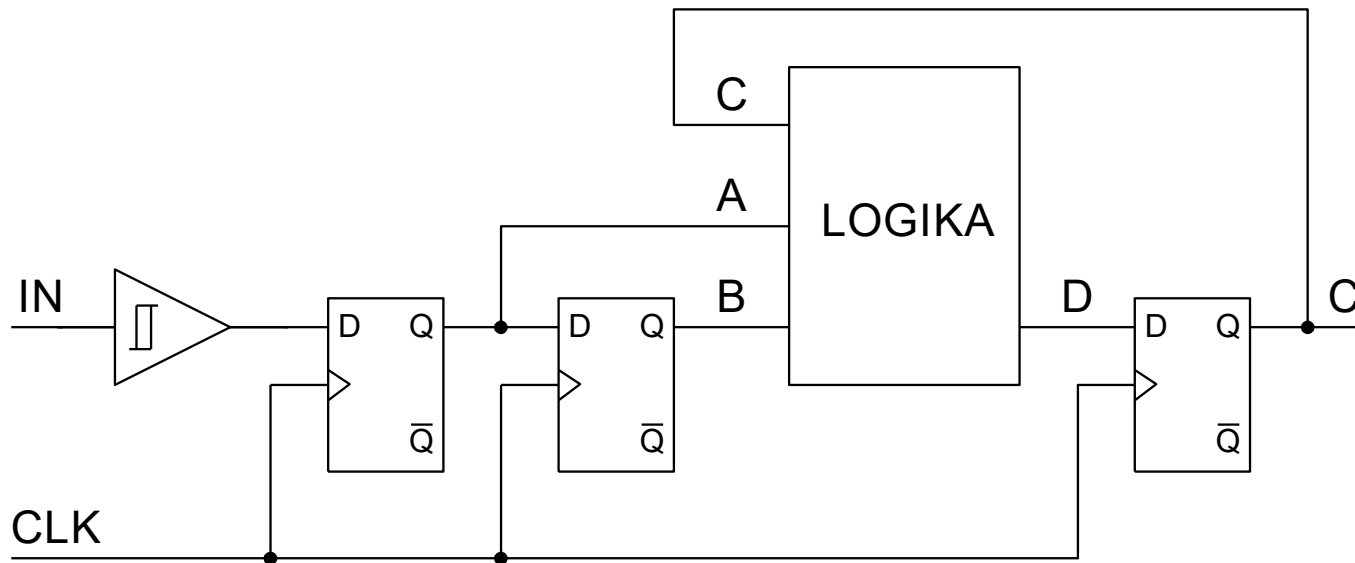
KOLO ZA DEBAUNSIranJE

- Da se ne bi pogrešno protumačilo veći broj promena stanja tastera ili prekidača umesto jedne promene, obavlja se hardversko i/ili softversko debaunsiranje tastera i prekidača (“*debouncing*”).
- Najjednostavnije kolo za hardversko debaunsiranje:



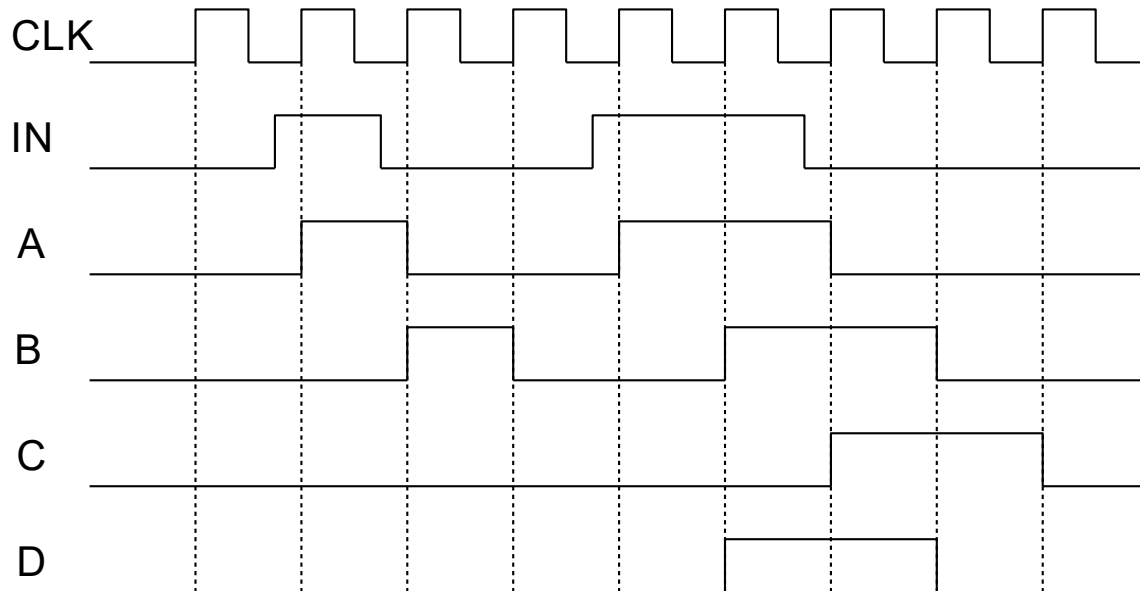
DIGITALNI FILTAR

- Poboljšano hardversko debaunsiranje ulaznog signala IN se izvodi pomoću digitalnog filtra koji eliminiše sve ulazne signale IN čije je trajanje kraće od jedne periode referentnog takta CLK, tj. sve signale u toku kojih se ne pojave bar dve uzlazne ivice referentnog takta CLK



DIGITALNI FILTAR

- Logika za debaunsiranje se projektuje na osnovu vremenskih dijagrama sa i odgovarajuće tabelle istinitosti

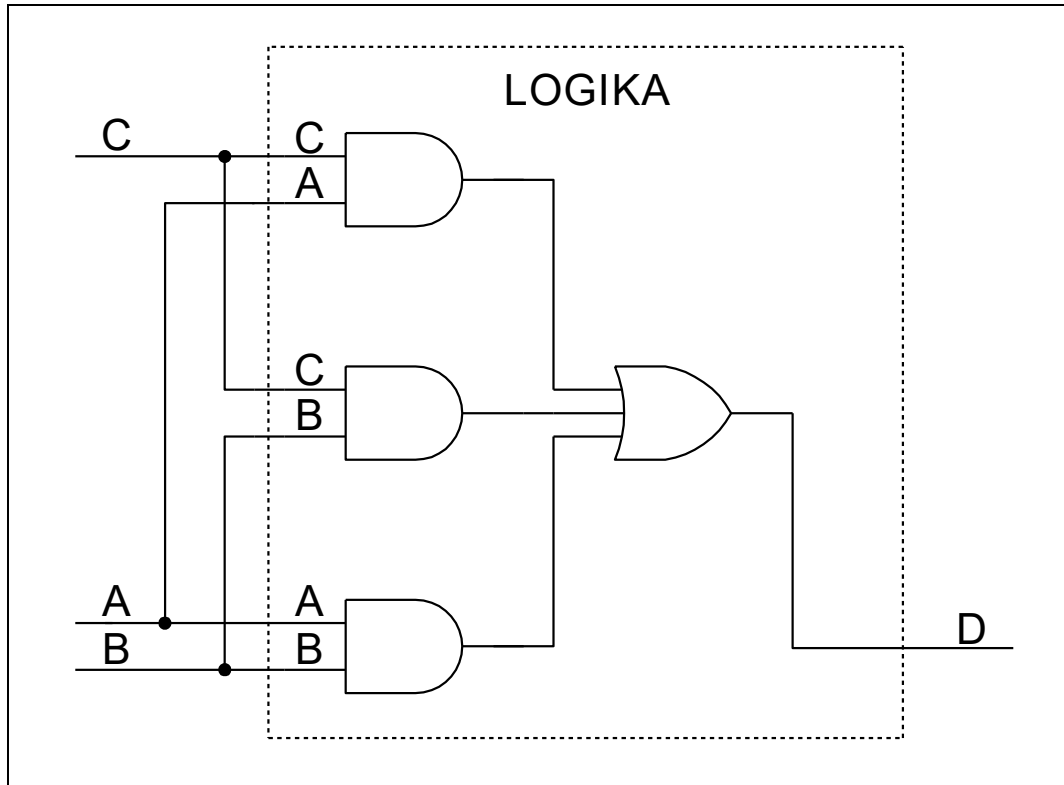


A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$D = A \cdot B + B \cdot C + C \cdot A$$

DIGITALNI FILTAR

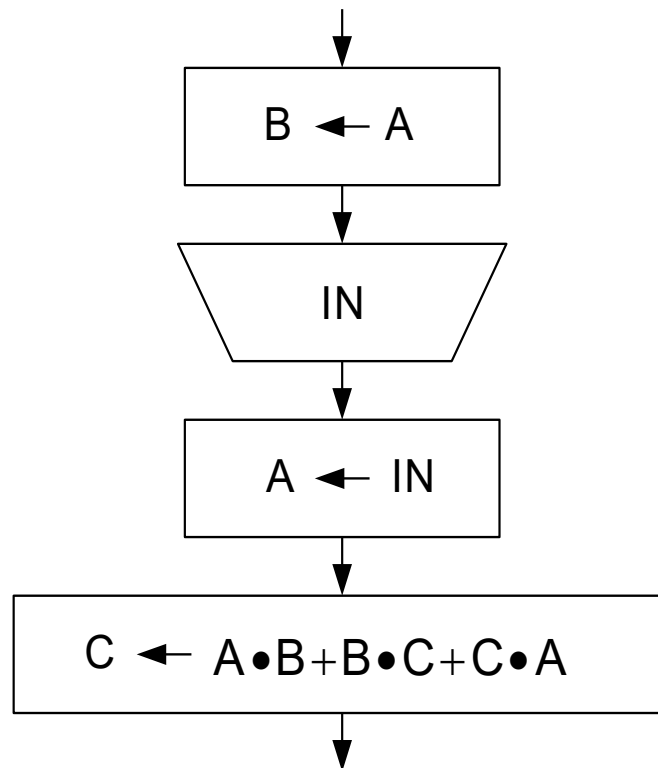
- Logika hardvera za debaunsiranje signala



- Hardversko debaunsiranje se prvenstveno može primeniti za nemultipleksiranu tastaturu i prekidače
- Primena hardverskog debaunsiranja u sistemima sprege sa multipleksiranim tastaturama je izuzetno komplikovana

SOFTVERSKO DEBAUNSIranJE

- Softversko debaunsiranje predstavlja repliku hardverskog debaunsiranja, pri čemu referentni takt odgovara periodu između dva obraćanja istom tasteru ili prekidaču
- Algoritam za debaunsiranje ulaznog signala IN:



SOFTVERSKO DEBAUNSIranJE

- Multipleksirana tastatura i prekidači se takođe debaunsiraju u prekidnoj proceduri, samo opisani algoritam postaje deo složenijeg algoritma koji istovremeno obavlja i skaniranje tastature
- Raspored tastera u multipleksiranoj tastaturi $M \times N$
 $M = 3$ reda povezana na linije za skaniranje tastature
 $N = 4$ tastera u svakom redu

3	2	1	0	
IN3	IN2	IN1	IN0	RED 0
IN7	IN6	IN5	IN4	RED 1
IN11	IN10	IN9	IN8	RED 2

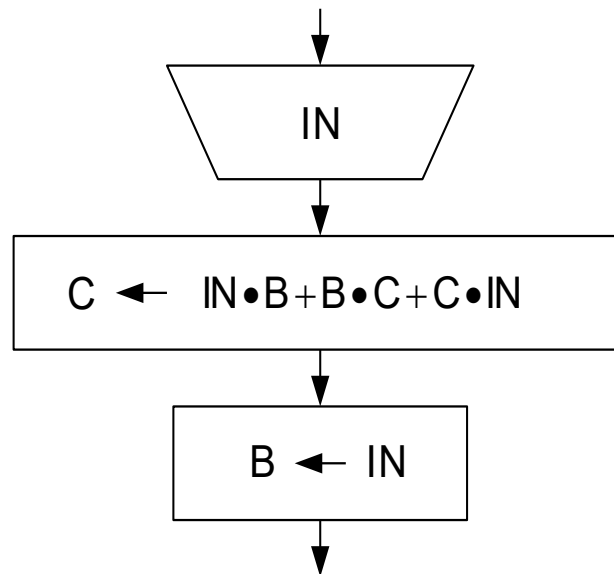
SOFTVERSKO DEBAUNSIranJE

- Memorijske lokacije za softversko debaunsiranje tastature

				3	2	1	0	
X	X	X	X	A3	A2	A1	A0	RED 0
X	X	X	X	A7	A6	A5	A4	RED 1
X	X	X	X	A11	A10	A9	A8	RED 2
X	X	X	X	B3	B2	B1	B0	RED 0
X	X	X	X	B7	B6	B5	B4	RED 1
X	X	X	X	B11	B10	B9	B8	RED 2
X	X	X	X	C3	C2	C1	C0	RED 0
X	X	X	X	C7	C6	C5	C4	RED 1
X	X	X	X	C11	C10	C9	C8	RED 2

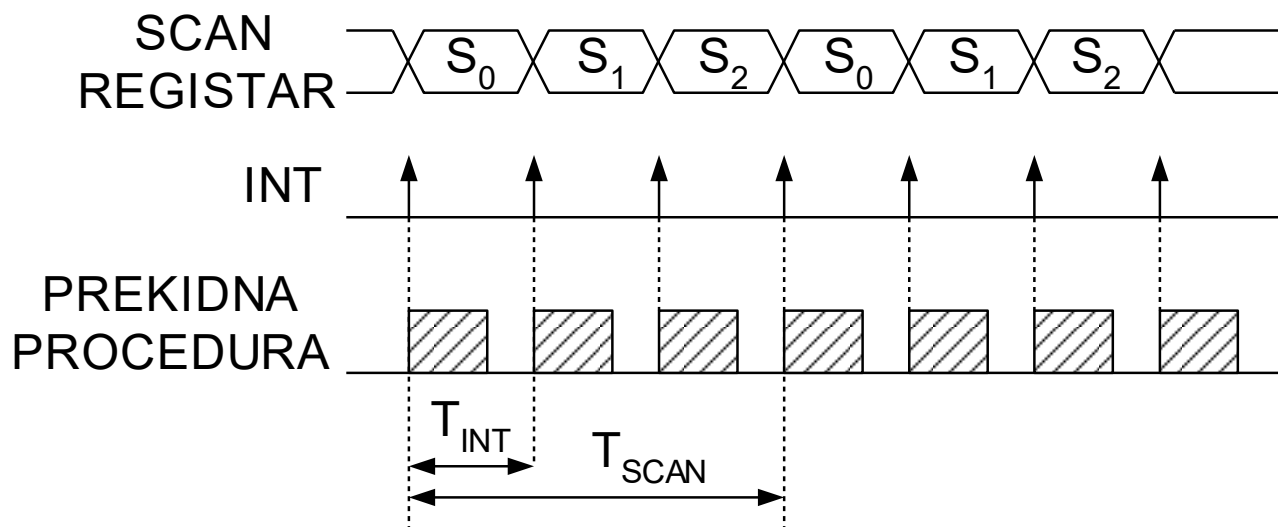
SOFTVERSKO DEBAUNSIranJE

- Svakom tasteru IN0..IN11 odgovara po jedan bit A0..A11, B0..B11, C0..C11, respektivno, unutar odgovarajućih bajtova koji odgovaraju redovima, odnosno linijama za skaniranje, dok su $8 - N = 4$ bita neiskorišćena u svakom bajtu radi jednostavnosti realizacije.
- Poboljšani algoritam za debaunsiranje ulaznog signala IN:



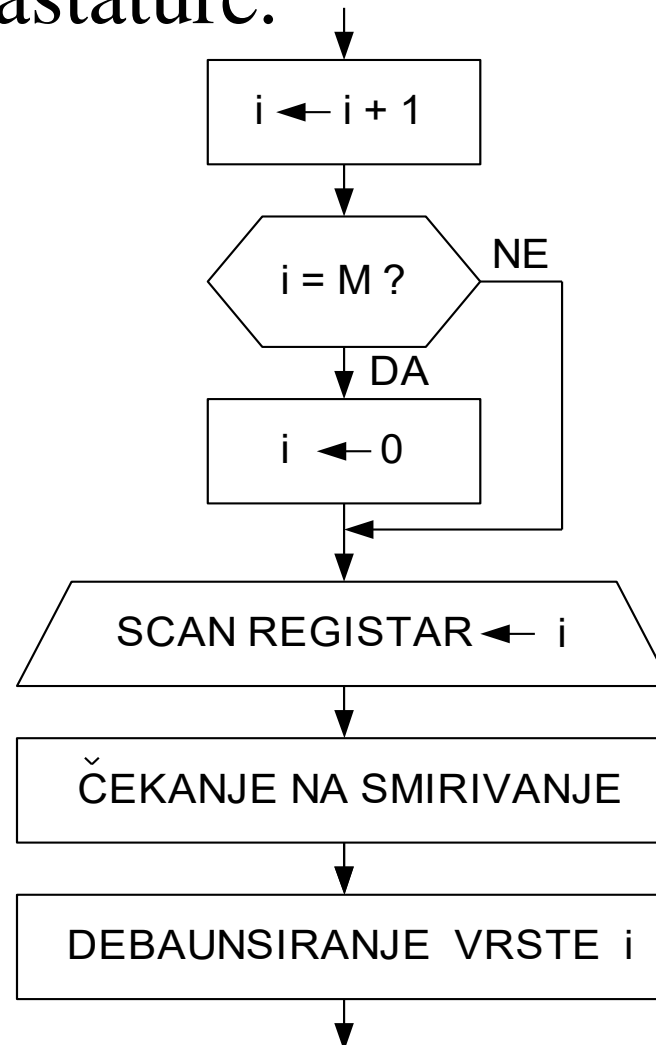
SKANIRANJE TASTATURE

- Prekidna procedura koja skanira i debaunsira tastere se obavlja posle prekidnog zahteva INT koga aktivira programabilni tajmer.
- Skaniranje multipleksirane tastature



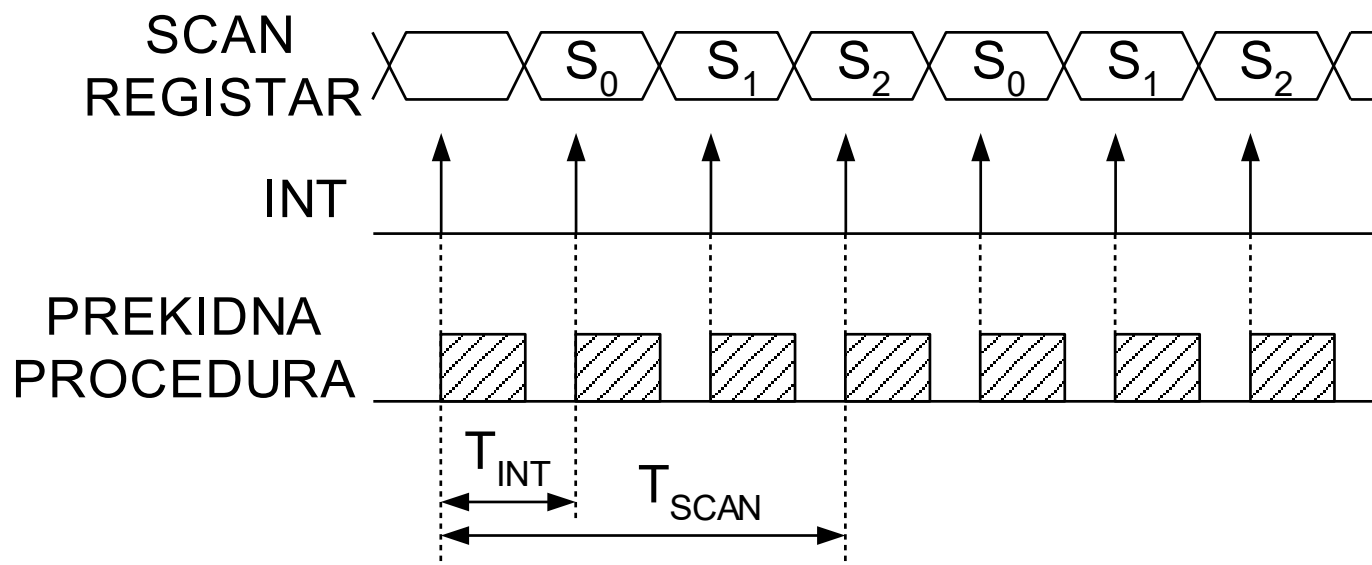
SKANIRANJE TASTATURE

- Algoritam za skaniranje i softversko debaunsiranje multipleksirane tastature:



SKANIRANJE TASTATURE

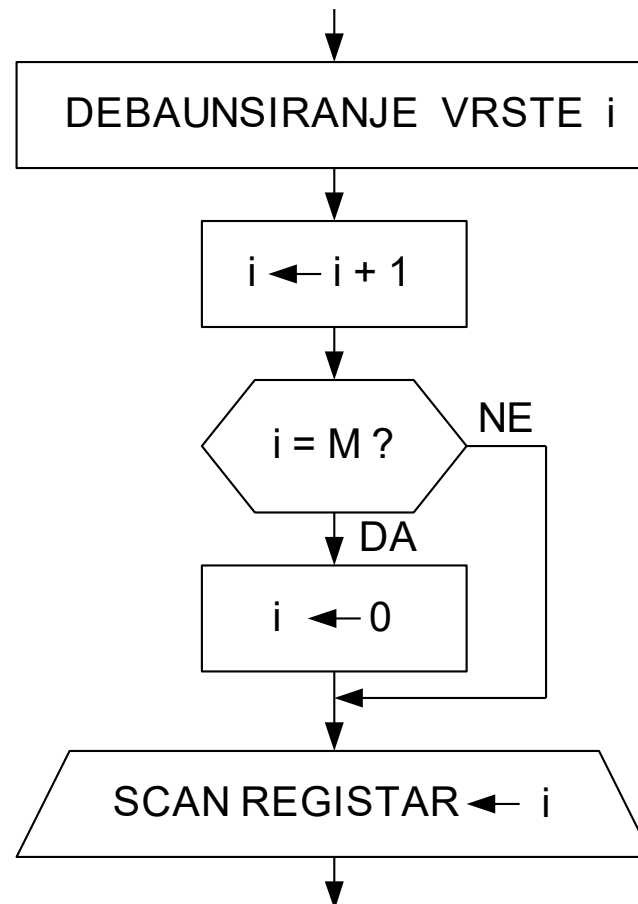
- Poboljšano skaniranje multipleksirane tastature



- Poboljšani algoritam zahteva da se u tekućoj prekidnoj proceduri po završetku algoritma za debaunsiranje, aktivira naredna linija za skaniranje, tako da se omogući da se signali na ulazu bafera smire do naredne prekidne procedure, koja debaunsira taj red.

SKANIRANJE TASTATURE

- Poboljšani algoritam za softversko debaunsiranje i skaniranje multipleksirane tastature



IZBOR PERIODA SKANIRANJA

- Izbor perioda generisanja prekidnih zahteva INT se obavlja na osnovu vremena odskakivanja (“*bounce*”), funkcionisanja digitalnog filtra i brzine kucanja.
- U slučaju nemultipleksirane tastature, period prekidne procedure treba da zadovoljava sledeće uslove da bi se izbeglo odlučivanje u toku vremena odskakivanja, ali i mogućnost nedetektovanja pritisnutog tastera:

$$T_{\text{BOUNCE}} < T_{\text{INT}} < (T_{\text{KEY}} - T_{\text{BOUNCE}}) / 2 \quad 2\text{ms} < T_{\text{INT}} < 24\text{ms}$$

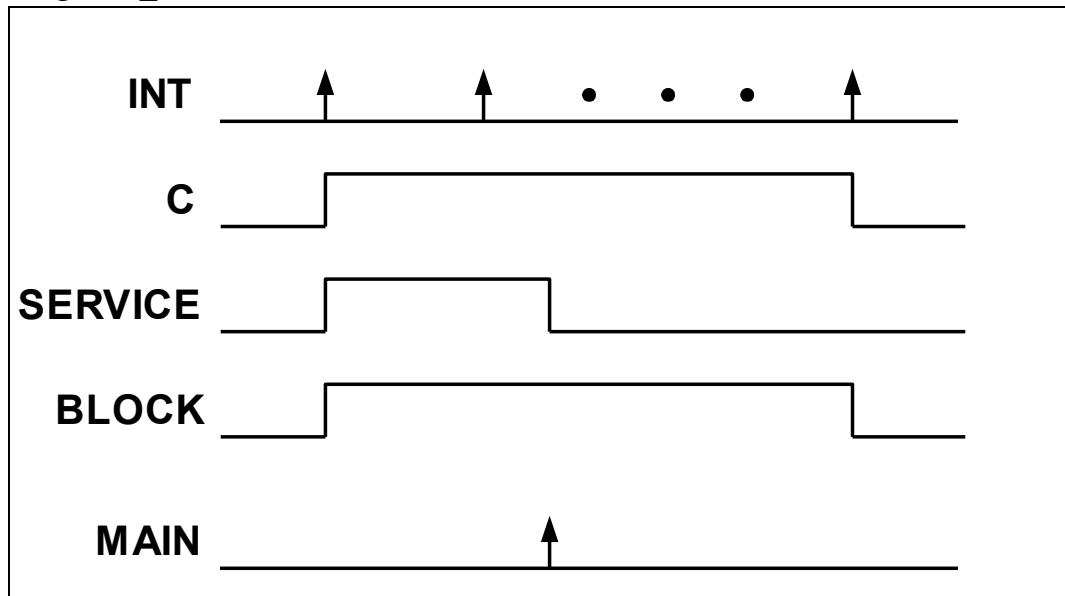
- U slučaju multipleksirane tastature sa M redova

$$T_{\text{BOUNCE}} < T_{\text{SCAN}} < (T_{\text{KEY}} - T_{\text{BOUNCE}}) / 2 \quad 2\text{ms} < T_{\text{SCAN}} < 24\text{ms}$$

$$T_{\text{INT}} = T_{\text{SCAN}} / M$$

SERVISNA PROCEDURA

- Servisiranje pritiska na taster:



- Pri promeni stanja debaunsiranog izlaza C sa “0” na “1” se aktivira odgovarajući servisni bit SERVICE, koji signalizira glavnom programu MAIN da je zahtevana akcija.
- Glavni program MAIN po završetku akcije resetuje servisni bit SERVICE, i time omogućava da naredni pritisak tastera zahteva istovetnu akciju.