

NUMĂRĂTOARE

Numărător Asincron legat la un Numarator Sincron

Martinovici Iasmina-Patricia
Anul 2, AC CTI RO, subgrupa 4.1

Cuprins

- 1. Introducere
 - 1.1 Numărătoare
 - 1.2 Bistabile
- 2. Descrierea de bază a numărătorului
 - 2.1 Numărător Asincron
 - 2.2 Numărător Sincron
- 3. Funcționarea proiectului
- 4. Testbench personal
- 5. Referințe

1. Introducere

1.1 Numărătoare

Numărătoarele sunt circuite logice secvențiale, destinate numărării impulsurilor care apar la intrarea lor și memorării rezultatului numărării. Acestea sunt structuri automate care își incrementează sau decrementează valoarea stocată sub forma de stare internă.

Numărătoarele se pot realiza cu ajutorul bistabilelor și a porților logice, cele din urmă având rolul de a stabili modul corect în care numărătorul își schimbă stările în procesul de numărare.

Evoluția numărătoarelor în cadrul stărilor se face într-o anumită ordine, fixată prin construcția numărătorului, în ritmul unui semnal de tact (clock).

Caracteristicile principale ale unui numărător:

- capacitatea numărătorului N_{max} , care reprezintă numărul maxim de stări stabile distincte ale numărătorului, $N_{max} < 2^n$, iar n fiind numărul de celule bistabile utilizate.

- rapiditatea în funcționare

- frecvența impulsurilor de numărare

- timpul de poziționare a numărătorului

Clasificarea numărătoarelor

Dupa modul de aplicare a impulsurilor de tact

- asincrone – tactul se aplica numai bistabilului celui mai puțin semnificativ, următoarele bistabile au semnalul de tact provenit de la ieșirea Q sau $\sim Q$ a bistabilului precedent;

- sincrone – impulsul de tact se aplica simultan tuturor bistabilelor.

1.2 Bistabile

În electronică, un bistabil (cum ar fi flip-flop-ul), este un circuit care are două stări stabile și poate fi utilizat pentru a stoca informații de stare. Circuitul poate fi realizat pentru a schimba starea prin semnale aplicate la una sau mai multe intrări de control și va avea una sau două ieșiri. Este elementul de stocare de bază în logica secvențială. Bistabilele sunt elemente fundamentale ale sistemelor electronice digitale utilizate în computere, comunicații și multe alte tipuri de sisteme.

Bistabilul JK.

J	K	Q_{next}	Actiune
0	0	Q	Se pastreaza valoarea
0	1	0	Se reseteaza
1	0	1	Se seteaza
1	1	$/Q$	Iesirea este valoarea precedenta negata

Aceste bistabile mai pot avea o intrare EN (Enable), care le va activa, în sensul că dacă intrarea EN este 0, atunci nu se va produce nicio schimbare. Tabelele sunt exstine în felul urmator, incluzând intrarea EN:

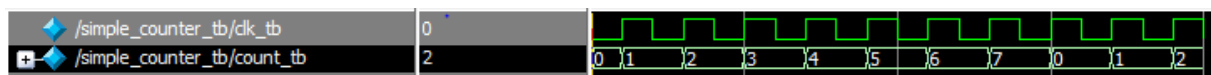
- X == “indiferent”

EN	J	K	Q_{next}	Actiune
0	X	X	Q	Bistabilul nu face nimic
1	0	0	Q	Se pastreaza valoarea
1	0	1	0	Se reseteaza
1	1	0	1	Se seteaza
1	1	1	$/Q$	Iesirea este valoarea precedenta negata

2. Descrierea de bază a numărătorului

2.1 Numărător Asincron

- Un numărător, în esență, trebuie sa numere de cate ori un semnal isi schimba valoarea de pe 0 pe 1 , adica tranzitiile de la 0 la 1, a unei intrari, care in proiect se va nota clk (de la clock, sau semnal de tact).
- Functionarea unui numarator se poate descrie conform diagramei de mai jos:
 - semnalul de clk_tb basculeaza(trece) de pe 0 pe 1, dar numararea (reprezentata prin count_tb) numara doar fronturile crescatoare (trecerea de pe 0 pe 1 a intrarii clk_tb)



- Descrierea unui astfel de comportament se poate face printr-un tabel:

CLK	COUNT
0	0
1	1
0	1
1	2
0	2
1	3
0	3
1	4
0	4
1	5
0	5
1	6
0	6
1	7
0	7

- Se notează cu $COUNT_0$ cel mai puțin semnificativ bit din COUNT și cu $COUNT_2$ cel mai semnificativ bit din COUNT
- Mai departe, se reprezintă ieșirea COUNT din zecimal în binar, fiecare bit fiind separat, după cum se observă în tabel.
- Din tabel, se observă că valorile lui $COUNT_2$, $COUNT_1$ și $COUNT_3$ nu se modifică față de valoarea anterioară atunci când CLK este 0, ci doar când CLK trece de pe 0 pe 1.
- Valorile scrise îngrosat, constituie un exemplu în acest sens.
- În urma acestei observații, se poate deduce faptul că în modul au loc modificări doar atunci când intrarea CLK este 1.

(1)

CLK	COUNT2	COUNT1	COUNT0
0	0	0	0
1	0	0	1
0	0	0	1
1	0	1	0
0	0	1	0
1	0	1	1
0	0	1	1
1	1	0	0
0	1	0	0
1	1	0	1
0	1	0	1
1	1	1	0
0	1	1	0
1	1	1	1
0	1	1	1

În continuare, se realizează un tabel în care se va pune accent doar pe intrarea CLK și ieșirea COUNT₀.

Se poate observa din tabel, analizând doar momentele când CLK trece de pe 0 pe 1, faptul că ieșirea COUNT₀ basculează între 0 și 1. Acest lucru poate fi descris, de asemenea, prin faptul că ieșirea COUNT₀ este egală cu valoarea precedentă negată atunci când CLK e 1. Astfel, tabelul acesta poate fi simplificat și să rezulte un tabel format doar din 2 cazuri, care vor depinde doar de clk.

CLK	COUNT ₀
0	0
1	1
0	1
1	0
0	0
1	1
0	1
1	0
0	0
1	1

În descriere liberă, valoarea lui COUNT₀ de la momentul t este egală cu valoarea lui COUNT₀ de la momentul $t - 1$, dacă CLK este 0, altfel este egală cu valoarea lui COUNT₀ de la momentul $t - 1$, negată

Luând în considerare aceste aspecte și cunoscând funcționarea bistabilelor, se poate deduce implementarea funcționării bitului COUNT₀ folosind bistabile JK, astfel: $J = 1$, $K = 1$, $EN = CLK$, $Q = COUNT$

CLK	COUNT ₀ _next
0	COUNT ₀
1	$\sim COUNT_0$

În continuare, ne vom axa pe alte 2 coloane din tabelul (1), și anume: Count₀ și Count₁

Se poate observa faptul că ieșirea COUNT₁ își schimbă valoarea doar atunci când COUNT₀ se schimbă de pe 1 pe 0. Dacă adăugăm funcția NOT, aplicată pentru COUNT₀, în acest tabel, se observă că ieșirea COUNT₁ se schimbă doar la tranziția lui NOT(COUNT₀) de pe 0, pe 1.

Această funcționare seamănă izbitor de mult cu funcționarea lui COUNT₀ față de intrarea CLK, putând descrie această funcționare printr-un tabel caracteristic ca mai jos, cu specificarea faptului că acest lucru nu se întâmplă tot timpul, ci doar la o modificare a lui NOT(COUNT₀)

NOT(COUNT ₀)	COUNT ₁ _next
0	COUNT ₁
1	$\sim COUNT_1$

COUNT0	NOT(COUNT0)	COUNT1
0	1	0
1	0	0
1	0	0
0	1	1
0	1	1
1	0	1
1	0	1
0	1	0

Tinand cont de observatiile anterioare, se poate lua decizia ca si COUNT₁ este implementat cu un bistabil JK, insa intrarea EN a acestui bistabil nu va fi CLK ca in cazul lui COUNT₀, ci COUNT₀ insusi negat.

Folosind aceeasi regula si pentru COUNT₂ reiese faptul ca intrarea pentru bistabilul JK responsabil iesirii COUNT₂ va avea intrarea EN egala cu NOT(COUNT₁)

Pentru resetarea tuturor bitilor (COUNT₂, COUNT₁, COUNT₀) va trebui sa se extindă functionarea bistabilului JK cu o inca intrare care va reseta iesirea Q, indiferent de intrarea EN.

O astfel de intrare este denumita uzual RESET (RST) sau CLEAR (CLR) si este de obicei active la 0, adica atunci cand aceasta intrare este 0 va provoca reactive de setare a bitului / bitilor pe 0. De obicei, aceasta intrare are prioritate fata de restul.

Avand noul numarator care se poate reseta la comanda, mai este nevoie doar de logica de resetare. Tinand cont ca atunci cand vrem sa resetam trebuie sa setam intrarea CLR_counter pe 0, iar cand nu vrem trebuie sa o mentinem pe 1, urmatorul tabel poate fi creat:

COUNT2	COUNT1	COUNT0	CLR_counter
0	0	0	1
0	0	1	1
0	0	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
0	0	1	1
0	0	1	1

2.2 Numărător Sincron

Din pacate, din cauza intarzierilor de propagare in circuite HW, descrise in Lucrarea de Laborator 8, metoda descrisa pana acum nu este fiabila din 2 motive:

1. Va avea rezultate intermediare gresita, pana cand informatia poate ajunge si la ultimul bistabil JK
2. Nu e scalabil: Daca intarzierea acumulata pana cand ajunge informatia la ultimul bistabil JK este mai mare de perioada bascularii intrari clk, atunci un astfel de numarator poate pierde cicluri de tact

O rezolvare pentru astfel de problema, este sa facem toate bistabilele JK active in acelasi timp, adica sa conectam intrarea clk la toate intrarile EN ale bistabilelor.

Avand toate bistabilele JK active in acelasi timp, trebuie sa gasim o alta metoda de a controla iesirea decat cea prezentata anterior, in care setam J si K tot timpul pe 1 logic.

Se va face acest lucru analizand din nou tabelul (1) si incercand sa creem o logica pentru J si K.

CLK	COUNT2	COUNT1	COUNT0
0	0	0	0
1	0	0	1
0	0	0	1
1	0	1	0
0	0	1	0
1	0	1	1
0	0	1	1
1	1	0	0
0	1	0	0
1	1	0	1
0	1	0	1
1	1	1	0
0	1	1	0
1	1	1	1

In continuare, se va extinde acest tabel astfel incat sa contina si informatiile despre J si K pentru fiecare bistabil in parte. Se va nota cu J_2 si K_2 intrarile J si K pentru bistabilul JK_2 , cu J_1 si K_1 intrarile J si K pentru bistabilul JK_1 si cu J_0 si K_0 intrarile pentru bistabilul JK_0 .

Se va tine cont de regula unui bistabil JK, si anume, cand $J == 0 \ \&\& \ K == 0$, iesirea e neschimbata, iar cand $J == 1 \ \&\& \ K == 1$, iesirea este valoarea precedenta negata.

Se vor ignora de asemenea, valorile cand $CLK = 0$. Astfel, se poate observa tabelul rezultat, unde coloana Timp este doar ajutatoare:

Timp	CLK	J2	K2	COUNT2	K1	J1	COUNT1	J0	K0	COUNT0
1	1	0	0	0	1	1	0	1	1	1
2	1	0	0	0	0	0	1	1	1	0
3	1	1	1	0	1	1	1	1	1	1
4	1	0	0	1	0	0	0	1	1	0
5	1	0	0	1	1	1	0	1	1	1
6	1	0	0	1	0	0	1	1	1	0
7	1	0	0	1	1	1	1	1	1	1

A se tine cont ca J_x si K_x sunt setate pe 1 doar atunci cand $COUNT_x$ urmeaza sa se schimbe.

Altfel zis:

- Daca J_x si K_x sunt ambele 1 la momentul t , atunci la momentul $t+1$, iesirea $COUNT_x$ se modifica fata de valoarea ei la timpul t .
- Daca J_x si K_x sunt 0, atunci iesirea $COUNT_x$ ramane aceeaasi ca la timpul t .
($x = \{0, 1, 2\}$)

Analiznd pe rand toate intrarile $J_2, K_2; J_1, K_1; J_0, K_0$, in raport cu iesirile $COUNT_2, COUNT_1, COUNT_0$, putem trage urmatoarea concluzie: J_x si K_x sunt ambele 1 doar cand toate celelalte iesiri $COUNT_{x-1}, COUNT_{x-2}, COUNT_{x-3} \dots$ sunt toate 1 in acelasi timp, oricare ar fi x numar natural > 0 , altfel J_x si K_x sunt 0.

Cunoscand tabelul de adevar al portilor logice, se poate spune ca J_x si K_x sunt egale cu operatia AND aplicata tuturor iesirilor $COUNT_{x-1}, COUNT_{x-2} \dots$

Exceptie face J_0 si K_0 care sunt tot timpul 1 !

3. Funcționarea proiectului

Inițial se determină de câte module este nevoie pentru realizarea proiectului. Modul meu de rezolvare a necesitat 9 module, iar ordinea lor reiese din enumerarea lor:

- JK_flip_flop – se descrie functionarea bistabilului JK, cu intrarea EN
- JK_sync_counter
- JK_sync_counter_clear - pentru resetarea bitilor (count) s-a extins functionarea bistabilului JK cu inca o intrare care va reseta iesirea Q, indiferent de intrarea EN
- JK_sync_to_async
- JK_sync_counter_tb
- JK_async_counter
- JK_async_counter_clear – pentru resetarea bitilor (count) s-a extins functionarea bistabilului JK cu inca o intrare care va reseta iesirea Q, indiferent de intrarea EN

- h. JK_async_counter_tb
- i. JK_async_sync_tb

4. Testbench personal

Au fost realizate 3 testbench-uri.

Primul, JK_sync_counter_tb, reprezintă testbench-ul pentru numărătorul sincron.

Al doilea, JK_async_counter_tb, reprezintă testbench-ul pentru numărătorul asincron.

Ultimul testbench, JK_async_sync_tb, este pentru numărătoarele asincron și sincron.

5. Referințe

Referinte interne

Proiect FC,

<https://docs.google.com/presentation/d/18CaJzgQz1uC6Gtd2XeJYpuy0oSeVWycH/edit#slide=id.p1>

Bistabile, https://docs.google.com/document/d/1Eq7M9C-ACNcU-ZdQmThcXE-1WnBg_rN3rGWumh3tB78/edit#heading=h.azxuuyrx7h66

Sintaxa pentru front crescator sau descrescator

https://docs.google.com/document/d/1zLWCyt6pa1UYGo3qWaocylgxtgLo-_dlxReI5XQfzN4/edit#heading=h.ufi7ra1ho1f6

Curs 7, Fundamentele Calculatoarelor,

<https://drive.google.com/file/d/1MA8ZuWNiaJNXfRYvYE44jV7SfPWir5Gi/view?usp=sharing>

Allen Farcaș, Lucrare laborator 5, Fundamentele Calculatoarelor,

https://drive.google.com/file/d/1OxsZtdn0k3iAex_Xdjlhr0_LROo9_ffN/view?usp=sharing

Referinte externe

<https://circuitdigest.com/tutorial/asynchronous-counter>, Asynchronous Counter

<https://circuitdigest.com/tutorial/synchronous-counter>, Synchronous Counter