

## Sinteza sistemelor secvențiale asincrone folosind dispozitive de sincronizare

**Asist.ing.Mihai Timis**  
**Technical University “Gh.Asachi” Iași**  
**mtimis@cs.tuiasi.ro**

**Abstract.** In present paper, I propose a new method for analysis and synthesis an asynchronous sequential system. It contains a clock signal who will drive on the states of the sequential system. To control sequential system, it is used a clock signal who contain a set of Master – Slave D latches circuits, logic gates. It is build from the sequential system transition matrix.

**Keywords:** Transition Matrix, FSM Graph, FSM Transition Table, Latch D, Latch D Master-Slave, Local Clock, Logic Gates.

### 1 Introducere

În cele ce urmează voi prezenta o metodă de analiză și sinteză a unui sistem secvențial asincron, folosind un semnal de clock local pentru controlul tranzițiilor sistemului.

Ideea de bază constă din construirea unui semnal de clock cu circuite logice combinaționale, folosind tabela de funcționare a sistemului secvențial. Acesta este implementat folosind porți logice elementare (ȘI, SAU, OR).

Folosind această metodă sistemul secențial asincron poate fi modelat mult mai ușor, evitându-se de asemenea fenomenele de hazard logic.

### 2 Descrierea Metodei

Un astfel de sistem de control folosind semnale de clock interne sistemului secvențial asincron, este prezentat în figura 1.

Acesta este alcătuit dintr-u sistem logic combinațional, elemente de stocare date – latch de tip D, semnale de intrare – input1, input2...inputN, semnale de ieșire – output1, output2...outputM, variabile de stare – s1, s2...sk.

Sistemul logic combinațional ce implementează semnalul de clock local mai poate fi denumit unitate de sincronizare, este folosit pentru a controla modalitatea de generare a stărilor sistemului secvențial asincron cât și pentru eliminarea posibilităților de apariție a fenomenelor de hazard.

Spre deosebire de sistemele secvențiale sincrone, în cele asincrone, modificarea stărilor apare numai atunci când un set nou de date de intrare este generat. Astfel, semnalul de clock intern depinde numai de stări și de intrări. Acesta este generat local, pentru fiecare modul în parte.

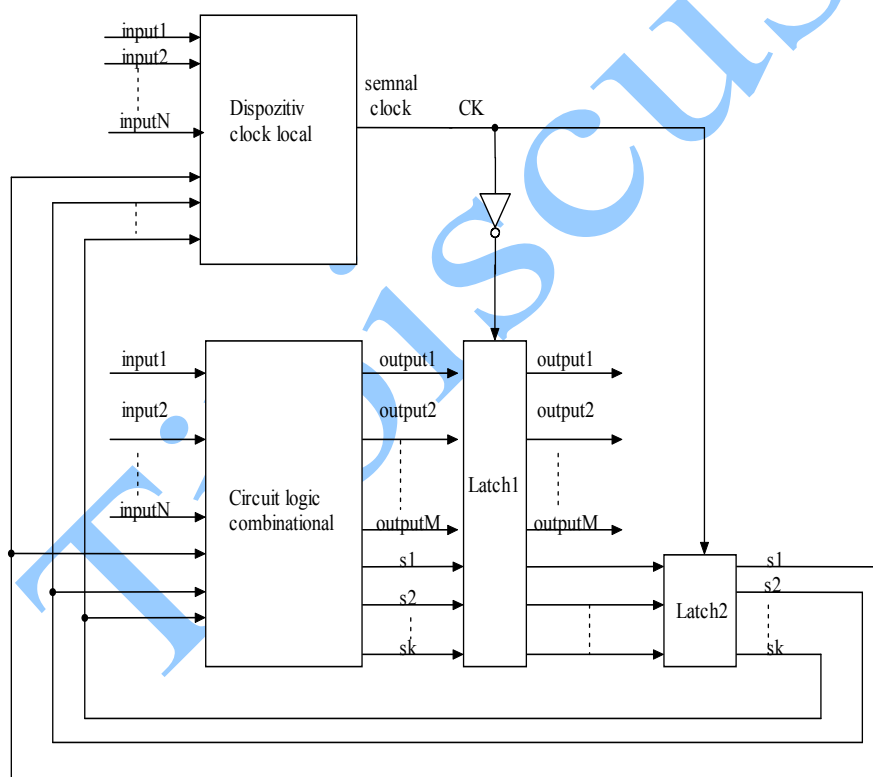


Figura 1

Semnalul de clock nu este memorat inițial, după cum se observă este inversat pentru a putea obține două valori logice, 0 și 1. Acesta controlează circuitele de tip latch – memorie, latch1 respectiv latch2.

Fiecare variabilă de ieșire este memorată într-un circuit latch – latch1, comandat de semnalul de clock CK, pe front negativ. De asemenea, fiecare variabilă de stare posedă un circuit de tip Master – Slave, latch1 – latch2. Variabilele de stare sunt memorate în latch atunci când semnalul de clock este high.

Atunci când semnalul de clock CK este low, circuitele latch1 sunt comandate să mențină valorile pentru semnalele de ieșire respectiv pentru stările sistemului secvențial asincron. În momentul când CK devine High, se inhibă circuitul latch1 - Master, deschizându-se circuitul latch 2 – Slave. Acesta preia stările sistemului, le memorează și le transmite prin bucla de reacție – feedback către dispozitivul de clock cât și spre circuitul logic combinațional.

Astfel, fiecărei stări îi va corespunde date de ieșire, sistemul procesând informațiile prezente la intrare conform tabelii de tranziție – funcționare.

Un circuit latch2 este un latch D, Master – Slave, prezentat în figura

2.

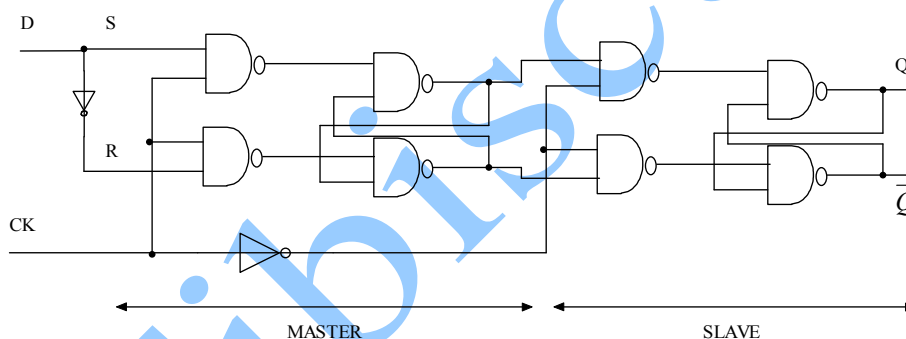


Figura 2

Acesta necesită un mecanism la care modificarea ieșirii să aibă loc la tranziția semnalului de clock CK – activare pe front, iar intrările pot să se modifice în celelalte perioade de timp fără a afecta ieșirile. La activarea pe front pozitiv, comanda este generată de tranziția de la 0 logic la 1 logic.

Pentru o bună funcționare este necesară o perioadă de timp de prestabilire – time setup, în acest timp, intrarea nu trebuie să se schimbe. De asemenea este necesar un semnal de menținere – time hold, imediat după tranziție, intrarea nu trebuie să se schimbe. Acești timpi sunt activi în jurul tranziției pozitive a semnalului de ceas CK.

O abordare a acestei probleme a fost utilizarea a două latch-uri în cascadă, Master – Slave. Primul latch, numit stăpân – Master, captează

valorile de intrare pe durata pozitivă a semnalului de clock CK, iar al doilea latch, numit sclav – Slave, folosește ieșirea primului latch pe durata negativă a semnalului de clock CK. Semnalul de ieșire a latch-ului se va modifica pe tranziția negativă a semnalului de clock CK.

### 3 Exemplu de implementare

Presupunem sistemul secvențial descris de graful de fluentă din figura 3.

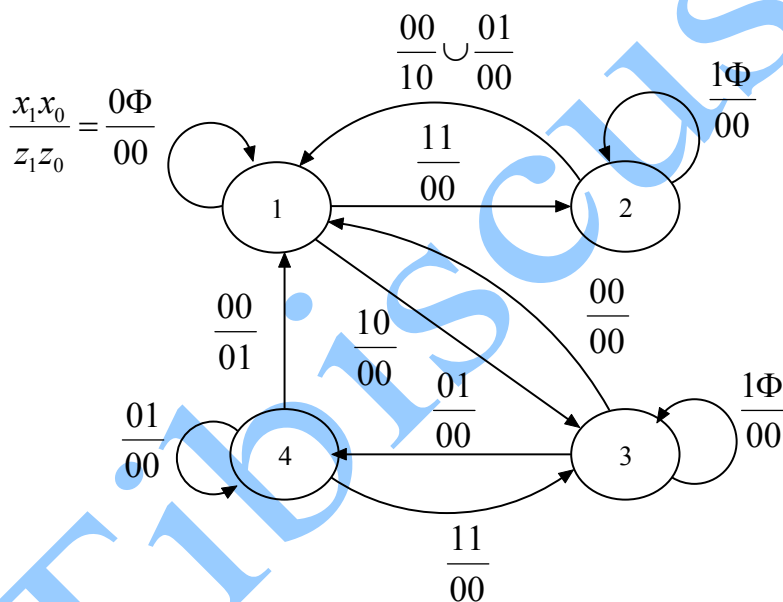


Figura 3

Ne propunem sinteza sistemului secvențial asincron descris prin graful de fluentă din figura 3.

Astfel, aplicând metoda expusă anterior, se obține implementarea din figura 4.

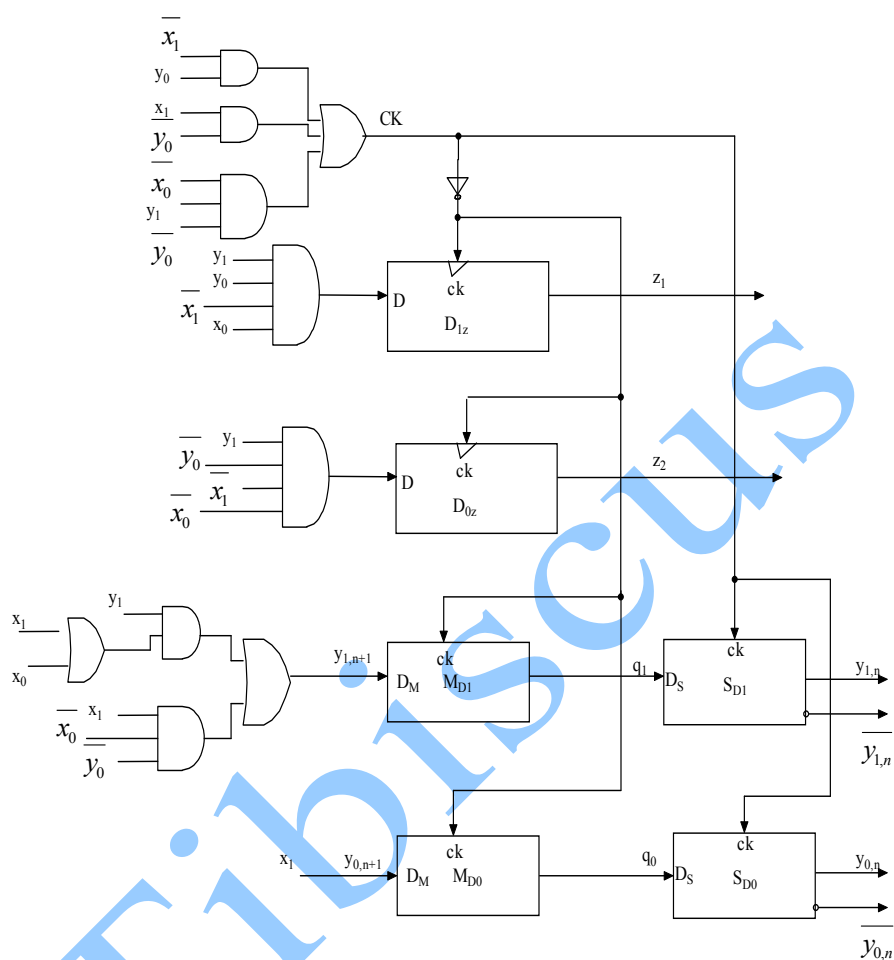


Figura 4

Fiecare variabilă de ieșire are atașat un circuit latch de tip D, circuitele latch1 mențin variabilele de ieșire ale sistemului secvențial asincron, în timp ce circuitele latch2 sunt blocate.

Fiecare variabilă de stare este menținută într-un set de latch-uri master-slave de tip D. Acestea sunt comandate de semnalul de clock CK doar atunci când are loc o tranziție către o nouă stare.

Semnalul de clock local CK este unlatched, este realizat folosind porți logice elementare. Acesta este low pe timpul cât comandă latch1, în momentul în care semnalele de intrare indică o schimbare a stării prezente către starea următoare, semnalul de clock CK devine high, apoi devine low,

conform tabelii de fluență. Dacă se rămâne în starea prezentă, semnalul de clock CK rămâne low. În funcționarea normală a semnalului de clock pot apărea fenomene de hazard.

Inițial, semnalul de clock CK este low, semnalele de intrare, stare, ieșire sunt de asemenea low. Astfel circuitele latch1 sunt transparente, datele trec direct prin acestea. De asemenea trebuie evitate apariția fenomenului de hazard la generarea semnalelor de ieșire. În momentul în care  $x_1x_0$  devin high, ieșirile  $z_1z_0$  rămân low până ce sunt generate complet semnalele de intrare date. Conform grafului de fluență, se observă că semnalele de ieșire vor rămâne low în continuare, sistemul tranzitând în starea 2. În momentul în care  $x_1x_0 = 00$  în starea 2, semnalele de ieșire rămân low până ce se generează semnalele de intrare, moment în care  $z_1z_0=10$ , sistemul reîntorcându-se în starea 1. Se observă faptul că semnalul de ieșire  $z_0=1$  în momentul în care sistemul se află în starea 4, semnalele de intrare fiind  $x_1x_0 = 00$ . Toate aceste tranziții, de fapt respectarea tabelii de fluență și a grafului de fluență a sistemului secvențial este implementată folosind circuite latch, un circuit logic combinațional ce implementează semnalul de clock generat intern, CK, ce depinde de semnalele de intrare, variabilele de stare, figura 4.

#### 4 Concluzii

Din moment ce sistemul se stabilizează într-o anumită stare, semnalul de clock CK este low, resetat. De asemenea, semnalul de clock este generat numai atunci când are loc o tranziție dintr-o stare în alta, în restul cazurilor acesta fiind null. Semnalele de ieșire se generează pe durata low a semnalului de clock CK. Se observă că logica combinațională a generatorului de clock CK este destul de simplă, fiind realizată folosind porți logice elementare.

#### 5. Bibliografie

- [MT01] **Calin Monor, Mihai Timis** –*Synthesis Of Reconfigurable Sequential Digital Devices*, 7<sup>th</sup> International Symposium on Automatic Control and Computer Science - SACCS, Iasi, October 2001.

- [Rot99] **Roth, Charles H.** – *Fundamentals of Logic Design*, West Publishing Company, 1999.
- [Tim03] **Mihai Timiș** – *Analiza și Sinteza Dispozitivelor Numerice. Aplicații*, Performantica, Iași, 2003.
- [VB86] **Alexandru Valachi, M.Barsan** – *Digital Techniques For Automata Synthesis*, Junimea, Iași, 1986.
- [VHS93] **Alexandru Valachi, Fl.Hoza, R. Silion** – *Analysys and Synthesis of Digital Systems*, Nord Est, Iași, 1993.
- [VM00] **Al.Valachi, C. Monor** – *Asynchronous Digital Systems Design*, Renasterea Romana, Iași, 2000.
- [VST04] **Alexandru Valachi, Radu Silion, Mihai Timis** – *Improvement of FSM Synthesis Using MSI and LSI Circuits*, 8<sup>th</sup> International Symposium on Automatic Control and Computer Science - SACCS, Iasi, October 2004.
- [Wil02] **Barry Wilkinson** – *Digital Electronics. Based concepts*. Teora, 2002.

Tibiscus