

Memorii

1. Memorii de tip FIFO (First In First Out)

O astfel de memorie este caracterizată prin faptul că primul cuvânt înscris, va fi primul cuvânt ce va apărea la ieșire la o operație de citire. Pentru a realiza o astfel de memorie, sunt necesare RD. Întrucât aplicațiile cele mai utilizate folosesc cuvinte de 8 biți, vom prezenta o memorie de 8 biți:

Capacitatea unei astfel de memorii este dată de lungimea RD (de numărul celulelor RD).

Se utilizează pentru memorarea temporară a datelor ce intră într-un sistem (circuit tampon). După recepționarea unui bloc de date (memoria este plină), se transferă conținutul în sistem. Astfel nu se întrerupe funcționarea (bucla principală de program) la recepționarea fiecărui cuvânt, înlăturându-se timpii pierduți pentru tratarea operațiilor de întrerupere. De exemplu conectarea unui convertor A/D la un sistem cu microprocesor se face prin intermediul unei astfel de memorii.

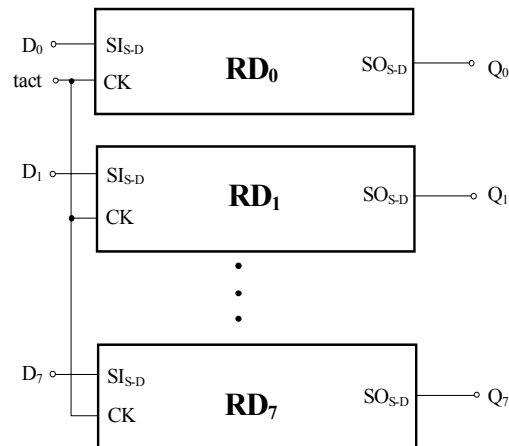


Figura 1

a) memorii de tip LIFO (Last In First Out)

Aceasta este o memorie de tip “stivă”: ultimul cuvânt înscris va fi primul cuvânt citit. Realizarea unei memorii LIFO se face cu registre universale, ca în figura următoare. Lungimea cuvântului ce poate fi memorat fixează numărul de RU necesare, iar capacitatea maximă a memoriei stabilește

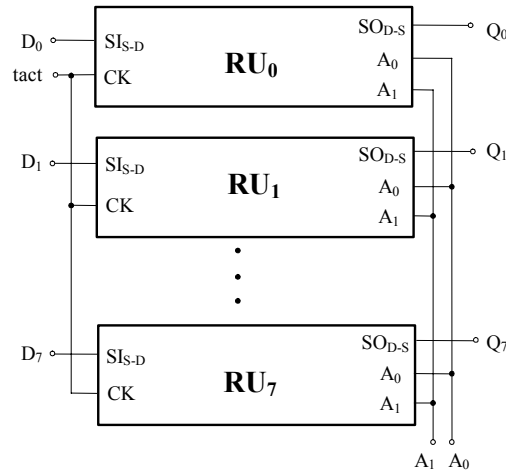


Figura 2

lungimea fiecărui RU. Tabelele următoare ilustrează funcționarea memoriei LIFO de 4 cuvinte. Pe intrările SI_{S-D} se aplică cuvântul de memorat. Modul de funcționare al RU este selectat de deplasare stânga-dreapta prin $A_0=A_1=0$. Fiecare impuls de tact va înscrie cuvântul aflat la D_{0-7} în registre. La tactul următor, primul cuvânt se deplasează în celulele următoare, iar noul cuvânt aflat pe intrări se memorează în primele celule din fiecare RU, ș.a.m.d. Când se dorește citirea din memorie, se stabilește $A_0=1$ și $A_1=0$, fiecare tact realizând citirea informației existente în primele celule, și deplasarea conținutului din registre spre stânga.

$A_0=0, A_1=0 \Rightarrow$ înscriere registru (deplasare S-D)

	RU ₀	RU ₁	RU ₂	RU ₃	RU ₄	RU ₅	RU ₆	RU ₇
Q ₀	x ₀	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇
Q ₁								
Q ₂								
Q ₃								

↓

$A_0=0, A_1=0 \Rightarrow$ înscriere registru (deplasare S-D)

	RU ₀	RU ₁	RU ₂	RU ₃	RU ₄	RU ₅	RU ₆	RU ₇
Q ₀	y ₀	y ₁	y ₂	y ₃	y ₄	y ₅	y ₆	y ₇
Q ₁	x ₀	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇
Q ₂								
Q ₃								

↓

LUCRAREA nr. 13

A ₀ =1, A ₁ =0 ⇒ înscrie registru (deplasare S-D)								
	RU ₀	RU ₁	RU ₂	RU ₃	RU ₄	RU ₅	RU ₆	RU ₇
Q ₀	z ₀	z ₁	z ₂	z ₃	z ₄	z ₅	z ₆	z ₇
Q ₁	y ₀	y ₁	y ₂	y ₃	y ₄	y ₅	y ₆	y ₇
Q ₂	x ₀	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇
Q ₃								

↓

A ₀ =0, A ₁ =0 ⇒ citire registru (deplasare D-S)								
	RU ₀	RU ₁	RU ₂	RU ₃	RU ₄	RU ₅	RU ₆	RU ₇
Q ₀	y ₀	y ₁	y ₂	y ₃	y ₄	y ₅	y ₆	y ₇
Q ₁	x ₀	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇
Q ₂								
Q ₃								

Acest tip de memorie se utilizează la memorarea adreselor, de la care se abandonează programul, pentru executarea unor subrutine de tratare a cererilor de întrerupere, pentru ca programul să știe unde să revină. În cazul în care a apărut o cerere de întrerupere, sistemul a salvat adresa instrucțiunii la care ajunsese programul $x_7x_6...x_1x_0$, după care începe execuția subprogramului de tratare a întreruperii. În acest timp survine o altă întrerupere, care determină suspendarea activității și saltul la altă adresă. Se salvează în memorie adresa curentă $y_7y_6...y_1y_0$ și se execută o altă subrutină. Și aceasta este întreruptă la rândul ei. Se salvează în memoria LIFO noua adresă curentă $z_7z_6...z_1z_0$ și se sare la adresa subrutinei de întrerupere asociate noii întreruperi. Terminare tratării acestei întreruperi, duce la revenirea în program, la adresa $z_7z_6...z_1z_0$ și execuția până la capăt a subrutinei abandonate. La terminarea ei se va relua subrutina anterioară de la adresa $y_7y_6...y_1y_0$, și în final se va reveni în bucla principală de program, la adresa $x_7x_6...x_1x_0$.

5. Lucrări de efectuat în laborator

Se completează fișa de laborator disponibilă la adresa:

http://www.ee.tuiasi.ro/~demm/Digital_Circuits/FișaLab13.DOC

FIȘA LABORATOR

1. Se realizează un registru de deplasare cu CBB de tip D cu 8 celule. Se utilizează acesta pentru a realiza o memorie FIFO de 8 locații de 8 biți. Se aplică la intrare cuvinte de 8 biți și se urmărește deplasarea acestora prin memorie de la un capăt la altul. Ieșirile se realizează sub formă de magistrală. Formele de undă rezultate se copie mai jos. Se notează timpzii de întârziere și valorile logice pe formele de undă. Se compară rezultatele cu tabelul de adevăr.

Ck																				
Q ₀ [7..0]																				
Q ₁ [7..0]																				
Q ₂ [7..0]																				
Q ₃ [7..0]																				
Q ₄ [7..0]																				
Q ₅ [7..0]																				
Q ₆ [7..0]																				
Q ₇ [7..0]																				
Q[7..0]																				

2. Se realizează un registru universal cu $N=4$ în MaxPlus II și se construiește apoi o memorie LIFO de 8 biți. Se înscriu 3 cuvinte în memorie și apoi se extrag. Formele de undă rezultate se copie mai jos. Se notează timpzii de întârziere și valorile logice pe formele de undă. Se compară rezultatele cu tabelul de adevăr.

Ck																				
A ₀																				
A ₁																				
D[7..0]																				
Q ₀ [7..0]																				
Q ₁ [7..0]																				
Q ₂ [7..0]																				
Q ₃ [7..0]																				