Lucrarea 4

Sumatoare Carry Select şi Carry-Skip

Această lucrare prezintă tehnicile de construcție eficientă a sumatoarelor bazate pe structurile RCA: Carry Select şi Carry Skip. Folosind redundanța structurală, comportamentul specific al semnalelor generate de circuitele în tehnologia CMOS, şi alegerea inteligentă a dimensiunii segmentelor CSeA şi CSkA se pot obține sumatoare performante, a căror întârzieri sunt evaluate şi comparate pe parcursul lucrării.

1 Carry Select Adder (CSEA)

Este o modalitate diferită de a efectua operația de adunare în mod paralelizat, pe baza unei investiții suplimentare care este sugerată de figura de mai jos (Figura 4.1) .

În anumite tehnologii faptul că semnalul de carry trebuie să comande foarte multe intrări duce la întârzieri şi de aceea semnalul trebuie amplificat. Carry-ul real selectează suma calculată în paralel atât pentru eventualitatea în care el va fi ‘1’, cât şi în cea în care carrz-ul va veni ca ‘0’, (i.e. suma pe segnment este anterior calculată). Problema care se pune este cum să se facă împărțirea intrărilor în segmente RCA astfel incât adunarea să se facă în forma cea mai favorabilă.

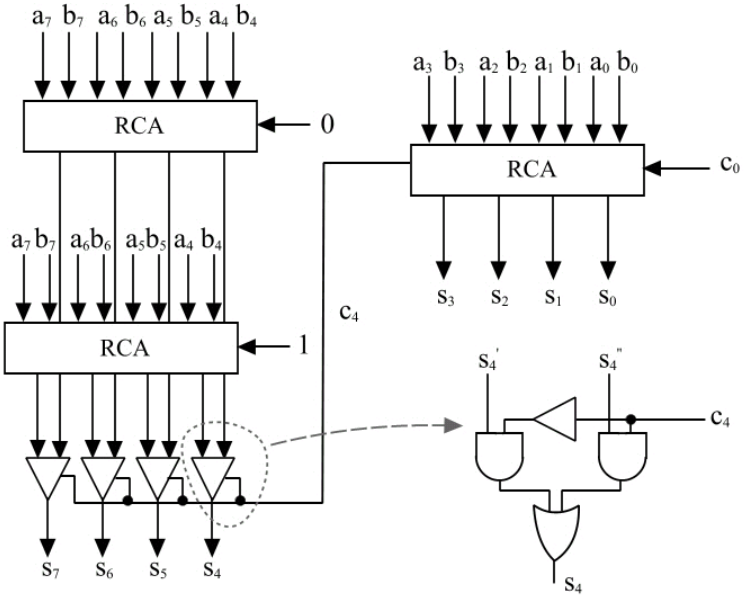


Figura 4.1 Soluția Carry Select Adder pe 8 biți: se împarte sumatorul RCA în segmente CSEA, care sunt dublate, şi care precalculează în paralel suma pe segment, atât în eventualitatea în care carrz va fi '0', cît şi în cea în care va fi '1'.

În Figura 4.1 nu este prezentat multiplexorul 2 la 1 care trebuie să selecteze carry out-ul pe care în generează segmentele CSEA. Acest multiplexor va ma genera o întârziere de 2*d*, adică exact atât cât generează un rang suplimentar RCA. Prin urmare, este favorabilă o implementare CSEA în care segmentele CSEA să fie inegale.

De exemplu, un sumator pe 32 de biți ar fi implementat cel mai favorabil în această tehnologie cu segmente 6-7-6-5-4-4.

Ideea este următoarea: din moment ce multiplexorul oricum întârzie carry-ul cu 2*d*, ce ce să nu aglomerăm adunarea pe încă un rang în proximul segment, pentru că selecția oricum va întârzia până la apariția carry-ului?

2 Carry Skip Adder (CSkA)

Această soluție este situată ca și cost și performană de siliciu între RCA și CLA, și ea pornește de la ecuația corespunzătoare lui CLA:

$$\begin{cases} P_{0,3} = p_3 \cdot p_2 \cdot p_1 \cdot p_0 \\ G_{0,3} = g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0 \end{cases} \quad (4.1)$$

S-a ajuns soluția tehnică sugerată de Figura 4.2:

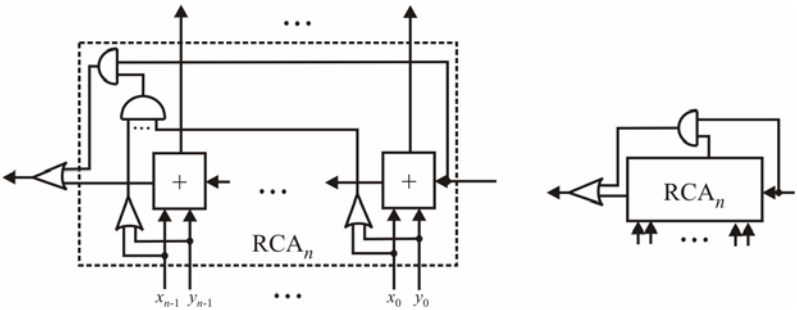


Figura 4.2 Celula de însumare CskA pe *n* biți.

Aceast soluție tehnică se bazează pe faptul că în ceea ce privește tehnologia CMOS, prin precharging, este posibil să se realizeze inițial carry-urile cu „0“ (tot lanțul de carry este pe ‘0’), prezentate în Figurile 4.3 și 4.4. Carry-ul se poate genera simultan pentru toate segmentele, ignorând carry-ul care intră în porțile *AND*, șuntându-se astfel propagarea carry-ului prin întreg segmentul.

În situația în care din intrări pentru un anumit segment nu s-a generat carry datorită carry-ului de intrare în segment, este posibil totuși să fie generat carry dar acesta nu va fi generat în manieră serială conform soluției RCA ci va omite („skip“) propagarea serială prin traversarea porții *AND*, condiționată pe intrări de factorul logic $P_{i,j}$ (ieșirile porților *AND* din Figura 4.2) care, în manieră CLA, reprezintă produsul $p_3 \cdot p_2 \cdot p_1 \cdot p_0$. Se spune că în mod corespunzător fiecărui segment RCA se generează termeni de tipul „G“. Termenii $P_{0,0}, P_{0,1}, \ldots$ se obțin mult mai simplu decât termenii $G_{0,0}, G_{0,1}, \ldots$ .Termenii de tip „P“ sunt

generați la segmentele externe. Să admitem că în primul segment RCA avem în general k ranguri și să admitem că pe 2 niveluri logice avem o întârziere d. În aceste condiții în situația cea mai defavorabilă când avem o propagare de transport este valabilă relația de mai jos:

$$T_{ad} = (k + (\frac{n}{k} - 2) + k) \cdot \tau \qquad (4.2)$$

unde:
T_{ad} - timpul maxim în care sumatorul adună două numere;
k - numărul de ranguri corespunzătoare unui segment;
n - numărul total de ranguri ale sumatorului;
τ - întârzierea pe 2 niveluri logice.

De exemplu, pentru un sumator pe 20 de biți cu segmante egale, prezentat în Figura 4.3, timpul de propagare maxim este:

$$T_{ad} = (4 + (\frac{20}{4} - 2) + 4) \cdot \tau = 11\tau \qquad (4.3)$$

Se poate obține o soluție mai performantă dacă se împarte sumatorul în segmente inegale (se apelează la un artificiu). Această soluție, prezentată în Figura 4.4 pe segmente 2-5-6-5-2, are timpul de propagare:

$$T_{ad} = (6.5 + 1 + 2) \cdot \tau \qquad (4.4)$$

Concluzia este că în anumite situații este posibil ca prin divizarea numărului de ranguri al sumatorului în segmente inegale RCA, să se obțină o soluție mai performantă decât cea corespunzătoare împărțirii în segmente RCA egale.

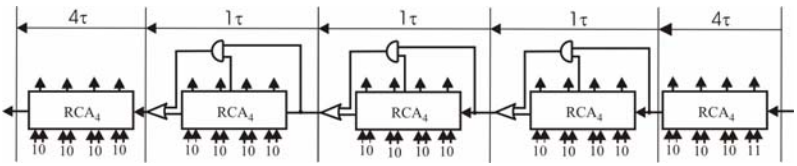


Figura 4.3 Situația cea mai defavorabilă pentru întârzierea celui mai semnificativ carry pentru sumatorul CSkA pe 20 biți cu segmente egale.

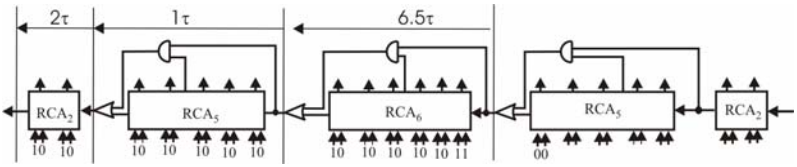


Figura 4.4 Situația cea mai defavorabilă pentru întârzierea celui mai semnificativ carry pentru sumatorul CSkA pe 20 biți cu segmente inegale 2-5-6-5-2.

3 Comparația Sumatoarelor

O comparație calitativă între sumatoarele prezentate în aceste lucrări de laborator este prezentată în Tabela 4.1.

Type	Space	Time
RCA	$O(n)$	$O(n)$
RCA	$O(n \log n)$	$O(\log_2 n)$
CSkA	$O(n)$	$O(n)$
CSeA	$O(n)$	$O(n)$

Tabela 4.1 Comparație a sumatoarelor studiate din punct de vedere al complexității temporale şi spațiale.

4 Aplicații

Problema 4.1
Se dă un sumator Carry-Skip-Adder (CSKA) pe 8 biți, segmente 2-4-2. Se cere întârzierea maximă ce apare pe biții sumei ($a_7a_6...a_0 + b_7b_6...b_0 = S_7S_6...S_0$) pentru următoarele cazuri particulare:

a) $A = a_7a_6...a_0 = 11110101$
$B = b_7b_6...b_0 = 01011111$

b) $A = 01110011$
$B = 11100110$

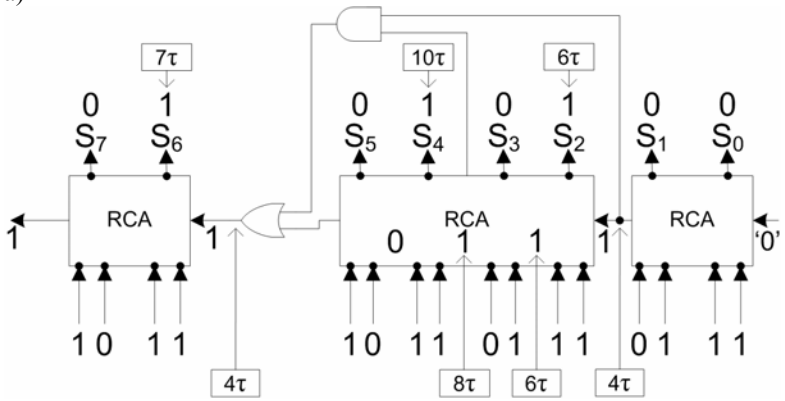
c) $A = 10100011$
$B = 10111011$

Observație: Întârzierea pe porțile AND și OR este de τ, iar cea pe porțile XOR de 2τ, iar întârzierile se consideră a fi perfecte (i.e. durata lor nu are fluctuații).

Rezolvare:

Tehnologia folosită pentru implementarea CSKA este CMOS, prin urmare – inițial – semnalele de carry și de sumă sunt pe '0' logic. Astfel, toate semnalele de sumă care sunt '0' nu au întârziere. Întârzierea se calculează doar pentru semnalele de sumă $S_7...S_0$ care sunt ' 1'.

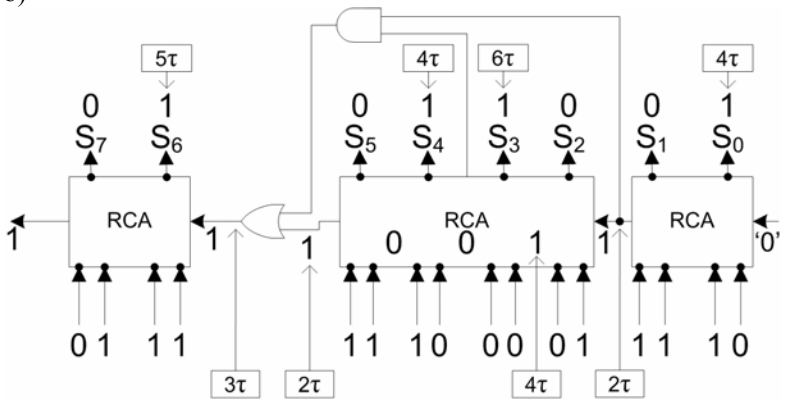
a)



În această situație se activează bucla de anticipare a carry-ului de pe segmentul RCA pe 4 ranguri binare.

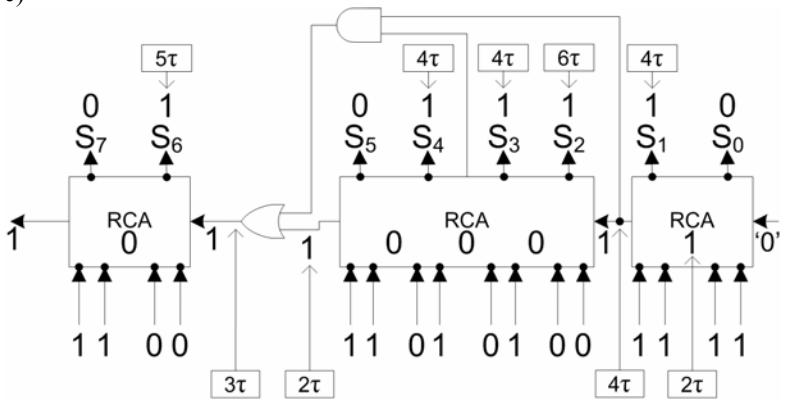
Întârzierea maximă $\rightarrow$ bitul S_4 cu valoarea 10τ.

b)



În această situație bucla de anticipare a carry-ului este blocată de perechea $a_3 b_3 = 00 \Rightarrow$ Înârzierea maximă este pe bitul S_3 cu valoarea 6τ.

c)



În această situație, bucla de anticipare a carry-ului este blocată de perechea $a_2 b_2 = 00 \Rightarrow$ Întârzierea maximă este pe bitul S_2 cu valoarea 6τ.

Problema 4.2 (propusă)
În Figura 4.4 este prezentat cazul cel mai defavorabil pentru întărzierea carry-out-ului sumatorului pe 20 de biți. Pentru același sumator CSKA pe 20 de biți cu segmante 2-5-6-5-2, să se prezinte cazul cel mai defavorabil pentru un bit al sumei şi să se calculeze această întârziere în termeni de τ.

Problema 4.3 (propusă)
Se consideră un sumator CSKA pe 24 de biți cu segmente inegale. Care soluție este mai bună din punct de vedere al întărzierii maxime a carry out-ului: 4-5-6-5-4 sau 3-4-5-5-4-3? Justificați alegerea, prin estimarea întârzierilor în termeni de τ.