4.6. Aritmetikai áramkörök

Azokat a digitális áramköröket, amelyek számtani műveletek (pl. összeadás és kivonás) végzésére alkalmasak ***aritmetikai áramköröknek*** nevezik. Az aritmetikai áramkörök bemenetét képező számokat megfelelő bináris kódban kifejezve kell megadni. Az eredményként kapott számok ugyanabban a kódban adódnak.

4.6.1. Bináris összeadó áramkörök

A bináris összeadás két egybites szám összeadására vezethető vissza. Az A és B, két egybites szám összeadását a következő szabály (*4.11. táblázat*) szerint lehet elvégezni:

				C		*(átvitel)*
A	+	B	=		S	*(összeg)*
0	+	0	=	0	0	
0	+	1	=	0	1	
1	+	0	=	0	1	
1	+	1	=	1	0	

4.11. táblázat. Két bináris szám összeadása

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

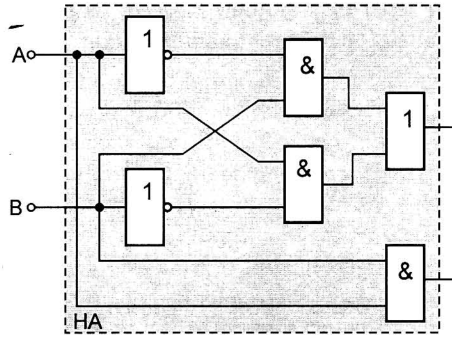
4.12. táblázat. Félösszeadó igazságtáblázata

Ha a 0 számjegyet a bináris **0**, az 1 számjegyet a bináris **1** állapothoz rendeljük, akkor a *4.12. táblázat* szerinti igazságtáblázat adódik. Az ***S*** összeg és ***C*** átvitel kifejező logikai függvények segítségével is:

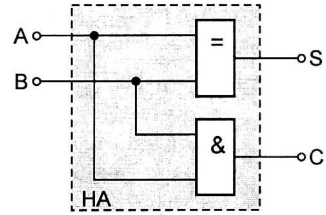
$$S = (A \cdot \overline{B}) + (\overline{A} \cdot B) = A \oplus B \qquad (4.1)$$

$$C = A \cdot B$$

A két logikai függvényt a *4.74. ábra* szerint megvalósító kapcsolást ***félösszeadónak*** nevezik. A félösszeadó (amely két bináris számot tud összeadni), a legegyszerűbb aritmetikai áramkörnek tekinthető.



a) alapkapukból felépített változat



b) KIZÁRÓ-VAGY kapus változat

4.74. ábra. Félösszeadó (jelölése– ***HA***)

A félösszeadó önmagában nem alkalmas több bites számok helyértékenkénti összeadására, mert az átvitelt nem tudja figyelembe venni. Ez egy összeadási példából is látható (*4.13. táblázat*):

4.	3.	2.	1.	0.	*oszlop*
1	1	0	0		*átvitelbitek*
	1	1	1	0	} *összeadandó számok*
	1	1	0	0	
1	1	0	1	0	*eredmény*

4.13. táblázat.

A 0., 1. és 2. oszlop összeadását el lehet végezni félösszeadóval is, de a 3. oszlopét a keletkező átvitel miatt már nem.

Az átvitelt is figyelembe vevő összeadó az úgynevezett ***teljes összeadó***. Ennek működése a következő összefüggéseken alapszik (*4.14. táblázat*):

C_{i-1}	+	A_i	+	B_i	=	C_i	S_i
0	+	0	+	0	=	0	0
0	+	0	+	1	=	0	1
0	+	1	+	0	=	0	1
0	+	1	+	1	=	1	0
1	+	0	+	0	=	0	1
1	+	0	+	1	=	1	0
1	+	1	+	0	=	1	0
1	+	1	+	1	=	1	1

4.14. táblázat.

ahol A_i és B_i az *i.* oszlop összeadandó bitjei, C_{i-1} az $(i-1)$. oszlop összeadása után keletkező átvitel, S_i az összeg, C_i pedig a keletkező átvitel.

A teljes összeadót a következő logikai függvények írják le:

$$S_i = C_{i-1} \oplus P_i \qquad (4.2)$$

$$C_i = C_i' + G_i$$

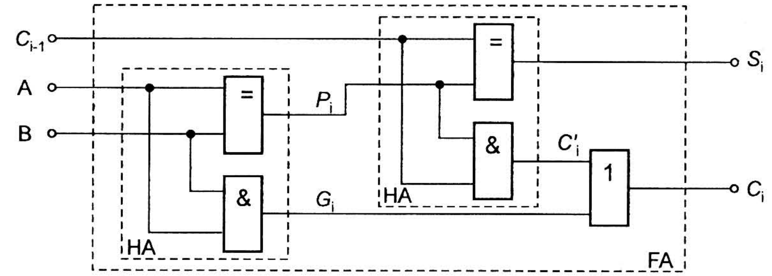
ahol

$$S_i = C_{i-1} \oplus P_i \qquad \text{a terjedő (propagate) átvitel,}$$

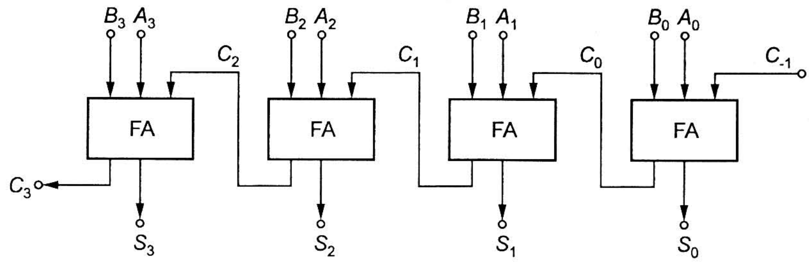
$$G_i = A_i \cdot B_i \qquad \text{a keletkező (generate) átvitel.}$$

$$C_i' = C_{i-1} \cdot P_i$$

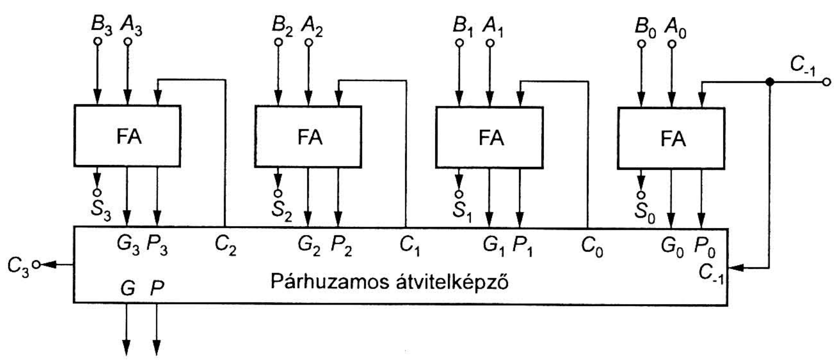
Egy teljes összeadó, amint a fenti összefüggések alapján is látható, két félösszeadóból és egy VAGY kapuból alakítható ki (*4.75. ábra*). A teljes összeadó a mikroprocesszor aritmetikai és logikai egységének alapvető építőeleme. A *4.76. ábra* egy 4-bites összeadó egység felépítését mutatja be. A több bites összeadó egység számolási ideje hosszabb mint, mint az ezt felépítő egybites teljes összeadóé. Ennek a magyarázata az, hogy a nagyobb helyértékű bitek összeadásánál várni kell az alacsonyabb helyértékű bitek átvitelképzésére. A késleltetési idők láncszerűen összeadódnak. A számlási idő lecsökkentését párhuzamos átvitelképző áramkör (angolul: *look k-ahead carry generator*) segítségével lehet elérni (*4.77. ábra*).



4.75. ábra. Teljes összeadó két félösszeadóból (jelölése ***FA***)



4.76. ábra. 4-bites teljes összeadóegység



4.77. ábra. 4-bites összeadóegység párhuzamos átvitelképzéssel

Az átvitel általános kifejezése a teljes összeadót leíró függvényekből (4.2) származik:

$$C_i = G_i + P_i \cdot C_{i-1} \tag{4.3}$$

Ennek alapján a *4.76. ábrán* bemutatott négybites összeadó egység átvitelei:

$$C_0 = G_0 + P_0 \cdot C_{-1} \tag{4.4}$$

$$C_1 = G_1 + P_1 \cdot C_0 = G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_{-1}$$

$$C_2 = G_2 + P_2 \cdot C_1 = G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_1$$

$$C_3 = G_3 + P_3 \cdot C_2 = G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + \\ + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot C_{-1} = G + P \cdot C_{-1}$$

ahol:

$$G = G_3 + P_3 \cdot G_2 + G_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 \quad \text{és}$$
$$P = P_3 \cdot P_2 \cdot P_1 \cdot P_0$$

A párhuzamos átvitelképző a fenti összefüggések szerint párhuzamosan állítja elő a C_0, C_1, C_2 és C_3 átviteleket, valamint a további lánckapcsoláshoz szükséges G és P jeleket.

4.6.2. Bináris kivonó áramkörök

Az 1-es és 2-es komplemens segítségével a kivonás összeadásra vezethető vissza. Két n-bites szám különbsége:

$$D = A - B \text{, vagyis } D = A + (-B) \tag{4.5}$$

ahol $(-B)$ 1-es vagy 2-es komplemensű.

A (2.6) és (2.10) egyenlőségek alkalmazásával a fenti különbség kifejezése a következőképpen írható:

$$D^{(1)} = A^{(1)} + (-B)^{(1)} = A + (2^n + -1 - B) \tag{4.6}$$

az 1-es kiegészítős számábrázolás esetén, és

$$D^{(2)} = A^{(2)} + (-B)^{(2)} = A + (2^n - B) \tag{4.7}$$

2-es kiegészítős számábrázolás esetén.

A különbség kiszámítása 2-es komplemensű számábrázolásban sokkal célszerűbb, mint 1-es komplemensű számábrázolásban. A legnagyobb helyérték összeadása után keletkező átvitelt (2^n a (4.6) és (4.7) kifejezésekben) az n-bites eredmény nem veszi figyelembe. Ezért a 2-es komplemensű számábrázolás esetében a különbség valós értékét kapjuk meg (ugyancsak kettes komplemensben). Az 1-es komplemensű ábrázolás esetében ahhoz, hogy a különbség valós értékét kapjuk meg, az eredményhez még 1-et kell hozzáadnunk. Például számítsuk ki a $6_{10} - 3_{10}$ különbség eredményét először négybites 2-es komplemensben (*4.15. táblázat*) és azután négybites 1-es komplemensben (*4.16. táblázat*).

1	1	0	0	*(átvitel)*	
0	1	1	0	*(kisebbítendő)*	$+6_{10}$
1	1	0	1	*(kivonandó)*	-3_{10}
0	0	1	1	*(eredmény)*	$+3_{10}$

4.15. táblázat. Különbség számítása négybites 2-es komplemensben

1	1	0	0	*(átvitel)*	
0	1	1	0	*(kisebbítendő)*	$+6_{10}$
1	1	0	0	*(kivonandó)*	-3_{10}
0	0	1	0		$+2_{10}$
		+	1		$+1_{10}$
0	0	1	1	*(eredmény)*	$+3_{10}$

4.16. táblázat. Különbség számítása négybites 1-es komplemensben

Az alábbi példák az ún. 2-es komplemensű aritmetika tulajdonságait szemléltetik. Végezzük el a következő összeadásokat és kivonásokat:

$$+4_{10} + 3_{10},\ +4_{10} - 3_{10},\ +3_{10} - 4_{10} \text{ és } -3_{10} - 4_{10}.$$

A két legnagyobb helyértékű bit összeadásánál keletkező átvitelt (C_{n-2} és C_{n-1}) feljegyeztük, ugyanis a továbbiakban ennek jelentősége van:

C_{n-1}	C_{n-2}				
0	**0**				
	0	1	0	0	$+4_{10}$
	0	0	1	1	$+3_{10}$
	0	1	1	1	$+7_{10}$

C_{n-1}	C_{n-2}				
1	**1**				
	0	1	0	0	$+4_{10}$
	1	1	0	1	-3_{10}
	0	0	0	1	$+1_{10}$

C_{n-1}	C_{n-1}				
0	**0**				
	0	0	1	1	$+3_{10}$
	1	1	0	0	-4_{10}
	1	1	1	1	-1_{10}

C_{n-2}	C_{n-2}				
1	**1**				
	1	1	0	1	-3_{10}
	1	1	0	0	-4_{10}
	1	0	0	1	-7_{10}

Végül végezzük el a következő műveleteket: $+3_{10} + 6_{10}$ és $-3_{10} - 6_{10}$. Az eredmény ábrázolására nem elég 4 bit. Ezekben az esetekben túlcsordulás keletkezik, és az eredmény csak akkor helyes, ha a legnagyobb helyértéknél keletkező C_{n-1} átvitelt is figyelembe vesszük:

C_{n-1}	C_{n-2}				
0	**1**				
	0	0	1	1	$+3_{10}$
	0	1	1	0	$+6_{10}$
0	1	0	0	1	$+9_{10}$

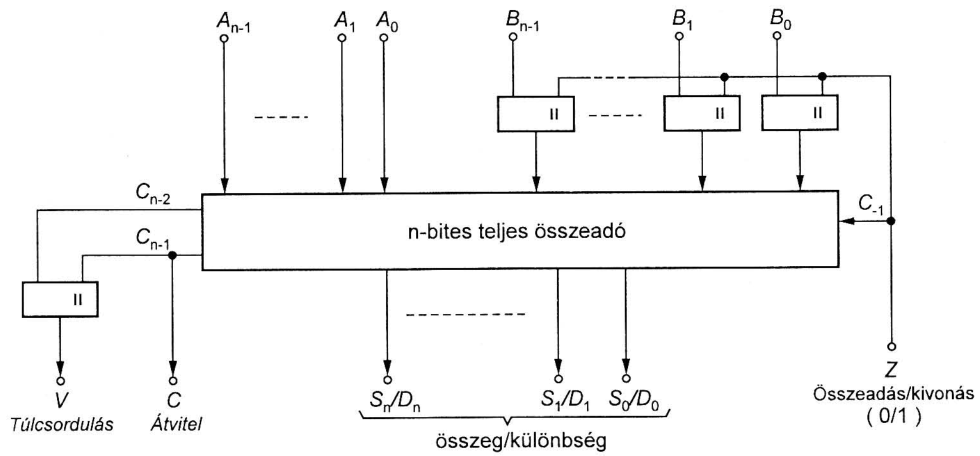
C_{n-1}	C_{n-2}				
1	**0**				
	1	1	0	1	-3_{10}
	1	0	1	0	-6_{10}
1	0	1	1	1	-9_{10}

Így az eredményt nem n-bites számként értelmezzük, hanem (n+1)-bitesként (ebben az esetben 5-bitesként). A *túlcsordulás* (angolul:*owerflow*) keletkezése a C_{n-2} és C_{n-1} átvitelek segítségével mutatható ki.

Abban az esetben, ha $C_{n-2} = C_{n-1}$, akkor nem keletkezik túlcsordulás. Viszont, ha $C_{n-2} \neq C_{n-1}$, akkor túlcsordulás keletkezik. Tehát a *V* túlcsordulást a következő *KIZÁRÓ-VAGY* függvény jelzi:

$$V = C_{n-2} \oplus C_{n-1} \tag{4.8}$$

A *4.78. ábra* egy *n*-bites összeadó/kivonó egységet mutat be. Az áramkör alapja egy *n*-bites teljes összeadó. Az *A* bináris szám $A_{n-2},...A_1$ és A_0 bitjei közvetlenül az összeadó egyik bemeneteire, míg a *B* szám $B_{n-1},...B_1$ és B_0 bitjei a *KIZÁRÓ-VAGY* kapukon keresztül az összeadó másik bemeneteire vannak csatolva. Ha egy *KIZÁRÓ-VAGY* kapu egyik bemenete logikai **0**, akkor a kimenete azonos a másik bemenetével; és abban az esetben, ha az egyik bemenete logikai **1**, akkor a kimenete a másik bemenetének komplemense. Így, ha az összeadás/kivonás vezérlő bemenet Z = **0**, akkor az egység összeadóként működik ($S = A + B$), egyébként, ha Z = **1**, akkor az egység kivonóként működik ($D = A - B$). Az utóbbi esetben a *KIZÁRÓ-VAGY* kapuk az 1-es komplemenst képezik, amely a $C_{-1} = 1$ átvitel hozzáadásával válik 2-es komplemensűvé.



4.78. ábra. *n*-bites kettes komplemensű összeadó-kivonó egység

Összefoglaló kérdések és feladatok:

1. Fogalmazza meg mi a különbség a multiplexer és a demultiplexer között!
2. Mit nevezünk analóg multiplexernek?
3. Mit nevezünk regiszternek és milyen regisztertípusok vannak?
4. Hogyan lehet megvalósítani egy párhuzamos beírású és kiolvasású léptetőregisztert?
5. Milyen szempontok szerint lehet csoportosítani a számláló áramköröket és milyen típusok vannak?
6. Milyen jellegzetességei vannak a bináris számláló áramköröknek?
7. Rajzolja fel egy szinkron bináris számláló kapcsolását és elemezze a működését! Szerkessze meg az állapotdiagramját!
8. Milyen lehetőségek vannak többhelyértékű (nagyobb mint 4) bináris számlálók realizálására?
9. Milyen tulajdonságokkal rendelkeznek a reverzibilis számlálók?
10. Miben különböznek a decimális számláló áramkörök a bináris változatoktól?
11. Mit nevezünk gyűrűs számláló áramköröknek és milyen változatai vannak?
12. Mire alkalmasak az aritmetikai áramkörök?
13. Rajzolja le egy félösszeadó áramkör kapcsolási rajzát és írja fel a kimeneti logikai függvényeket!
14. Mit nevezünk teljes összeadó áramkörnek?
15. Hogyan lehet megvalósítani a bináris kivonást digitális áramkörökkel?

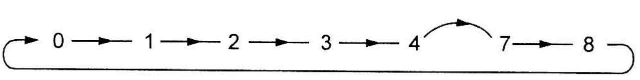
1. feladat:

Három darab *D* tárolóból felépített léptetőregisztert (D_0, D_1, D_2) $D_0 = Q_1 \oplus Q_2$ visszacsatolással látunk el.

a) Készítse el az áramkör logikai rajzát!
b) Szerkessze meg az állapotdiagramot!
c) Határozza meg annak a hálózatot kiegészítő kapcsolásnak a logikai függvényét, amelynek a hálózathoz kapcsolásával az áramkör csak 4 különböző állapotot vesz fel!

2. feladat:

Tervezzen olyan BCD kódú számlálót J-K flip-flopok felhasználásával, amelynek számlálási ciklusa a *4.79. ábrán* látható!



4.79. ábra. Számlálási ciklus

a) Írja fel a tervezéshez szükséges igazságtáblázatot!
b) Olvassa ki az állapotábrából a J-K bemenetek vezérlési függvényeit és rajzolja le a kapcsolást!