

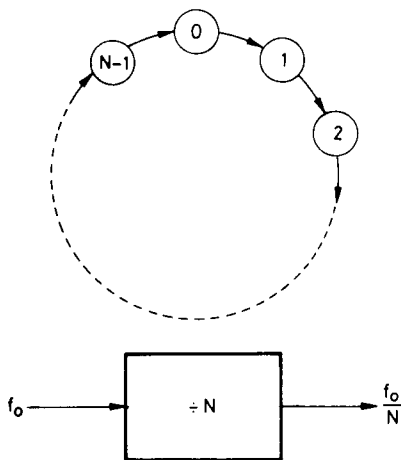
1.13. ábra.

Aszinkron MODULO-N számlálók (N-es frekvencia osztók)

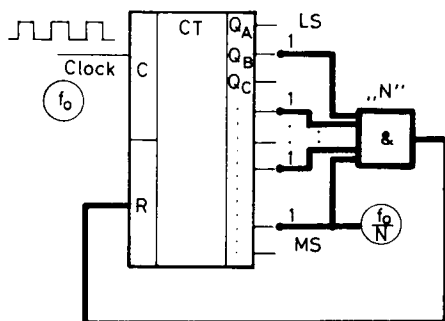
Digitális berendezésekben igen gyakori feladat, hogy egy adott (f_0) frekvenciájú jelből egy adott N-nel leosztott frekvenciájú jelet kell előállítanunk (f_0/N). Ehhez legegyszerűbb, ha egy N állapotú számlálót építünk fel, amelynek bemenetére a leosztandó frekvenciájú jelet adjuk. Tekintve, hogy a számláló N állapotonként ér egy-egy ciklus végére, és kezd egy újabb ciklust, vagyis N számú órajelenként "ismétli önmagát", ez a ciklus-ismétlődési frekvencia (amelyet a számláló MSB kimenetéről vehetünk le) lesz a kívánt N-nel osztott frekvencia. Hasonlóan gyakori feladat, hogy valamely, számlálót tartalmazó digitális áramkörben a számlálási ciklust adott értékre kell csökkenteni (maga a BCD számláló is jó példa erre: egy 4 db flip-flopot tartalmazó számlálót, amely 16 állapotot vehet fel, 10 állapotúra "rövidítünk"). A "modulo-N" elnevezés egyébként arra utal, hogy a lehetséges állapotok száma N, természetesen a zérust is beleértve. Ebből következik, hogy egy adott kódban történő számlálás esetén, a kimeneteken megjelenő legnagyobb szám N-1, ezután nulla következik, és kezdődik előlről a ciklus (1.14. ábra). Az N számot, a modulust tehát sohasem érjük el; a számlálás az N intervallum-határon belül lévő legnagyobb számig tart.

Az aszinkron megoldás alapvető jellemzője az egyszerűség, hátrány viszont, hogy felléphetnek az aszinkron működésből, a flip-flopok nem egyidejű billenéséből eredő hibák. A legtöbbször alkalmazott, legegyszerűbb megvalósítási elvet, amely felfelé számlálóhoz (előreszámlálóhoz) megfelelő, az 1.15. ábra

mutatja: egy ÉS-kapuvál "figyeljük", hogy a számláló mikor éri el a kiválasztott N számot. Abban a pillanatban, amikor a számláló kimenetén az N-nek megfelelő kombináció áll elő, az ÉS-kapu "jelzést ad" és nullázza (reset-eli) a számlálót. Más szóval az ÉS-kapuvál "kikapuzzuk" az N számot és ezzel állítjuk kezdeti, zérus állapotba a teljes számlálót. Az N szám kikapuzása alatt azt értjük, hogy az ÉS-kapu bemeneteit azokhoz a Q kimenetekhez vezetjük, amelyekben az adott N szám elérésekor 1-es szint jelenik meg, így az ÉS-kapu kimenetén is akkor jelenik meg 1-es, ami a RESET bemenetet aktiválja. A kapu természetesen lehet NEM-ÉS (NAND) is, ha a RESET 0 szintre működik, vagy ha még egy invertert iktatunk közbe.



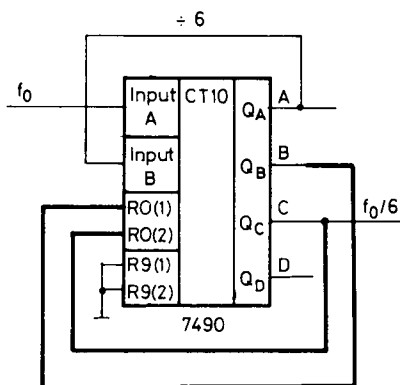
1.14. ábra.



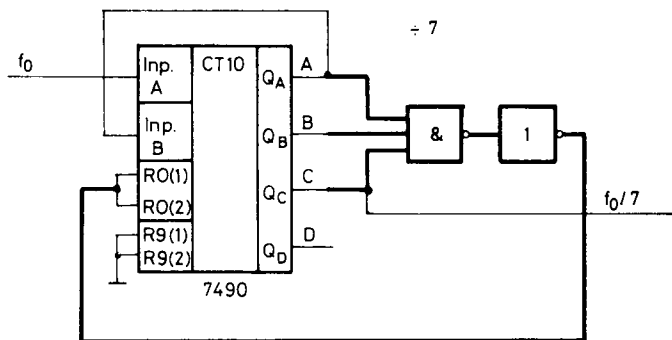
1.15. ábra.

Példaként készítsünk 6-os frekvencia osztót! Az előbbiek szerint ebben az esetben olyan felfelé számlálót kell választanunk, amelynek legalább 6, vagy ennél több lehetséges állapota van, például alkalmas akár a 7493-as bináris számláló, akár a 7490-es dekádszámláló típus. Bármelyiket is választjuk, a teendőnk az, hogy a bináris (BCD) 6-os számot kikapuzzuk és a RESET-re vezessük. Tekintve, hogy a 6-os számnak az A, B, C, D kimeneten 0 1 1 0 kombináció felel meg, olyan ÉS kapura van szükségünk, amely akkor ad 1-et a kimenetén, amikor B és C logikai 1-en van. Ez a kétbemenetű ÉS-kapu lehet maga a számlálóba beépített kétbemenetű RO kapu is, hiszen RESET akkor

következik be, amikor RO(1) és RO(2) is 1-es. Nincs más dolgunk tehát, mint a B, ill. C kimenetet az RO(1), ill. RO(2) bemenetre visszavezetni, amit a 7490-es esetére az 1.16. ábra mutat. Abban az esetben, ha kettőnél több bemenetű ÉS-kapura van szükség - pl. 7-es osztó esetében - természetesen külső, külön beépített áramkört, legtöbbször NAND kaput és invertert alkalmazunk (1.17. ábra). (A kapcsolási rajz 7493-ra is hasonló, csak az R9 bemenetek nélkül.)



1.16. ábra.



1.17. ábra.

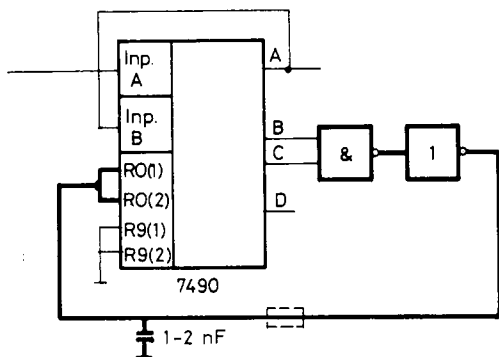
Tisztázandó kérdés az ilyen módon felépített N-es osztókkal kapcsolatban, hogy az N szám kapuzásakor miért csak azokra a kimenetekre csatlakozunk, amelyek az adott számnál logikai 1 szinten vannak, a 0 szinten lévő kimenetek negáltját miért nem vezetjük az ÉS-kapuba - amint azt általában kapuzási feladatoknál szokásos (azaz pl. a 6-os számot miért nem az

\overline{ABCD} kapcsolattal választjuk ki?). A válasz: tekintve, hogy felfelé számlálót használunk, felesleges lenne a nulla kimenetek negáltjának bevonása a kapuzásba, mivel a zérusról történő, bináris sorrendű felfelé számolásnál a kiválasztott 1-es kombináció először a kiválasztott N számnál lép fel. Ha megnézzük a bináris (BCD) felfelé számláló "kódtáblázatát", vagyis a CLOCK jelenként egymás után felvett állapotok táblázatát, láthatjuk, hogy pl. a B és C kimenet legelőször a 6-os számnál lesz egyszerre 1-es állapotú, nincs tehát szükség a nullákra; a B és C kimeneten együttesen bekövetkező 1-es egyértelműen kijelöli a 6-os számot. Tekintve, hogy ezzel a RESET feltétel teljesül, a számláló ismét 0000-ra ugrik, újabb 6-os szám eléréséig nem következik be ez az állapot. Általában bármely N-es osztó esetében bináris, BCD stb. felfelé számláló alkalmazásakor az előbbiek értelmében tehát elegendő az N számban lévő 1-esek "összegyűjtése".

	D	C	B	A	
0	0	0	0	0	←
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	RESET

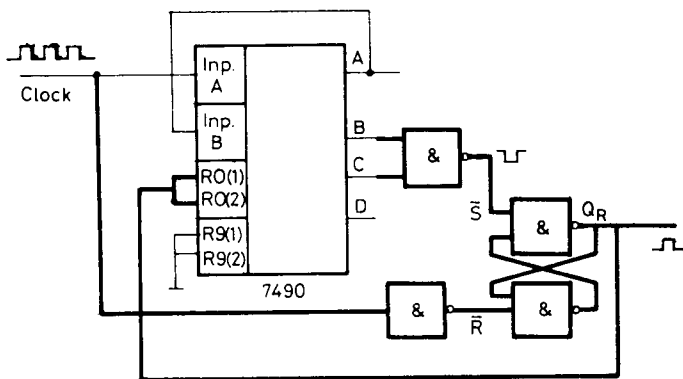
Pelmerülhet a kérdés; ha az N számot kapuzzuk ki, akkor végülis az így "visszacsatolt" számlálónak nem lesz-e N+1 állapota, hiszen a zérus is egy állapot, és N-ig számolunk el? A válasz, amely az aszinkron modulo-N számlálók működésének megértése szempontjából kulcsfontosságú: abban a pillanatban, ahogy a számláló eléri a kiválasztott N számot, a kimeneteket figyelő ÉS-kapu azonnal "jelez" és nullázva a számlálót, azaz a kimenéteken az N szám csak egy alig észlelhető rövid ideig áll elő, helyette azonnal a 0000 állapot "ugrik be". Az utolsó, teljes clock időtartamig előálló szám az N-1, és az ezután érkező órajelre már a 0000 állapotot észleljük, majd előlről kezdődik a számolási ciklus, vagyis valóban N az összes állapotok száma! Könnyű tehát megjegyezni a szabályt:

Az aszinkron nullázásnak, ciklus rövidítésnek előbbi módon való megoldása bizonyos hibalehetőséget is rejt magában. Lehetséges, hogy egy elvileg jól megtervezett áramkör - különösen, ha nagyobb számról, hosszabb osztásról van szó - nem működik az előre meghatározott osztásviszonnyal, hanem legtöbbször kisebbel. Ennek az a magyarázata, hogy legtöbbször még az integrált áramkörök flip-flopjai sem teljesen egyformák, vagy pedig külső terhelésük nem egyforma. Emiatt a RESET hatására a 0-ba billenés nem megy azonos idő alatt végbe a számláló minden tagjánál. A 6-os osztó (l. a 1.16. ábrát) példájánál maradva előfordulhat, hogy a 6-os szám megjelenésekor a létrejött RESET feltétel következtében ($R01 = R02 = 1$) a B és C flip-flop közül az egyik gyorsabban billen vissza nulla állapotba. Ebben a pillanatban azonban a RESET feltétel már nem teljesül, mivel a kimeneten már nem a 6-nak megfelelő kombináció van. A lassabb flip-flop viszont még nem billent vissza C állapotba, de a RESET megszűnése miatt már nem is fog. Más szóval, ha a flip-flopok RESET hatására történő visszabillenési ideje nem egyforma, egyes flip-flopok hamarabb billennek vissza, emiatt a RESET-jel "elmulik", a lassabbak már nem kapnak elegendően hosszú jelet a visszabillenéshez. Ezért a számláló nem zérusra ugrik vissza, hanem valamely közbelső állapotra, a ciklus N-nél rövidebb lesz. Mindez előre nehezen meghatározható, ha a hiba bekövetkezett, oka méréssel nehezen de-



21

ríthető fel. "Átmeneti megoldás"-ként egyes áramkörökben a RESET vezérlő ÉS-kapu (NAND kapu + inverter) kimenetére, vagyis a számláló RESET bemenetére egy "késleltető", "tartó" kondenzátort helyeznek el, amely a RESET-jelet hosszabbra "nyújtja", ami már biztosítja valamennyi flip-flop visszabillenését (1.18. ábra). Ennél jobb megoldás, ha a RESET jelet tároljuk, így a számláló minden flip-flopja elegendően hosszú nullázó jelet kap. A tárolás legegyszerűbben bistabil flip-floppal valósítható meg, ez a leginkább használatos. A RESET-jelet természetesen úgy kell tárolni, hogy a számlálást ne akadályozza. Az 1.19. ábra 6-os osztó áramkörében az RS flip-flopot

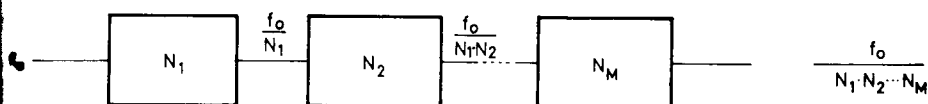


1.19. ábra.

NAND kapukkal valósítottuk meg. Amikor a RESET feltétel teljesül, azaz amikor a számláló éppen eléri a 6-os (0110) számot, akkor a kapuzó NAND áramkör az \bar{S} bemenetre 0 szintet ad, miáltal a flip-flop 1-be billen, $Q_R = 1$ lesz. Az \bar{S} jel ugyan nagyon rövid időn belül "elmulik", de a tárolásnak köszönhetően a számláló RO bemenetére tartósan 1 szint kerül, így a nullázódás üzembiztosan végbemegy. Arról azonban gondoskodni kell, hogy a számlálandó órajel következő 1-0 átmenete már érvényesülhessen. Ehhez a tárolót időben vissza kell billenteni, amit legegyszerűbben az órajel negáltjával tehetünk meg. Az utolsó 1-0 átmenetkor ugyanis a CLOCK-jel 0 szintű lett, ekkor következett be a RESET. Amikor 1-be megy, vissza kell billentennünk a RESET flip-flopot, hogy a következő átmenetet már számolja a számláló. A flip-flop visszabillentéséhez az \bar{R} bemenetre 0

szint kell, ezért ide vezetjük az inverternek használt NAND kapun keresztül a CLOCK jelet. A 6-os osztó kimenetének a RESET flip-flop kimenetét használhatjuk, mert ezen jelenik meg minden 6-os ciklus végén egy "fél órajel" időre logikai 1-es.

Adott frekvencia osztást természetesen úgy is elérhetünk, hogy akár eltérő modulusu számlálókat kötünk egymás után, aszinkron kaszkád elrendezésben az 1.20. ábra vázlata szerint. Tekintve, hogy mindegyik fokozat az előző kimenetén megjelenő jel frekvenciáját osztja a saját modulusával, az eredő modulus, frekvenciaosztás a komponensek szorzatával lesz egyenlő (1. az ábrát!). Az N_1, N_2, \dots osztást előállító egységek az eddigiekben tárgyalt aszinkron modulo- N számlálók lehetnek.



1.20. ábra.

Összefoglalva: az aszinkron számlálók előnye az egyszerűség, ennek köszönhetően az alacsony ár. Hátrány viszont, hogy a flip-flopok nem egyidejűleg billennek, hanem az órajel hatása "végighullámszik" a számlálón. Emiatt valamely szám kikapuzásakor, modulo- N számlánc készítésekor ("visszacsatolással") olyan átmeneti állapotok léphetnek fel, amelyek helytelen működéshez vezetnek. Ugyancsak hátrány, hogy nehézkes programozható modulusu számlálót építeni aszinkron rendszerrel.

1.1.3. Szinkron számlálók, frekvenciaosztók

A szinkron számlálóokban - amint erre a nevük is utal - az állapot változások egyidejűleg mennek végbe, azaz a flip-flopok egyszerre, az órajellel szinkronizálva billennek, így mentesek mindazon hibától, problémától, amelyet az aszinkron számlálók flip-flopjainak nem egyidejű működése okoz. Ezzel szemben felépítésük bonyolultabb, működésük nehezebben követethető, tervezésük is nehezebb. Ennek alapvetően az a magyarázata, hogy a szinkron működés létrehozásához valamennyi flip-