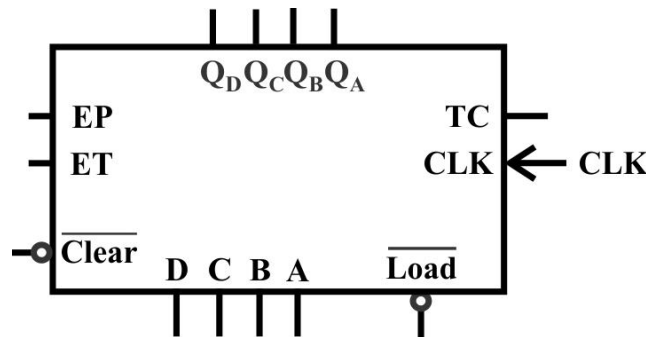


7.1. Elméleti bevezető

A feladatokat megoldása során decimális és hexadecimális számlálókkal dolgozunk. A decimális számlálók (160, 162 típusjelű) 0-tól 9-ig számolnak, a hexadecimálisak (161, 163 típusjelű) 0-F-ig számolnak. Az alábbi ábrán a 74HC161, 163 típusjelű számláló felépítését láthatjuk:



Nézzük meg a bemeneteket, és hogy mi a szerepük:

- $\overline{\text{Clear}}$: törlőbemenet. Ha erre a bemenetre alacsony szintű vezérlés kerül, akkor valamennyi kimenet (Q_A , Q_B , Q_C , Q_D) nullázódni fog. Ez a bemenet lehet szinkron (162, 163) és lehet aszinkron (160, 161) vezérlésű is. Szinkron vezérlés esetén a kimenetek, csak az órajel hatásos élére törlődnek, aszinkron vezérlés esetén az órajeltől függetlenül törlődnek, amikor ez a bemenet alacsony szintű vezérlést kap.
- $\overline{\text{Load}}$: betöltést megvalósító bemenet, amely szintén alacsony szinten aktív. Ha erre a bemenetre 0 kerül, akkor az órajel hatásos élére az A, B, C, D bemenetek értéke áttöltődik a Q_A , Q_B , Q_C , Q_D kimenetekre. Ez a bemenet mindig szinkron vezérlésű.
- EP: engedélyező bemenet. Magas szint esetén engedélyezi a számlálást, alacsony szint esetén tartás valósul meg, vagyis a számláló nem léphet a következő számlálási értékre.
- ET: engedélyező bemenet. Magas szint esetén engedélyezi a számlálást, és a TC kimenet kialakulását. Alacsony szint esetén tartás valósul meg, vagyis a számláló nem léphet a következő számlálási értékre, és a TC kimenet sem mehet 1-be.
- A, B, C, D: beíró bemenetek. Betöltés esetén, amikor a $\overline{\text{Load}}$ alacsony szintű vezérlést kap, ezeknek a bemeneteknek az értéke íródik be a megfelelő kimenetekre.
- A bemeneteknek van egy prioritási sora, ha mindegyik egyszerre kap ugyanolyan vezérlést, akkor meghatározták melyik hatása fog érvényesülni. Ez a prioritási sor a következő: 1. $\overline{\text{Clear}}$. 2. $\overline{\text{Load}}$. 3. ET. 4. EP.

Kimenetek szerepe:

- Q_A , Q_B , Q_C , Q_D : itt jelenik meg a számlálás során felvett érték.
- TC: total carry kimenet: akkor lesz magas szintű, amikor a számláló eléri a maximális számlálási értékét és az ET bemenet is magas szintű. A maximális számlálási érték decimális számláló esetén (160, 162) a 9, hexadecimális számláló esetén (161, 163) az Fh.

7.2. Feladat

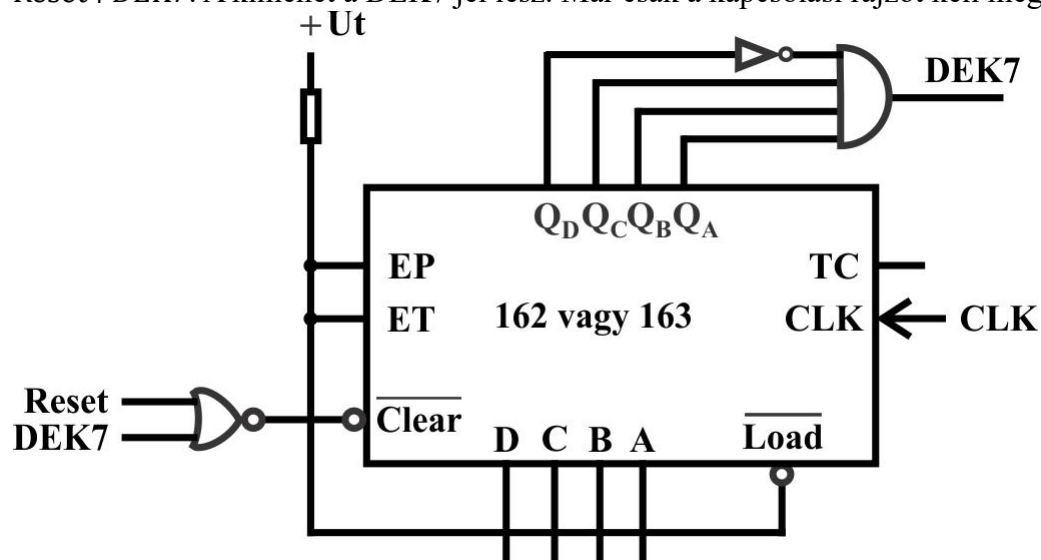
Szinkron számláló felhasználásával készítsünk olyan áramkört, amely 8 órajel periódusonként ad egy órajel periódusnyi magas szintű jelet!

Megoldás: a feladat két lehetséges megoldását adjuk meg. Az egyiknél törléssel, a másiknál betöltéssel oldjuk meg a kiírt feladatot. A törlésnél további két lehetőség lesz, az egyiknél szinkron, a másiknál aszinkron törlést valósítunk meg.

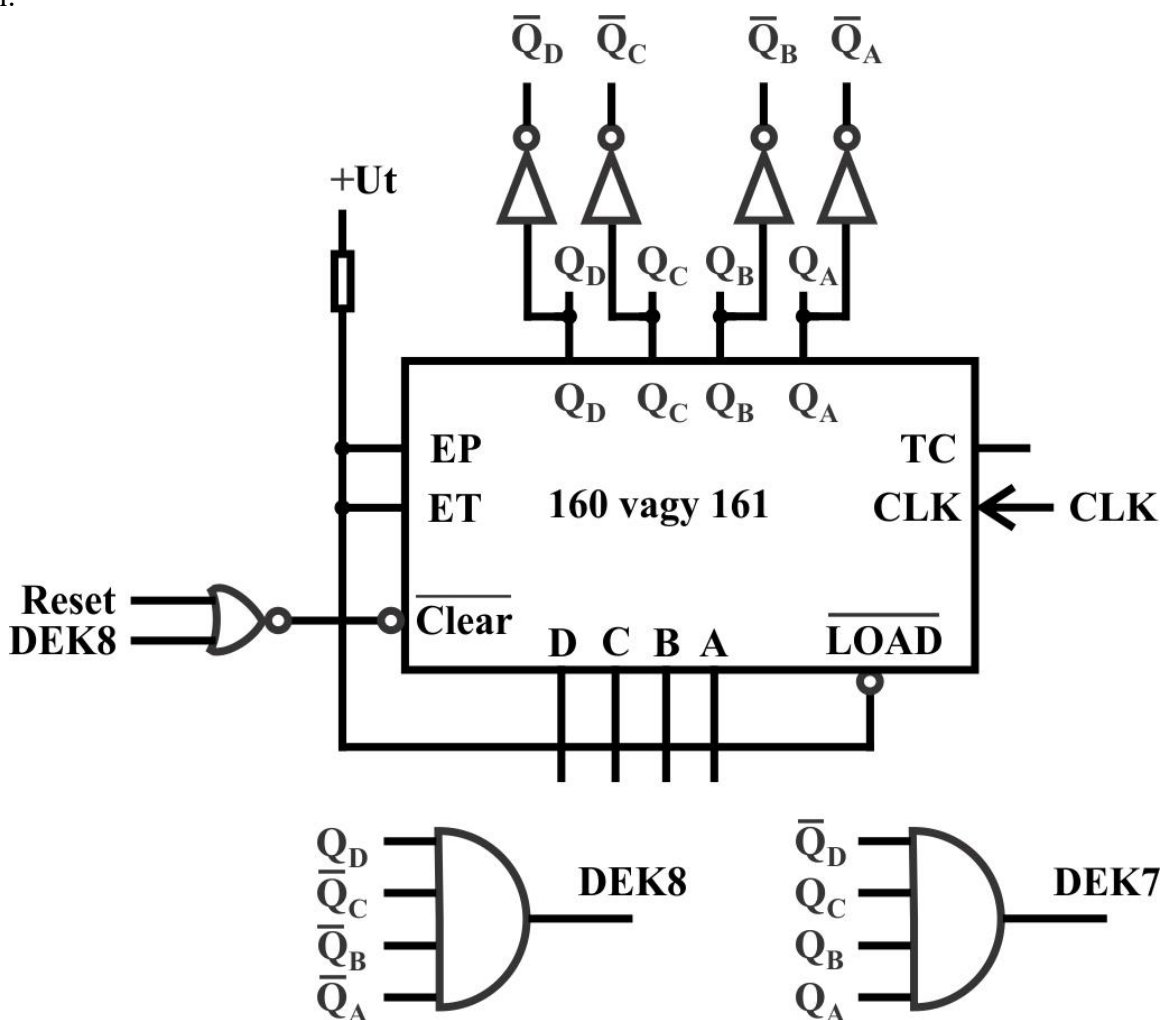
Megoldás törléssel

Nézzük a szinkron törléssel megvalósított feladatot. A 162, 163-as típusjelű számláló rendelkezik szinkron törléssel. Bármelyiket választhatjuk, mert csak 8 állapotot kell megvalósítani. Resetre törlünk és így 0-tól indul a számolás, 7-ig számolunk, majd törléssel 0-tól indulunk újra. Mivel szinkron törlés van, ezért kell 7-ig számolni, és utána újra 0-tól indulunk. Mivel 7 után törölni kell, ezért előállítunk egy DEK7 jelet, amely akkor lesz 1-es, ha a kimeneten előáll a 7-es érték. Olyan kaput keresünk, amelynek egyszer van a kimenetén 1-es. Ilyen az ÉS kapu, amelynek akkor van a kimenetén 1, ha valamennyi bemenete 1-es. Ezt figyelembe véve a dekódolt jel a következő lesz: $DEK7 = \overline{Q_D} * Q_C * Q_B * Q_A$. A Reset és a DEK7 jel is magas szinten aktív, a törlés viszont alacsony szinten, ezért olyan kaput keresünk, amelynek ha bármelyik bemenete 1-es a kimenetén 0-t ad. Ez a VAGY-NEM kapu. Így a törlő bemenet vezérlése:

$\overline{Clear} = \overline{Reset + DEK7}$. A kimenet a DEK7 jel lesz. Már csak a kapcsolási rajzot kell megadni.

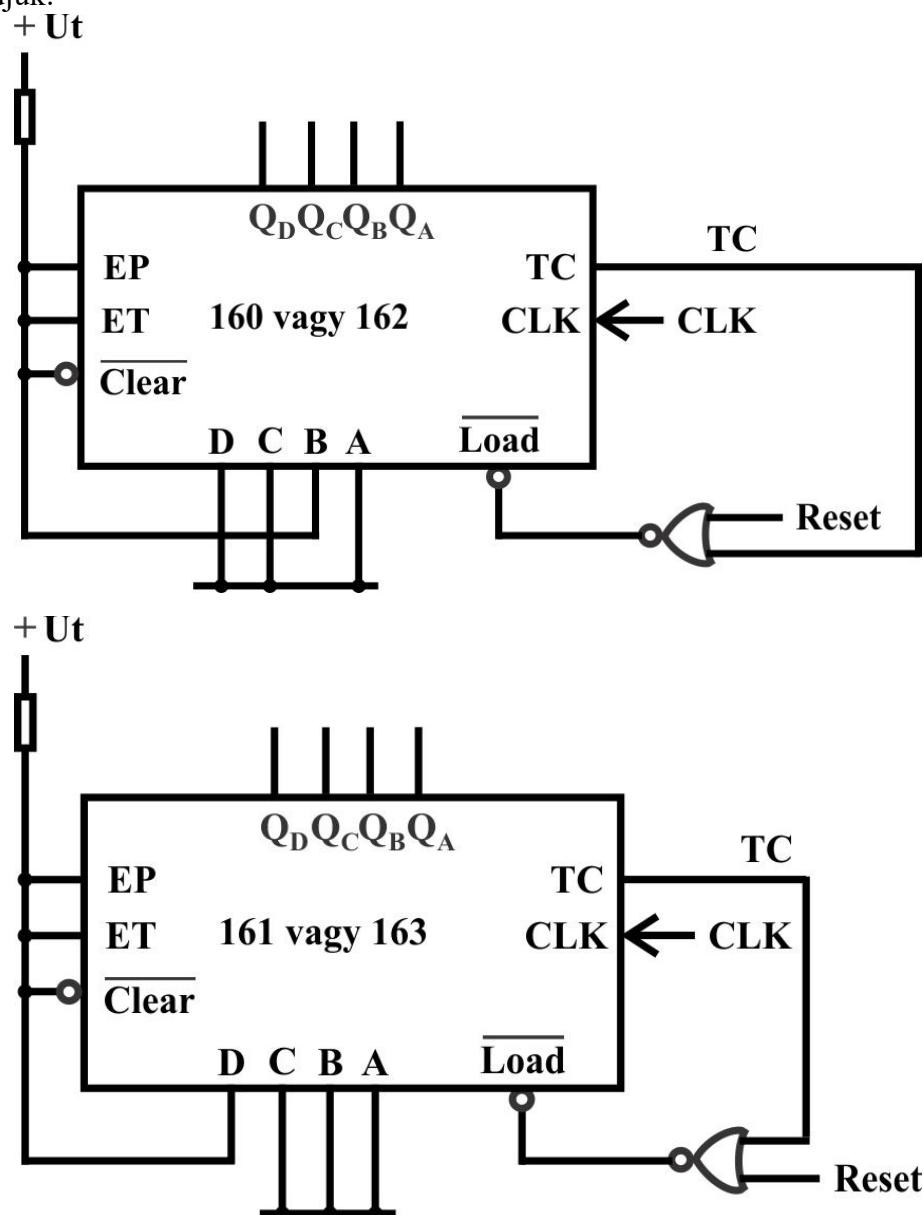


Nézzük a megoldást **aszinkron törléssel** ellátott számláló felhasználásával. Ezek a 160, 161-es típusjelű számlálók. Bármelyiket választhatjuk. Itt nem a 7 lesz a dekódolt érték, hanem a 8, mivel a számlálók aszinkron törléssel rendelkeznek, ezért így lesz meg a 8 különböző állapot. 8-nál azonnali törlés valósul meg, így egy órajel perióduson belül két állapot is lesz a 8 és a 0. Itt a 8-as értéket kell kidekódolni a törléshez. Előállítunk egy DEK8 jelet, amely akkor lesz 1-es értékű, amikor a kimeneteken előáll a 8-as érték. Olyan kaput keresünk, amelynek egyszer van a kimenetén 1-es. Ilyen az ÉS kapu, amelynek akkor van a kimenetén 1, ha valamennyi bemenete 1-es. Ezt figyelembe véve a dekódolt jel a következő lesz: $DEK8 = Q_D * \overline{Q_C} * \overline{Q_B} * \overline{Q_A}$. A Reset és a DEK8 jel is magas szinten aktív, a törlés viszont alacsony szinten, ezért olyan kaput keresünk, amelynek ha bármelyik bemenete 1-es a kimenetén 0-t ad. Ez a VAGY-NEM kapu. Így a törlő bemenet vezérlése: $\overline{Clear} = \overline{Reset + DEK8}$. A kimenet itt is a DEK7 jel lesz, mert a DEK8 jel nagyon rövid ideig lesz 1-es, az aszinkron törlés miatt. $DEK7 = \overline{Q_D} * Q_C * Q_B * Q_A$. Már csak a kapcsolási rajzot kell megadni.



Megoldás betöltéssel

A másik fajta megoldása a feladatnak, ha nem törléssel, hanem betöltéssel valósítjuk meg a kívánt áramkört. Ebben az esetben kimenetként a TC kimenetet fogjuk használni. Olyan kezdőértéket kell betölteni, hogy a maximális számlálási értékig 8 órajel periódus teljen el. Ez decimális számláló esetén, ahol 9 a maximális érték, 2 lesz, hexadecimális számláló esetén, ahol Fh a maximális érték 8h érték lesz. Tehát ha 160-as, vagy 162-es típusjelű számlálót használunk, akkor Resetre és TC-re is 2-t kell betölteni, míg ugyanez 161-es, 163-as típusjelű számláló esetén a 8h lesz. Betöltés Resetre és TC-re következik be, és mivel mindkettő magas szinten aktív, ezért VAGY-NEM kapun keresztül visszük a két jelet a $\overline{\text{Load}}$ bemenetre. $\overline{\text{Load}} = \text{Reset} + \text{TC}$. Mindkét megoldás kapcsolási rajzát megadjuk:



7.3. Feladat

Szinkron számláló felhasználásával valósítsuk meg az alábbi állapotokat bejáró hálózatot, NReset→ 2h, 3h, 4h, 5h, 7h, 8h, Ah, Bh, Dh, Eh, Fh és újra 3h!

Megoldás: Kezdetnek nézzük meg, hogy melyek azok az állapotok, amelyeknél nem a számlálási sorban következő érték szerepel. Jelen példában ezek az állapotok a következők: 5h, 8h, Bh, Fh. Ezekben az állapotokban betöltést valósítunk meg, és ehhez vesszük még az NReset jelet. Itt is dekódolt jeleket állítunk elő, amelyek az NReset jelhez hasonlóan alacsony szinten lesznek aktívak. NDEK5, NDEK8, NDEKB, NDEKF jeleket állítunk elő, amelyek akkor lesznek alacsony szintűek, amikor a kimeneteken előáll az 5h, 8h, Bh, illetve Fh érték. Olyan kaput keresünk a dekódolt jelek előállításához, amelynek egyszer van a kimenetén 0. Ilyen a VAGY kapu, illetve az ÉS-NEM kapu. Válasszuk most az ÉS-NEM kaput, amelynek akkor lesz a kimenetén 0, ha valamennyi bemenete 1-es lesz. Így a dekódolt jelek a következőképpen alakulnak:

Q _D	Q _C	Q _B	Q _A	dekódolt jel
0	1	0	1	$NDEK5 = \overline{Q_D} * Q_C * \overline{Q_B} * Q_A$
1	0	0	0	$NDEK8 = Q_D * \overline{Q_C} * \overline{Q_B} * \overline{Q_A}$
1	0	1	1	$NDEKB = Q_D * \overline{Q_C} * Q_B * Q_A$
1	1	1	1	Ide használhatjuk a TC kimenet negáltját

Itt nincs törlés, csak betöltés, ezért ennek a bemenetnek a vezérlését kell megadni. A NReset jel és a dekódolt jelek is alacsony szinten aktívak, ezért a \overline{Load} bemenet vezérlésekor egy olyan kaput keresünk, amelyiknek ha bármelyik bemenete 0, a kimenetén 0-át ad. Ez az ÉS kapu. Ennek ismeretében a \overline{Load} bemenet vezérlése:

$$\overline{Load} = NReset * NDEK5 * NDEK8 * NDEKB * NDEKF$$

Most már csak az A, B, C, D bemenetek vezérlésének felírása van hátra. Ehhez írjuk fel az alábbi táblázatot:

	D	C	B	A
NReset	0	0	1	0
NDEK5	0	1	1	1
NDEK8	1	0	1	0
NDEKB	1	1	0	1
NDEKF	0	0	1	1

Az NReset és a dekódolt jelek közül az NReset az erősebb. Ezt szem előtt tartva vegyük sorra a bemeneteket. Kezdjük a D-vel. Nézzük meg mi kell NResetre a D bemenetre. A táblázatból látszik, hogy 0. Kell még valamelyik dekódolt jelre 0 erre a bemenetre? Igen kell, NDEK5-re és NDEKF-re. Mivel az NReset és a dekódolt jelek is alacsony szinten aktívak, keresünk egy olyan kaput,

amelynek, ha bármelyik bemenete 0, akkor 0-t ad a kimenetén. Ez az ÉS kapu, ezért ennek a bemenetnek a vezérlése az alábbi lesz: $D = \overline{NReset} * NDEK5 * NDEKF$. Következő a C bemenet. Nézzük NResetre mi kell erre a bemenetre. A táblázat C-re vonatkozó oszlopából látszik, hogy 0. Még az NDEK8, NDEKF jelekre kell 0. Keresünk egy olyan kaput, amelynek ha bármelyik bemenete 0, 0-t ad a kimenetén. Ilyen az ÉS kapu, így a C bemenet vezérlése: $C = \overline{NReset} * NDEK8 * NDEKF$. Vegyük a B bemenetet. NReset-re erre a bemenetre 1 kell. Még az NDEK5, NDEK8, NDEKF jelekre kell 1, ezért ezeket a jeleket ÉS-NEM kapuval fogjuk össze. Így a B bemenet vezérlése: $B = \overline{NReset} * NDEK5 * NDEK8 * NDEKF$. Már csak az A bemenet vezérlését kell megadni. Nézzük NReset-re mi kell erre a bemenetre. 0 kell, és megkeressük melyik dekódolt jelekre kell még 0. Már csak az NDEK8-ra. Az NReset és a dekódolt jelek is alacsony szinten aktívak, ezért olyan kaput keresünk, amelynek ha bármelyik bemenete 0, akkor 0-t ad a kimenetén. Ez az ÉS kapu. Az A bemenet vezérlése: $A = \overline{NReset} * NDEK8$.

Összefoglalva:

$$D = \overline{NReset} * NDEK5 * NDEKF$$

$$C = \overline{NReset} * NDEK8 * NDEKF$$

$$B = \overline{NReset} * NDEK5 * NDEK8 * NDEKF$$

$$A = \overline{NReset} * NDEK8$$

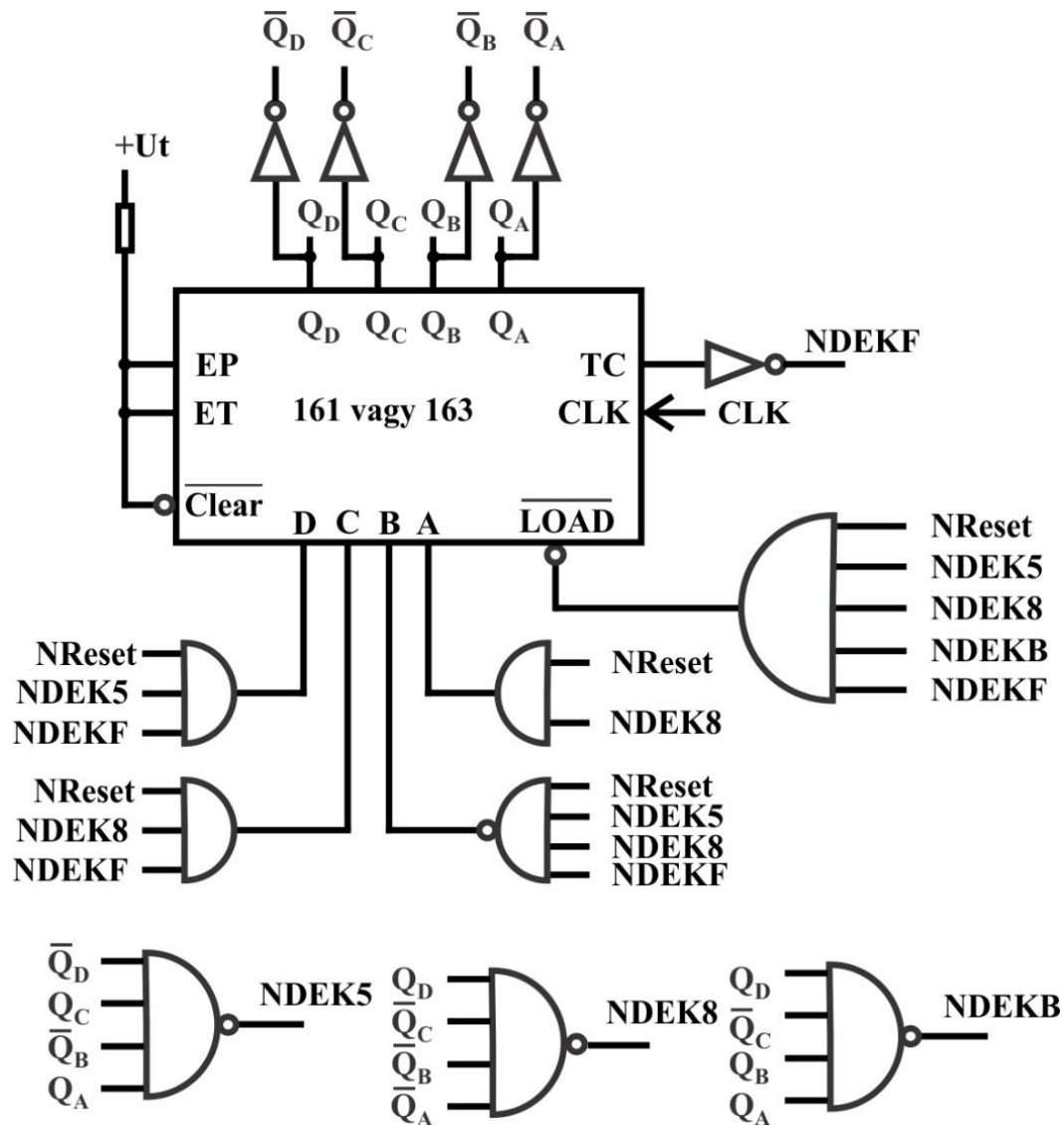
Levonható következtetések:

- Az NReset a legerősebb jel, mindig azt nézzük meg először, hogy NResetre mi kell a vizsgált bemenetre.
- A bemenetek (A, B, C, D) vezérlésének felírásakor az NReset jelnek mindig szerepelnie kell, kettő esetet kivételével. Ha az adott bemenetre minden jelre 0, vagy 1 kell, mert ilyenkor azt a bemenetet földre, vagy tápra lehet kötni.
- Ha az NReset és a dekódolt jelek is alacsony szinten aktívak, akkor ha a vizsgált bemenetre 1 kell NResetre, akkor amelyik dekódolt jelekre még 1 kell, azokat ÉS-NEM kapuval fogjuk össze.
- Ha az NReset és a dekódolt jelek is alacsony szinten aktívak, akkor ha a vizsgált bemenetre 0 kell NResetre, akkor amelyik dekódolt jelekre még 0 kell, azokat ÉS kapuval fogjuk össze.
- Ha a vizsgált bemenetre csak NResetre kell 1, akkor az adott bemenet vezérlése \overline{NReset} lesz.
- Ha a vizsgált bemenetre csak NResetre kell 0, akkor az adott bemenet vezérlése NReset lesz.

A bemenetek vezérlését más módon is felírhatjuk, ami az előzővel kompatibilis. A D, C, és az A bemenet vezérlése nem lehet egyszerűbb, de a B bemenetet egyszerűbben is felírhatjuk:

$$B = \overline{NReset} * NDEKB$$

Utolsó lépésként már csak a kapcsolási rajzot kell megadnunk. A B bemenet korábban megadott vezérlése alapján.



7.4. Feladat

Szinkron számláló felhasználásával valósítsuk meg az alábbi állapotokat bejövő hálózatot, Reset→0h, 1h, 2h, 4h, 5h, 6h, 8h, 9h, Bh, Ch, Dh, Fh és újra 0!

Megoldás: a léptetőregiszterrel megvalósított példákhoz képest, itt most nem kell működési táblát megadni. Azt kell megnéznünk, hogy melyek azok az állapotok, amelyeknél nem a számlálási sorban következő érték szerepel. Jelen példában ezek az állapotok a következők: 2h, 6h, 9h, Dh. Ezekben az állapotokban betöltést valósítunk meg. Ebben a feladatban Resetre nem betöltést valósítunk meg, hanem törlést. A törlőbemenet erősebb a betöltésnél, ezért a Reset jel prioritása biztosított lesz. Mindenesetre olyan hexadecimális számlálót használjunk, amelyiknek szinkron törlése van. Ez pedig a 163-as típusjelű. Itt is dekódolt jeleket állítunk elő, amelyek a Reset jelhez hasonlóan magas szinten lesznek aktívak. DEK2, DEK6, DEK9, DEKD jeleket állítunk elő, amelyek akkor lesznek magas szintűek, amikor a kimeneteken előáll a 2h, 6h, 9h, illetve Dh érték. Olyan kaput keresünk a dekódolt jelek előállításához, amelynek egyszer van a kimenetén 1-es. Ilyen az ÉS kapu, illetve a VAGY-NEM kapu. Válasszuk most az ÉS kaput, amelynek akkor lesz a kimenetén 1-es, ha valamennyi bemenete 1-es lesz. Így a dekódolt jelek a következőképpen alakulnak:

Q _D	Q _C	Q _B	Q _A	dekódolt jel
0	0	1	0	DEK2 = $\overline{Q_D} * \overline{Q_C} * Q_B * \overline{Q_A}$
0	1	1	0	DEK6 = $\overline{Q_D} * Q_C * Q_B * \overline{Q_A}$
1	0	0	1	DEK9 = $Q_D * \overline{Q_C} * \overline{Q_B} * Q_A$
1	1	0	1	DEKD = $Q_D * Q_C * \overline{Q_B} * Q_A$

Itt törlés, és betöltés is van, ezért mindkét bemenet vezérlését meg kell adni. Törlés csak Resetre következik be, ezért ennek a bemenetnek a vezérlése egyszerű lesz. A törlés alacsony szinten aktív, a Reset viszont magas szinten, ezért a Reset jelet negáltan kell a törlőbemenetre vezetni: $\overline{\text{Clear}} = \overline{\text{Reset}}$. Csak a dekódolt jelekre kell betölteni, és mivel ezek magas szinten aktívak, a $\overline{\text{Load}}$ bemenet pedig alacsony szinten, olyan kaput keresünk, amelynek ha bármelyik bemenete 1-es 0-t ad a kimenetén. Ez a VAGY-NEM kapu. Így a $\overline{\text{Load}}$ bemenet vezérlése: $\overline{\text{Load}} = \text{DEK2} + \text{DEK6} + \text{DEK9} + \text{DEKD}$.

Már csak a D, C, B, A bemenetek vezérlését kell megadni. Ehhez nézzük az alábbi táblázatot:

	D	C	B	A
DEK2	0	1	0	0
DEK6	1	0	0	0
DEK9	1	0	1	1
DEKD	1	1	1	1

A Reset és a dekódolt jelek közül a Reset az erősebb, és mivel ezt a jelet a törlőbemenetre kötöttük, a $\overline{\text{Load}}$ bemenet vezérlésekor csak a dekódolt jeleket kell figyelembe venni. Vegyük sorra a bemeneteket. Kezdjük a D-vel. A táblázatból látszik, hogy 0, csak DEK2-re kell, ezért ennek a bemenetnek a vezérlése a következőképpen alakul: $D = \overline{\text{DEK2}}$. Következő a C bemenet. Erre a bemenetre két dekódolt jelre kell 1-es, kettőre pedig 0. Válasszuk az 1-eseket. A dekódolt jelek magas szinten aktívak, ezért olyan kaput keresünk, amelynek ha a bármelyik bemenete 1-es, akkor 1-t ad a kimenetén. Ez a VAGY kapu. A C bemenet vezérlése ezek alapján: $C = \text{DEK2} + \text{DEKD}$. Vegyük a B bemenetet, amely megegyezik az A bemenettel. Ezekre a bemenetekre két dekódolt jelre 1, kettőre pedig 0 kell. Vegyük azokat a dekódolt jeleket, amelyekre 1 kell. Ezek a DEK9 és a DEKD. Mivel a dekódolt jelek magas szinten aktívak, ezért olyan kaput keresünk, amelyiknek ha bármelyik bemenete 1-es, akkor 1-t ad a kimenetén. Ez a VAGY kapu. Így a B és az A bemenet vezérlése a következő lesz: $B = A = \text{DEK9} + \text{DEKD}$.

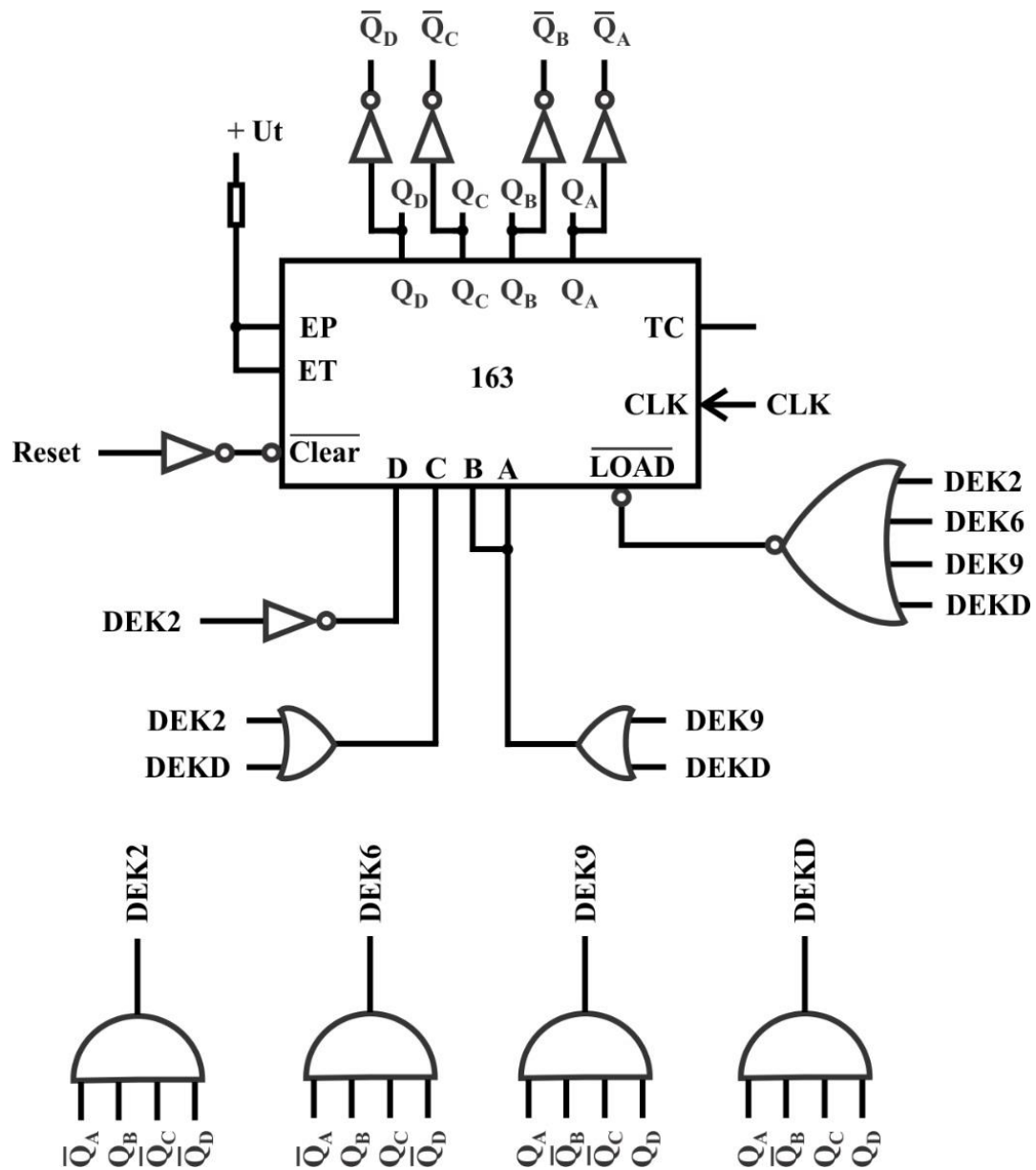
Összefoglalva:

$$D = \overline{\text{DEK2}}$$

$$C = \text{DEK2} + \text{DEKD}$$

$$B = A = \text{DEK9} + \text{DEKD}$$

Utolsó lépésként már csak a kapcsolási rajzot kell megadnunk.



7.5. Feladat

Szinkron számláló felhasználásával valósítsuk meg az alábbi állapotokat bejáró hálózatot, Reset→3h, 4h, 6h, 7h, 9h, Ah, Bh, Dh, Eh, Fh és újra 4h!

Megoldás: a léptetőregiszterrel megvalósított példákhoz képest, itt most nem kell működési táblát megadni. Azt kell megnéznünk, hogy melyek azok az állapotok, amelyeknél nem a számlálási sorban következő érték szerepel. Jelen példában ezek az állapotok a következők: 4h, 7h, Bh, Fh. Ezekben az állapotokban betöltést valósítunk meg, és ehhez vesszük még a Reset jelet. Itt is dekódolt jeleket állítunk elő, amelyek a Reset jelhez hasonlóan magas szinten lesznek aktívak. DEK4, DEK7, DEKB, DEKF jeleket állítunk elő, amelyek akkor lesznek magas szintűek, amikor a kimeneteken előáll a 4h, 7h, Bh, illetve Fh érték. Olyan kaput keresünk a dekódolt jelek előállításához, amelynek egyszer van a kimenetén 1-es. Ilyen az ÉS kapu, illetve a VAGY-NEM kapu. Válasszuk most az ÉS kaput, amelynek akkor lesz a kimenetén 1-es, ha valamennyi bemenete 1-es lesz. Így a dekódolt jelek a következőképpen alakulnak:

Q _D	Q _C	Q _B	Q _A	dekódolt jel
0	1	0	0	DEK4 = $\overline{Q_D} * Q_C * \overline{Q_B} * \overline{Q_A}$
0	1	1	1	DEK7 = $\overline{Q_D} * Q_C * Q_B * Q_A$
1	0	1	1	DEKB = $Q_D * \overline{Q_C} * Q_B * Q_A$
1	1	1	1	Ide használhatjuk a TC kimenetet

Itt nincs törlés, csak betöltés, ezért ennek a bemenetnek a vezérlését kell megadni. Resetre és a dekódolt jelekre kell betölteni, és mivel ezek magas szinten aktívak, a \overline{Load} bemenet pedig alacsony szinten, olyan kaput keresünk, amelynek ha bármelyik bemenete 1-es 0-t ad a kimenetén. Ez a VAGY-NEM kapu. Így a \overline{Load} bemenet vezérlése:

$\overline{Load} = \overline{Reset} + DEK4 + DEK7 + DEKB + DEKF$. Már csak a D, C, B, A bemenetek vezérlését kell megadni. Ehhez nézzük az alábbi táblázatot:

	D	C	B	A
Reset	0	0	1	1
DEK4	0	1	1	0
DEK7	1	0	0	1
DEKB	1	1	0	1
DEKF	0	1	0	0

A Reset és a dekódolt jelek közül a Reset az erősebb. Ezt szem előtt tartva vegyük sorra a bemeneteket. Kezdjük a D-vel. Nézzük meg mi kell Resetre a D bemenetre. A táblázatból látszik, hogy 0. Kell még valamelyik dekódolt jelre 0 erre a bemenetre? Igen kell, a DEK4 és a DEKF jelekre. Mivel mindegyik jel magas szinten aktív, olyan kaput keresünk, amelynek ha bármelyik bemenete

1-es 0-t ad a kimenetén. Ez a VAGY-NEM kapu.

Így a D bemenet vezérlése: $D = \overline{\text{Reset} + \text{DEK4} + \text{DEKF}}$. Következő a C bemenet. Nézzük Resetre mi kell erre a bemenetre. A táblázat C-re vonatkozó oszlopából látszik, hogy 0. Még a DEK7 jelre kell 0. Mivel a Reset és a dekódolt jelek is magas szinten aktívak, keresünk egy olyan kaput, amelynek, ha bármelyik bemenete 1-es, akkor 0-t ad a kimenetén. Ez a VAGY-NEM kapu. A C bemenet vezérlése ezek alapján: $C = \overline{\text{Reset} + \text{DEK7}}$. Vegyük a B bemenetet. Resetre erre a bemenetre 1 kell. Nézzük melyik dekódolt jelekre kell még 1. Csak a DEK4-re. Mivel a Reset és a dekódolt jelek is magas szinten aktívak, ezért olyan kaput keresünk, amelyiknek ha bármelyik bemenete 1-es, akkor 1-t ad a kimenetén. Ez a VAGY kapu. Így a B bemenet vezérlése a következő lesz: $B = \text{Reset} + \text{DEK4}$. Már csak az A bemenet vezérlésének megadása van hátra. Nézzük Resetre mi kell erre a bemenetre. 1-es, és megkeressük melyik dekódolt jelekre kell még 1-es. Ezek a DEK7, DEKB jelek. A Reset és a dekódolt jelek is magas szinten aktívak és most olyan kaput keresünk, amelynek ha bármelyik bemenete 1-es, akkor 1-et ad a kimenetén. Ez a VAGY kapu. Az A bemenet vezérlése ezek alapján: $A = \text{Reset} + \text{DEK7} + \text{DEKB}$.

Összefoglalva:

$$D = \overline{\text{Reset} + \text{DEK4} + \text{DEKF}}$$

$$C = \overline{\text{Reset} + \text{DEK7}}$$

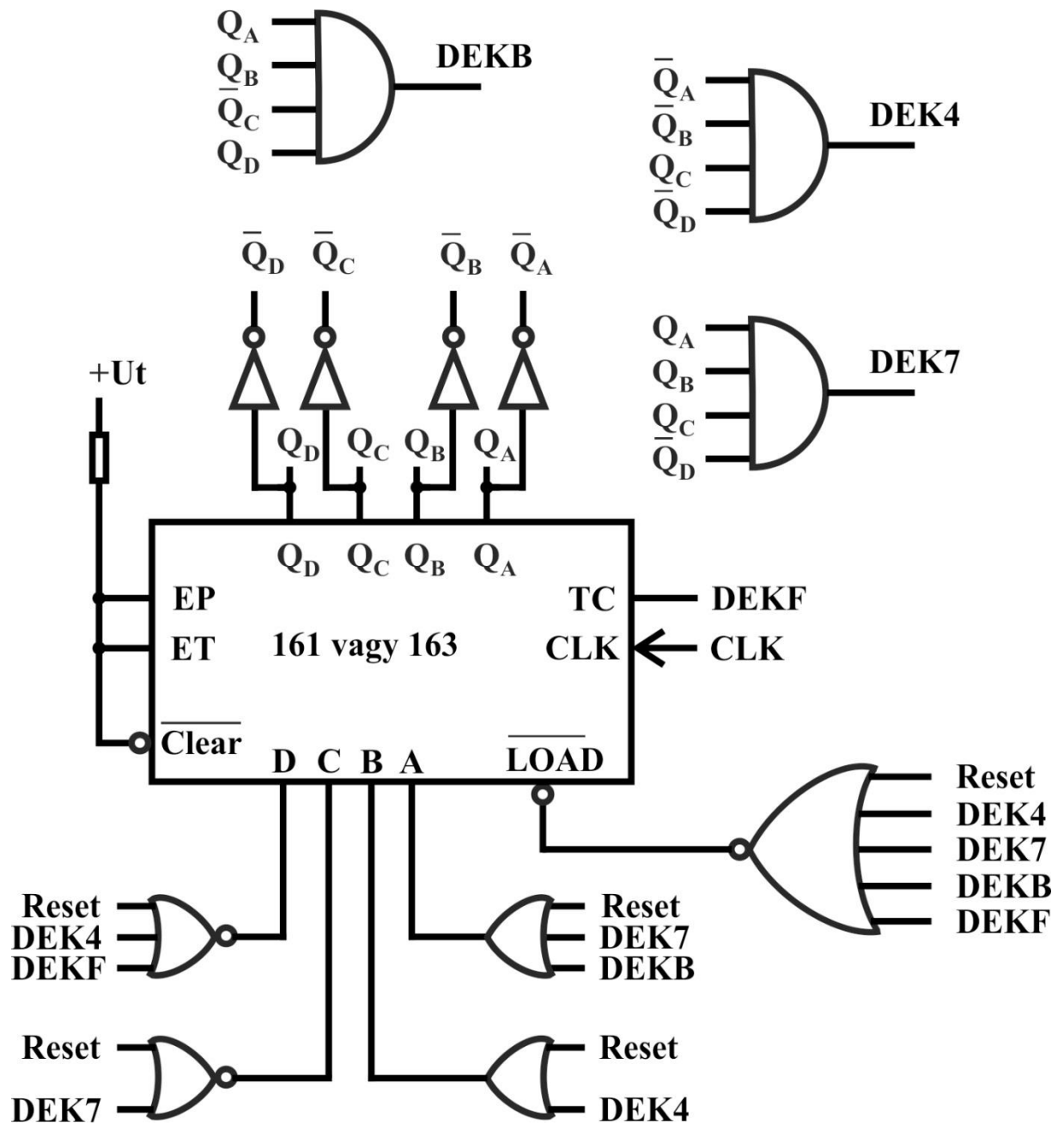
$$B = \text{Reset} + \text{DEK4}$$

$$A = \text{Reset} + \text{DEK7} + \text{DEKB}$$

Levonható következtetések:

- A Reset a legerősebb jel, mindig azt nézzük meg először, hogy Resetre mi kell a vizsgált bemenetre.
- A bemenetek (A, B, C, D) vezérlésének felírásakor a Reset jelnek mindig szerepelnie kell, kettő esetet kivételével. Ha az adott bemenetre minden jelre 0, vagy 1 kell, mert ilyenkor azt a bemenetet földre, vagy tápra lehet kötni.
- Ha a Reset és a dekódolt jelek is magas szinten aktívak, akkor ha a vizsgált bemenetre 1 kell Resetre, akkor amelyik dekódolt jelekre még 1 kell, azokat VAGY kapuval fogjuk össze.
- Ha a Reset és a dekódolt jelek is magas szinten aktívak, akkor ha a vizsgált bemenetre 0 kell Resetre, akkor amelyik dekódolt jelekre még 0 kell, azokat VAGY- NEM kapuval fogjuk össze.
- Ha a vizsgált bemenetre csak Resetre kell 1, akkor az adott bemenet vezérlése Reset lesz.
- Ha a vizsgált bemenetre csak Resetre kell 0, akkor az adott bemenet vezérlése $\overline{\text{Reset}}$ lesz.

Nincs olyan bemenet, amelynek a vezérlését egyszerűbb alakban is felírhatjuk. Utolsó lépésként már csak a kapcsolási rajzot kell megadnunk.



7.6. Feladat

Szinkron számláló felhasználásával valósítsuk meg az alábbi állapotokat bejáró hálózatot, Reset→11h-49h, 66h-84h, A3h-B7h, D0h-E9h, F4h-F9h és újra 27h !

Megoldás: Ennél a feladatnál nem lesz elég egy négybites számláló, hanem kettő kell. A kisebb helyi értéken jó lesz egy decimális számláló a nagyobb helyi értékre viszont hexadecimális kell. Ezután ugyanúgy járunk el, mint a korábbi feladatoknál, csak itt 8 adatbemenettel és 8 adatkimenettel kell számolnunk. Két 4 bites számlálót úgy kötünk össze 8 bitessé, hogy a kisebb helyi értékű számláló TC kimenetét kötjük a nagyobb helyi értékű számláló ET bemenetére. Mivel a kisebb helyi értéken 9 a legnagyobb érték, ezért ide használhatunk decimális számlálót (160, 162), a nagyobb helyi értékű számláló hexadecimális lesz (161, 163), mert itt 9h-nál nagyobb érték is előfordul. Itt is azt kell megnéznünk, hogy melyek azok az állapotok, amelyeknél nem a számlálási sorban következő érték szerepel. Jelen példában ezek az állapotok a következők: 49h, 84h, B7h, E9h, F9h. Ezekben az állapotokban betöltést valósítunk meg, és ehhez vesszük még a Reset jelet. Itt is dekódolt jeleket állítunk elő, amelyek a Reset jelhez hasonlóan magas szinten lesznek aktívak. DEK49, DEK84, DEKB7, DEKE9, DEKF9 jeleket állítunk elő, amelyek akkor lesznek magas szintűek, amikor a kimeneteken előáll a 49h, 84h, B7h, E9h, illetve F9h érték. Olyan kaput keresünk a dekódolt jelek előállításához, amelynek egyszer van a kimenetén 1-es. Ilyen az ÉS kapu, illetve a VAGY-NEM kapu. Válasszuk most az ÉS kaput, amelynek akkor lesz a kimenetén 1-es, ha valamennyi bemenete 1-es lesz. Így a dekódolt jelek a következőképpen alakulnak:

Q _H	Q _G	Q _F	Q _E	Q _D	Q _C	Q _B	Q _A	dekódolt jel
0	1	0	0	1	0	0	1	DEK49 = $\overline{Q_H} * Q_G * \overline{Q_F} * \overline{Q_E} * DEK9$
1	0	0	0	0	1	0	0	DEK84 = $Q_H * \overline{Q_G} * \overline{Q_F} * \overline{Q_E} * \overline{Q_D} * Q_C * \overline{Q_B} * \overline{Q_A}$
1	0	1	1	0	1	1	1	DEKB7 = $Q_H * \overline{Q_G} * Q_F * Q_E * \overline{Q_D} * Q_C * Q_B * Q_A$
1	1	1	0	1	0	0	1	DEKE9 = $Q_H * Q_G * Q_F * \overline{Q_E} * DEK9$
1	1	1	1	1	0	0	1	DEKF9-nél használjuk a nagyobb helyi értékű számláló TC kimenetét

Itt nincs törlés, csak betöltés, ezért ennek a bemenetnek a vezérlését kell megadni. Resetre és a dekódolt jelekre kell betölteni, és mivel ezek magas szinten aktívak, a \overline{Load} bemenet pedig alacsony szinten, olyan kaput keresünk, amelynek ha bármelyik bemenete 1-es 0-t ad a kimenetén. Ez a VAGY-NEM kapu. Így a \overline{Load} bemenet vezérlése:

$$\overline{Load} = \overline{Reset + DEK49 + DEK84 + DEKB7 + DEKE9 + DEKF9}.$$

Már csak a H, G, F, E, D, C, B, A bemenetek vezérlését kell megadni. Ehhez nézzük az alábbi táblázatot:

	H	G	F	E	D	C	B	A
Reset	0	0	0	1	0	0	0	1
DEK49	0	1	1	0	0	1	1	0
DEK84	1	0	1	0	0	0	1	1
DEKB7	1	1	0	1	0	0	0	0
DEKE9	1	1	1	1	0	1	0	0
DEKF9	0	0	1	0	0	1	1	1

A Reset és a dekódolt jelek közül a Reset az erősebb. Ezt szem előtt tartva vegyük sorra a bemeneteket. Kezdjük a H-val. Nézzük meg mi kell Resetre a H bemenetre. A táblázatból látszik, hogy 0. Kell még valamelyik dekódolt jelre 0 erre a bemenetre? Igen kell, a DEK49 és a DEKF9 jelekre. Mivel mindegyik jel magas szinten aktív, olyan kaput keresünk, amelynek ha bármelyik bemenete 1-es 0-t ad a kimenetén. Ez a VAGY-NEM kapu. Így a H bemenet vezérlése: $H = \overline{\text{Reset} + \text{DEK49} + \text{DEKF9}}$. Következő a G bemenet. Nézzük Resetre mi kell erre a bemenetre. A táblázat G-re vonatkozó oszlopából látszik, hogy 0. Még a DEK84, és a DEKF9 jelekre kell 0. Mivel a Reset és a dekódolt jelek is magas szinten aktívak, a megoldás hasonló a H bemenet vezérlésére. A G bemenet vezérlése tehát: $G = \overline{\text{Reset} + \text{DEK84} + \text{DEKF9}}$. Vegyük az F bemenetet. Resetre erre a bemenetre is 0 kell. Nézzük melyik dekódolt jelekre kell még 1. Csak a DEKB7-re. Az előzőeket figyelembe véve az F bemenet vezérlése: $F = \overline{\text{Reset} + \text{DEKB7}}$. Nézzük Resetre mi kell az E bemenetre. 1-es, és megkeressük melyik dekódolt jelekre kell még 1-es. Ezek a DEKB7, DEKE9 jelek. A Reset és a dekódolt jelek is magas szinten aktívak és most olyan kaput keresünk, amelynek ha bármelyik bemenete 1-es, akkor 1-et ad a kimenetén. Ez a VAGY kapu. Az E bemenet vezérlése: $E = \text{Reset} + \text{DEKB7} + \text{DEKE9}$. A D bemenetre valamennyi jelre 0 kell, ezért ezt a bemenetet földre kötjük. A C bemenetre Resetre 0 kell, ezért megkeressük mely dekódolt jelekre kell még 0, és azokat VAGY-NEM kapuval fogjuk össze. Így a C bemenet vezérlés $C = \overline{\text{Reset} + \text{DEK84} + \text{DEKB7}}$. A B bemenetre is 0 kell Resetre, és még a DEKB7 és DEKE9 jelekre is, ezért ezeket VAGY-NEM kapuval fogjuk össze. A B bemenet vezérlése: $B = \overline{\text{Reset} + \text{DEKB7} + \text{DEKE9}}$, ami E negáltjával egyezik meg. Az A bemenetre 1 kell Resetre. Nézzük mely dekódolt jelekre kell még 1-es. A DEK84 és a DEKF9 jelekre. Mivel a Reset és a dekódolt jelek is magas szinten aktívak, olyan kaput keresünk, amelynek ha bármelyik bemenete 1-es, 1-et ad a kimenetén. Ez a VAGY kapu. Ennek alapján az A bemenet vezérlése: $A = \text{Reset} + \text{DEK84} + \text{DEKF9}$, ez pedig a G bemenet negáltja lesz.

Összefoglalva:

$$H = \overline{\text{Reset}} + \text{DEK49} + \text{DEKF9}$$

$$G = \overline{\text{Reset}} + \text{DEK84} + \text{DEKF9}$$

$$F = \overline{\text{Reset}} + \text{DEKB7}$$

$$E = \text{Reset} + \text{DEKB7} + \text{DEKE9}$$

$$D = 0$$

$$C = \overline{\text{Reset}} + \text{DEK84} + \text{DEKB7}$$

$$B = \overline{E}$$

$$A = \overline{G}$$

Levonható következtetések:

- A Reset a legerősebb jel, mindig azt nézzük meg először, hogy Resetre mi kell a vizsgált bemenetre.
- A bemenetek (H, G, F, E, D, C, B, A) vezérlésének felírásakor a Reset jelnek mindig szerepelnie kell, kettő esetet kivételével. Ha az adott bemenetre minden jelre 0, vagy 1 kell, mert ilyenkor azt a bemenetet földre, vagy tápra lehet kötni.
- Ha a Reset és a dekódolt jelek is magas szinten aktívak, akkor ha a vizsgált bemenetre 1 kell Resetre, akkor amelyik dekódolt jelekre még 1 kell, azokat VAGY kapuval fogjuk össze.
- Ha a Reset és a dekódolt jelek is magas szinten aktívak, akkor ha a vizsgált bemenetre 0 kell Resetre, akkor amelyik dekódolt jelekre még 0 kell, azokat VAGY- NEM kapuval fogjuk össze.
- Ha a vizsgált bemenetre csak Resetre kell 1, akkor az adott bemenet vezérlése Reset lesz.
- Ha a vizsgált bemenetre csak Resetre kell 0, akkor az adott bemenet vezérlése $\overline{\text{Reset}}$ lesz.

Nincs olyan bemenet, amelynek a vezérlését egyszerűbb alakban is felírhatjuk. Utolsó lépésként már csak a kapcsolási rajzot kell megadnunk.

