

Ellenőrző kérdések(5. előadás)

Mit jelent a sorrendi hálózat fogalma, hogyan definiáljuk típusait?

A sorrendi hálózatok legfőbb ismérve, hogy a kimeneti kombináció nem csak a pillanatnyi bemeneti kombinációtól függ, hanem a korábbi bemeneti kombinációktól, sőt azok sorrendjétől is.

Sorrendi hálózatok típusai:

1. Mealy-típusú sorrendi hálózatok:
 - a. Mealy-típusú szinkron sorrendi hálózatok
 - b. Mealy-típusú aszinkron sorrendi hálózatok

1. Moore-típusú sorrendi hálózatok
 - a. Moore-típusú szinkron sorrendi hálózatok
 - b. Moore-típusú aszinkron sorrendi hálózatok

Mi a Mealy- és Moore-modell lényege?

A sorrendi hálózatoknak azt a típusát, amelynek kimeneti kombinációira az elsődleges változók és a szekunder változók is egyidejűleg hatnak, Mealy-típusú sorrendi hálózatnak nevezzük.

A sorrendi hálózatoknak azt a típusát, amelynek kimeneti kombinációira csak a belső állapotok hatnak, Moore-típusú sorrendi hálózatnak nevezzük.

Milyen tárolóelemeket használnak sorrendi hálózatokban?

A sorrendi hálózatokban –hálózattípustól függően- mind szinkron, mind aszinkron tárolóelemeket használhatnak.

Pl.

- szinkron tárolóelem: D-MS; JK-MS,...

aszinkron tárolóelem: DG; SR

Mit jelent az átlátszóság fogalma a DG tároló esetében, mi a master-slave elv lényege?

Az aszinkron DG tároló -felépítéséből származó- hátránya, hogy a $G=1$ helyzetben a D-re adott változások kijutnak a kimenetre. A $G=1$ helyzetben tehát a tároló a D-bemenet felől „átlátszó” (transzparens).

Az átlátszóság kiküszöbölésére alkalmas a master-slave (mester-szolga) elv alkalmazása. Ennek lényege, hogy két tároló elemet kötünk sorba egymás után, és egy központi vezérlő jellel (órajel) ütemezzük a működését. Az ilyen, két ütemben beírható tárolókat nevezzük mester-szolga (master-slave) tárolóknak, mivel az első fokozatba beírt értéket a második szolgai módon bemásolja. A MS tároló működését tehát az órajel két fázisra bontja: az első az adatbemenet mintavételezése és a mintavételezett érték tárolása, miközben a kimenet változatlan, őrzi az utolsóként beállt értéket. A második ütem a kimenetre a mintavételezett érték rákapcsolása és tárolása, miközben az adatbemenet változásai már hatástalanok maradnak.

Hogyan kell értelmezni egy hálózat (vagy tároló) működését az állapot-átmeneti tábla alapján (felépítés)?

Egy hálózat működését egyszerű megadni az állapot-átmeneti tábla (állapottábla) segítségével: a táblázatból kiolvasható hogy egy adott aktuális állapotból, egy adott bemenő érték (vagy bitkombináció) hatására milyen következő állapotba kerül a hálózat, mind a szekunder változó(k), mind a kimenet(ek) tekintetében.

A táblázat felépítése a következő:

a táblázat bal szélső oszlopban felsoroljuk az aktuális kimeneti állapotokat, a többi oszlopot pedig a bemeneti kombinációkkal jelöljük. Az összetett igazságtáblázatban szereplő, következő előírt állapotként megadott értékeket pedig bemásoljuk a táblázat megfelelő pontjaiba. A stabil állapotokat célszerű külön jelöléssel feltüntetni a táblázatban.

Mi a vezérlési tábla, miből és hogyan származtatjuk?

A vezérlési tábla tartalmazza -egy tároló szempontjából- az előírt (szekunder változó) kimeneti értékek biztosításához szükséges tároló vezérlési kódokat. A vezérlési táblát a tároló összetett igazságtáblázatából származtatjuk.

Milyen segédbemenetekkel rendelkezik egy tároló és milyen célt szolgálnak?

A kezdeti állapot beállításának céljából kétféle segédbemenettel rendelkezhet egy tároló:

- preset
- clear

A 'preset' ('Pr') bemenet a tárolót a funkcionális bemenetektől függetlenül logikai magas szintre, a 'clear' ('Cl' vagy 'Clr') a funkcionális bemenetektől függetlenül logikai alacsony szintre állítja.