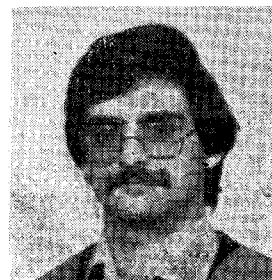


Új eljárás szinkron fázisregiszteres vezérlő egységek tervezésére



BÁNYAI ERVIN

ÖSSZEFOGLALÁS

Az eddigi módszerek a vezérlőegységet, mint nem teljesen határozott sorrendi hálózatot kezelik. A szakirodalomban hozzáférhető eljárások közül [1], [2], [3] szinkron, [4] aszinkron, [5] mindkét leírási mód együttes kezelésére alkalmas. [1], [2], [3], [4] folyamatábrás leírásból míg [5] összetartozó bemeneti-kimeneti változás sorozat specifikációjából indul ki. A dolgozatban csak szinkron esettel foglalkozunk.

A dolgozat célja

Folyamatábrával vagy sorozatokkal leírható, specifikálható feladatra egy számítógépesíthető, szinkron fázisregiszteres vezérlőegység-tervezési módszer kidolgozása, amely:

1. A maximális működési sebességet biztosítja, adott tulajdonságú építőelem-készlet esetén.
2. Adott állapotértelmezés és a maximális működési sebesség követelménye mellett a specifikált működést minimális számú állapot felvételével oldja meg.
3. Lehetőséget ad a szekunder flip-flop-ok számának és a kombinációs hálózatrészt bonyolultságának egymáshoz képesti beállítására.

Fázishatár értelmezés

A kiindulási állapottól — kiindulási állapotig tartó, összetartozó bemeneti-kimeneti változás sorozatokat nevezzük bejárési utaknak. (Lásd 1-es ábra).

A folyamatábrával megadott feladatspecifikációból kiinduló tervezési módszerek a vezérlési folyamat végrehajtási sebességének optimalizálására nem alkalmasak. A következőkben bemutatjuk, hogy ennek oka az alkalmazott fázishatár értelmezés.

A dolgozat új fázishatár értelmezést javasol a maximális működési sebesség elérése érdekében.

A vezérlőegység egy működési periódusa alatt azonos kimeneti változót ellentétes irányban állítani nem képes. Ezért (lásd 1-es ábra) az 5-ös kimeneti elem előtt fázishatár felvétele szükséges.

A régi fázishatár-értelmezésben a vezérlési folyamat mindkét bejárési út esetén váraкоzik az adott fázishatárnál, pedig ez csak az 1-es bejárési út esetén indokolt. Mivel a vezérlési folyamat olyan bejárési út esetén is váraкоzik, ahol ez a specifikált működés szempontjából nem szükséges, ezért a végrehajtás sebessége nem maximális.

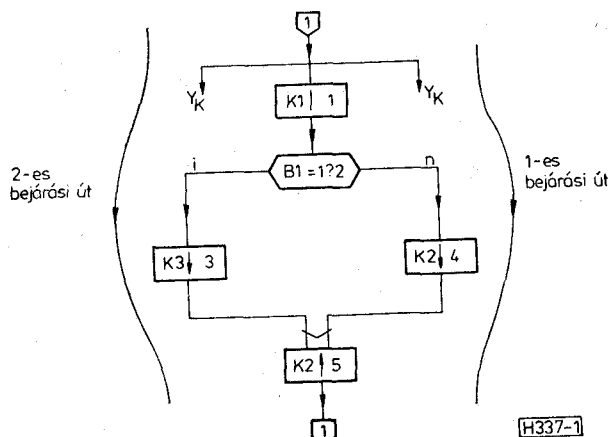
Az új fázishatár-értelmezés lényege, hogy a fázishatárok figyelembevételét a bejárési utakhoz köti. Példánkban a 2-es bejárési úton haladva a vezérlési folyamat végrehajtása során az 5-ös kimeneti elem előtt felvett fázishatárt, nem vesszük figyelembe.

Állapotösszevondás

Mint láttuk, ha egy bejárési úton létezik két ellentétes irányú kimeneti változó állítás, akkor e két kimeneti elem között a vezérlési folyamat során létre kell jönnie fázisátmenetnek, vagyis fázishatár felvétele szükséges.

A dolgozatban a kimeneti változók alapján felvett fázishatárokat nevezzük útfüggőnek, hiszen ezek figyelembevétele a bejárési úttól függ (példát lásd 2/a ábrán).

Ha bemeneti váraкоzó hurok előtt létezik döntési elem, akkor a vezérlési folyamat végrehajtása során e két elem között fázisátmenet létrejötte szükséges, ha a vezérlési folyamat a váraкоzó huroknál megáll. Hiszen példánkban (2b. ábra) amíg a vezérlési folyamat a váraкоzó hurokban áll, B1 bemeneti változó megváltoztathatta értékét. Ez a fázishatár felvételének elmulasztása esetén hibás működést okoz. A váraкоzó hurok által okozott hibás működés elkerülése érdekében felvett fázishatárokat nevezzük útfüggetlennek, hiszen felvétele a bejárési úttól függ, hanem az éppen aktuális bemeneti kombinációtól ti., hogy a vezérlési folya-

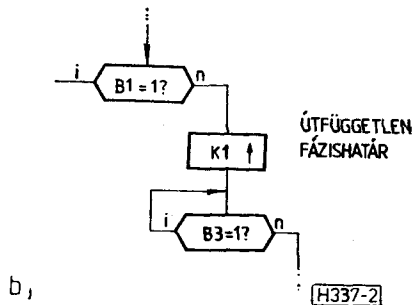
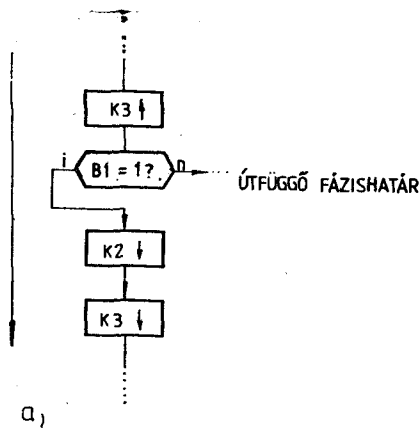


1. ábra

H337-1

Beérkezett: 1987. II. 25. (H)

FOLYAMATÁBRA
EGY
BEJÁRÁSI ÚTJA



2. ábra

mat megáll-e az adott várakozó huroknál vagy sem.

Mint látható a szükséges fázishatárookra feltételként, tartományok jönnek létre. E tartományokon belül bármelyik elem előtt felvett fázishatár biztosítja a specifikált működést.

Az eljárás akkor teljes, ha az állapotok felvételére és minimalizálására módszert tud adni valamilyen jellegű optimalitási minősítéssel. Nyilvánvaló, hogy legjobb esetben olyan eredményt lehet elérni, mint a nem teljesen határozott sorrendi hálózatok állapotösszevonásának elvi korlátja.

A dolgozatban leírt állapotösszevonás két lépésben történik.

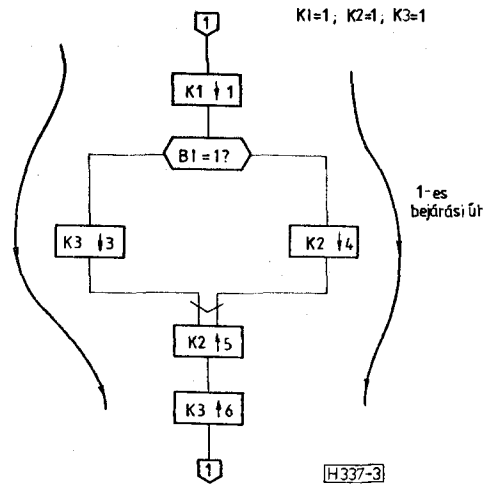
Az állapotösszevonás első részében meghatározzuk az összes olyan szakaszt, amelyben a specifikált működés biztosításához fázishatár felvétele szükséges. Példánkban (3-as ábra) két ilyen szakasz van. A 3-as és 6-os elem között, valamint a 4-es és 5-ös elem között. Az 1-es bejárési úton az 5-ös kimeneti elem előtt, a 2-es bejárési úton mind az 5-ös, mind a 6-os kimeneti elem előtt felvett fázishatár megfelel a specifikált működésnek.

Belátható, hogy ha a folyamatábra bármelyik olyan eleme előtt veszünk fel fázishatárt, amely nem esik e tartományok egyikébe sem, akkor a vezérlési folyamat végrehajtási sebessége biztosan nem lesz maximális.

Az állapotösszevonás második részében a fázishatárok felvehetőségének összes lehetséges alternatívája alapján képezünk lépcsős táblákat. Az innen kiinduló klasszikus állapotösszevonási eljárással a

Kezdeti állapot:
K1=1, K2=1, K3=1

2-es
bejárési út



3. ábra

nem teljesen határozott sorrendi hálózatok esetében elérhető állapotminimalitást a maximális sebesség követelménye mellett biztosítottuk.

Ha azonban minden egyes ilyen alternatívához külön lépcsős táblát rendelünk, akkor egy gyakorlati méretű folyamatábra esetén már igen nagy számítás kapacitásra lenne szükségünk.

A dolgozatban bemutatunk egy olyan eljárást, amelyről nem bizonyítható, hogy minimális állapotszámot ad, de kis számítás igénye, szemléletessége és a nagy hatásfokú összevonási eredménye alapján alkalmazása indokolt.

Az állapotösszevonás első részében a kiadódott szakaszokon belül és ezen szakaszok között is a fázishatárok helyét „közös rész” (metszet) képzésével határozzuk meg.

Ha egy bejárési út adott szakaszán létrehozandó fázishatár helye több metszetben is szerepelhet, akkor a döntést a minimális fázishatárszám követelményének megfelelően hozzuk. Példánkban az 5-ös kimeneti elem előtt felvett fázishatár biztosítja a specifikált működést és látható módon teljesíti a minimális állapotszám követelményét is.

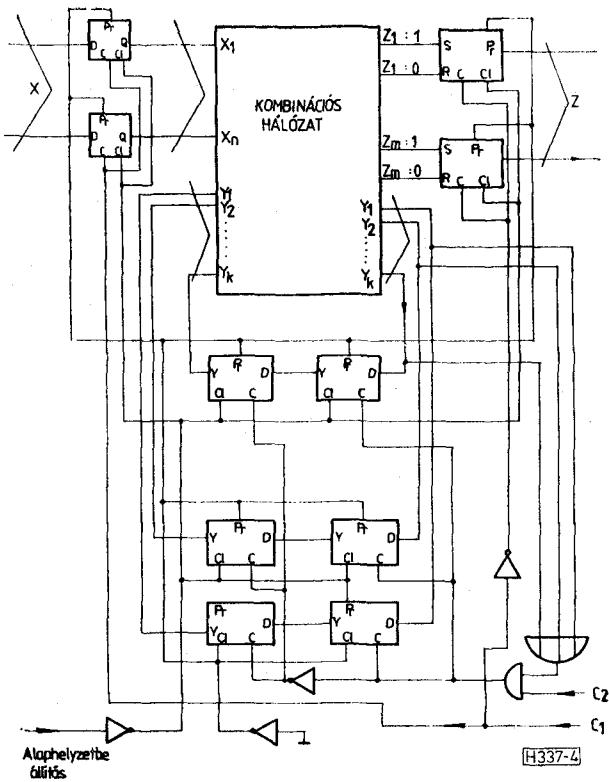
Szekunder változók számának csökkentése

Folyamatábrából kiinduló tervezési módszerek esetén a szekunder változók számának csökkentésére az állapotelrendezés kialakulása után eddig nem volt lehetőség. A dolgozat eljárást javasol a feladat megoldására.

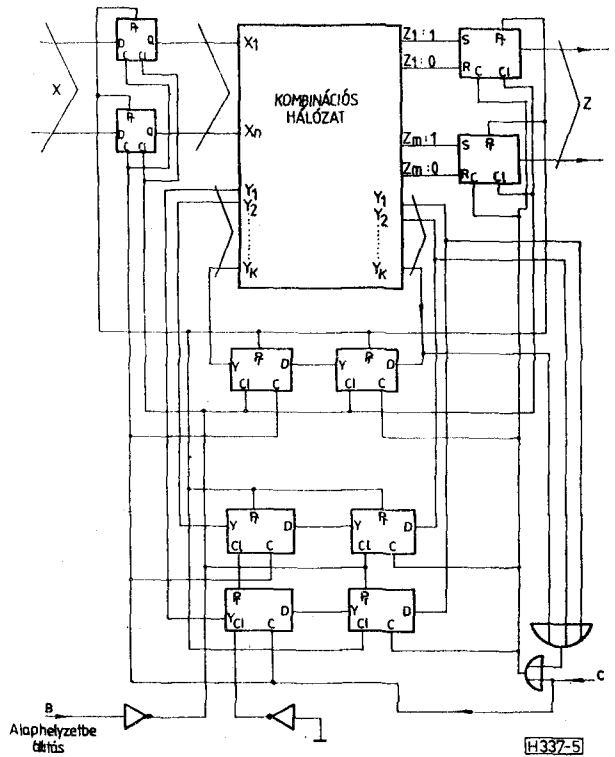
Az állapotelrendezés kialakulása után a vezérlőegység állapotainak száma tovább nem csökkenthető. A pillanatnyi állapotot mind Mealy, mind Moore modell esetén a szekunder változók fennálló kombinációja jellemzi. Tehát a szekunder változók száma csak abban az esetben csökkenthető, ha a pillanatnyi állapotok jellemzésére új jeleket — változókat keresünk.

Ilyen változók lehetnek a pillanatnyi állapotot megelőző állapotban létrejött pótlólagosan tárolt bemeneti (B^*), kimeneti (K^*) vagy szekunder (y^*) változók.

Vizsgáljuk meg, hogyan különböztet meg egy pótlólagosan tárolt például kimeneti változó két



4. ábra



5. ábra

pillanatnyi állapotot egymástól. Tételezzük fel, y_1 állapotban $K^*_1=1$ (K^*_1) illetve y_2 állapotban $K^*_1=0$ (\bar{K}^*_1). A szekunder változók számának csökkentése után y_1 és y_2 -vel jellemzett állapotot már csak y_2 szekunder változó jellemzi. Ebben az esetben e két pillanatnyi állapotot K^*_1 (pótlólagosan tárolt) kimeneti változó ellenétes értéke különbözteti meg. Azaz $y_1=y_a \cdot K^*_1$ illetve $y_2=y_a \cdot \bar{K}^*_1$. Vizsgáljuk meg, hogy e pótlólagos tárolások, milyen hatékonysággal alkalmazhatók. Ha egy adott folyamatábrában kialakult állapotelrendelés után:

elrendezés után:

a, b állapotban $K^*_1=1 / K^*_1/$ akkor

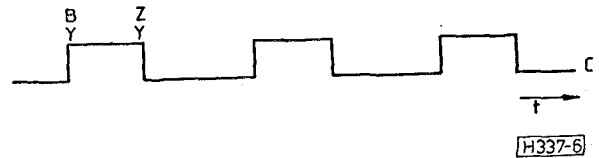
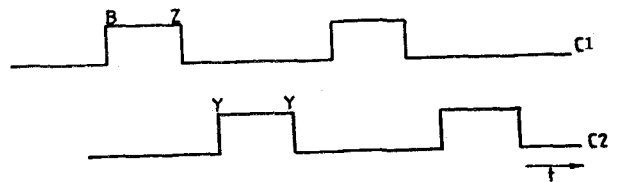
c, d állapotban $K^*_1=0 / \bar{K}^*_1/$

„a” állapot megkülönböztethető „c” és „d” állapottól

„b” állapot is megkülönböztethető „c” és „d” állapottól. Azt látjuk, hogy 4 állapot esetén, egyetlen kimeneti változó pótlólagos tárolásával 4 megkülönböztetési lehetőséget kaptunk. Egy 100 állapotú folyamatábra esetén az említett módszert alkalmazva, drasztikus szekunder változószám csökkenést lehet elérni.

Javaslat a hardware felépítés módosítására.

A szinkron fázisregiszteres vezérlőegység klasszikus hardware felépítése a 4-es ábrán látható. A bemeneti flip-flop sor a saját, a kimeneti flip-flop sor a következő fokozat szinkronizációs feladatait látja el.



6. ábra

A visszacsatoló ágban a szinkron működést, a hazard jelenségek elkerülése érdekében data-look-out flip-flopok biztosítják.

A specifikált működés alapvető feltétele a szekunder változók N-ből 1 kódjának biztosítása. Az 1-nél több aktív szekunder változó létrejöttét a tervezés módszere, a csupa „0”-kód kialakulását az órajel kapuzó hálózat és a kétfázisú órajel akadályozza meg.

A dolgozat a hardware felépítés módosítását javasolja, aminek alapján a specifikált működést egyfázisú órajellel biztosítjuk. (5. ábra)

Ezzel a megoldással a vezérlési folyamat végrehajtási sebessége az eddigi módszerekhez képest kétszeresére nő.

A 6-os ábrán a klasszikus és a dolgozatban javasolt órajel-funkció hozzárendelések láthatók.

A csupa 0-kód kialakulása abban az esetben fordulhat elő, ha a várakozó hurokban álló vezérlési folyamat továbbhaladását biztosító bemeneti kombináció nem éri meg. Ebben az esetben az órajel kapuzó hálózat megakadályozza C_2 órajel hatását az áramkörre mindaddig, ameddig a bemeneti flip-flop-sor e kombináció létrejöttét nem érzékeli. Ilyen esetben a dolgozatban javasolt órajel hozzárendelés mellett a lefutó él hatását szintén az órajel-kapuzó hálózat semlegesíti. Ahhoz, hogy a vezérlési folyamat továbbhaladását előíró bemeneti kombináció megjelenését érzékeltessük a bemeneti flip-flop-sor nem kapuzott, hanem „szabadon futó” órajelet kap.

A módosított felépítésű vezérlőegység hazardmentes működése a dolgozatban bizonyított.

A dolgozatban javasolt tervezési eljárásban a szekunder változók számának csökkentésére pótlólagos tárolásokat használunk. Az ilyenkor szükséges órajel hozzárendeléssel a dolgozat részletesen foglalkozik.

A dolgozatban javasolt tervezési eljárással megtervezett vezérlőegység egy és kétfázisú órajellel egyaránt működtethető.

Az önálló eredmények összefoglalása

A dolgozat olyan tervezési eljárást javasol, amely biztosítja a szinkron fázisregiszteres vezérlőegység maximális sebességű működését és az állapot-minimalizálás szempontjából egyenértékű eredményt szolgáltat a nem teljesen határozott sorrendi hálózatok ismert állapotösszevonási algoritmusaival.

A javasolt eljárás az alábbi önálló eredményeken alapul:

1. Folyamatábrából vagy bemeneti—kimeneti sorozatokból kiinduló alapállapot-értelmezés kidolgozása;
2. Új fázishatár-értelmezés bevezetése a maximális sebességű működés biztosítás érdekében;
3. A maximális működési sebesség peremfeltétele mellett minimális számú állapot felvételére szolgáló módszer kidolgozása;
4. A szekunder változók számának csökkentése pótlólagos tárolásokkal és az ehhez tartozó hardware felépítés;
5. A szakirodalomból ismert szinkron fázisregiszteres vezérlőegységek egyfázisú órajellel történő működtetésére alkalmas hardware felépítés kidolgozása.

Köszönetnyilvánítás

Ezúton is megköszönöm Dr. Arató Péter professzornak, TDK vezető tanáromnak azt az emberi és szakmai támogatást, mely nélkül a dolgozat nem készülhetett volna el.

I R O D A L O M J E G Y Z É K

- [1] *S. Wendt.*: Entwurf komplexer Schaltwerke. Springer 1974.
- [2] *C. B. Clare.*: Designing Logic Systems Using State Machines. Mc Graw-Hill Book Company, 1973.
- [3] *Kalmár P.*: Vezérlő egységek folyamatára alapján történő logikai tervezésének egy módszere. Kandidátusi értekezés. Budapest, 1978.
- [4] *Terplán S.*: Aszinkron jellegű logikai hálózatok tervezésének egy módszere a folyamatára alapján. Kandidátusi értekezés. Budapest, 1981.
- [5] *Arató Péter.*: Vezérlőegységek logikai tervezése előírt bemeneti és kimeneti változások alapján. Doktori értekezés. Budapest, 1984.

Bányai Ervin

BME Folyamatszolgáltatási Tanszék.

Témavezető: Dr. Arató Péter, egyetemi tanár

Lapunk példányonként megvásárolható:
az V., Váci utca 10. és
az V., Bajcsy-Zsilinszky út 76. szám alatti
hírlapboltban
