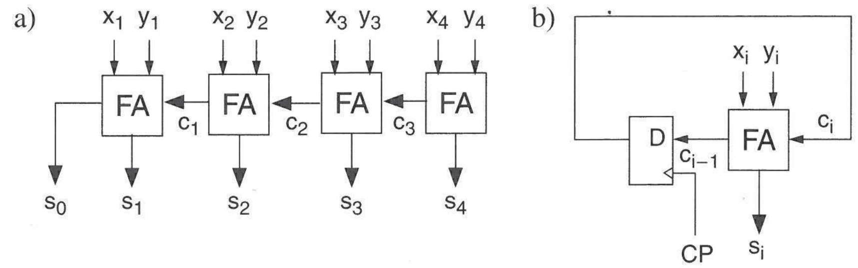
Nätet enligt figur 9.26d) utnyttjar ett minimalt antal tillstånd. I det generella fallet är minimeringen dock inte avslutad, eftersom det inte är säkert att de maximala förenlighetsmängderna ger ett minimalt nät. Problemet att ur de maximala förenlighetsmängderna ta fram ett minimalt antal förenlighetsmängder är dock ganska besvärligt och vi redovisar ingen metod för detta.

9.10 Sekvensnät och iterativa kombinatoriska nät

Kombinatoriska nät och sekvensnät kan ibland vara alternativa lösningar till samma informationsbehandlingsproblem. Det kombinatoriska nätet kan generera samma funktion som sekvensnätet förutsatt att i ett visst ögonblick hela ingångskombinationen (= argumentet) finns tillgänglig. Insignalerna måste komma in parallellt. Sekvensnätet kan däremot acceptera att insignalerna flyter in i serieform. Vi ska illustrera parallell-serie problemet med fallet addition av två positiva binära tal $X = <x_1, x_2, x_3, x_4>$ och $Y = <y_1, y_2, y_3, y_4>$. Se figur 9.27.

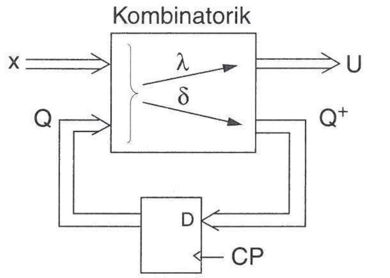
I figur 9.27a) visas den tidigare presenterade adderaren som förutsätter att de binära sifforna uppträder i parallell form. Resultatet blir tillgängligt på det kombinatoriska nätets utgångar med en fördröjning som beror på den enskilda grindens tröghet samt nätets maximala djup.

I figur 9.27b) visas ett sekvensnät som utför serieaddition. Talen X och Y inkommer till heladderaren med minst signifikant bit först och i varje nytt klockpulsintervall uppträder en ny bit. Heladderaren bildar på vanligt sätt summan, s_i, med samma intervalluppdelning som förutsätts gälla för insignalerna. Utgående carry ska användas som ingående i närmast högre signifikanta bit och måste återkopplas via en D-vippa.



Figur 9.27 a) Parallell addition b) Seriell addition

Om vi jämför figur 9.27a) med 9.27b) gör vi en intressant iakttagelse. De block som bygger upp det iterativa kombinatoriska nätet är exakt identiska med motsvarande sekvensnäts kombinatoriska del. Detta är ingen tillfällighet utan gäller generellt. Låt oss betrakta ett sekvensnät av Mealy-typ i figur 9.28 och se vad som händer i de olika klockintervallen (index på respektive signal anger signalens värde i motsvarande klockintervall):



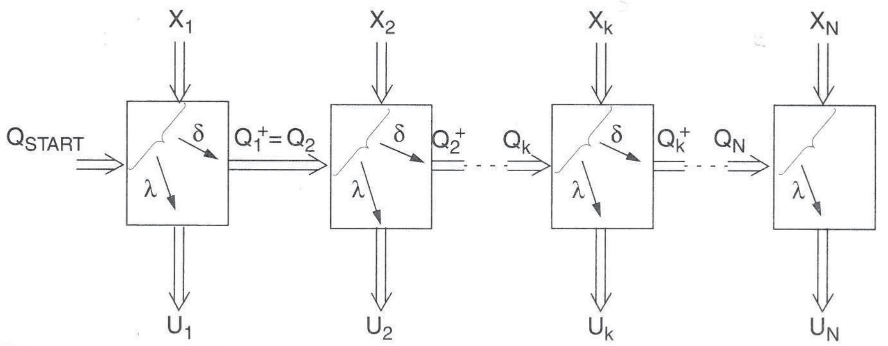
Figur 9.28 Sekvensnät av Mealy-modell

klockintervall 1: $Q_1{}^+ = \delta(Q_1, X_1)$ $\quad Q_1 = Q_{start}$
$U_1 = \lambda(Q_1, X_1)$

klockintervall 2: $Q_2{}^+ = \delta(Q_2, X_2)$
$U_2 = \lambda(Q_2, X_2)$ $\quad$ där $Q_2 = Q_1{}^+$
$\vdots$ $\quad \vdots$

klockintervall k: $Q_k{}^+ = \delta(Q_k, X_k)$
$U_k = \lambda(Q_k, X_k)$ $\quad$ där $Q_k = Q^+{}_{k-1}$

I klockintervall 1 bildas alltså, i det kombinatoriska nätet, $Q_1{}^+$ och U_1 ur Booleska uttryck av Q_{start} och X_1. I stället för att återkoppla $Q_1{}^+$ via en D-vippa skulle vi nu kunna leverera $Q_1{}^+$ som inre tillstånd (Q_2) till ett identiskt nät. Detta näts insignaler ska då korrespondera mot de insignaler motsvarande sekvensnät tar emot under klockintervall 2, nämligen X_2. Q_2 och X_2 genererar i sin tur U_2 och $Q_2{}^+$ av vilka $Q_2{}^+$ kopplas vidare till ett nytt identiskt nät o.s.v., se figur 9.29.



Figur 9.29 Principiellt utseende för ett iterativt kombinatoriskt nät

Det torde härmed stå klart att det iterativa kombinatioriska nätet i figur 9.29 parallellt utför samma informationsbehandling på variabeln X som sekvensnätet i figur 9.28 utför seriellt.

Det allmänna iterativa kombinatoriska nätet består alltså av ett antal identiska enheter, som vi kallar celler. En cell är vidare exakt identisk med kombinatoriken för motsvarande sekvensnät. Eftersom de tidigare presenterade syntesmetoderna för sekvensnät behandlar just realiseringen av den kombinatoriska delen kan följaktligen samma syntesmetoder utnyttjas för iterativa kombinatoriska nät.

Den enda anpassning vi behöver göra är att formulera om vårt problem till ett sekvensnätsproblem. Insignalerna till en viss cell, säg cell k, i det iterativa nätet ska då korrespondera mot de insignaler sekvensnätet tar emot under klockintervall k. Med detta i minnet torde tillståndsgrafen lätt kunna ställas upp.

Såsom visas i kommande exempel kan alltid några, ibland alla, celler förenklas i något avseende.

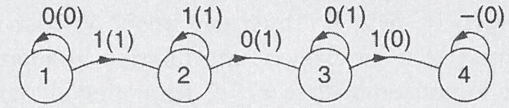
Exempel 9.9

Ett iterativt kombinatoriskt nät med de binära insignalerna <x_1, x_2, …, x_N> ska ge utsignalen u = 1 omm det bland insignalerna finns exakt en grupp av sammanhängande ettor.

Lösning:

För seriefallet gäller att värdena för signalen x i klockpulsintervallen 1, 2, …, N korresponderar mot de parallellt inkommande signalerna <x_1, x_2, …, x_N> och att u = 1 omm alla inkommande ettor uppträder i konsekutiva klockpulsintervall. Tillståndsgraf, tillståndstabell, flödestabell och Karnaughdiagram framgår av figur 9.30.

a)



b)

δ(λ)	x = 0	x = 1
1	1(0)	2(1)
2	3(1)	2(1)
3	3(1)	4(0)
4	4(0)	4(0)

c)

δ(λ)		x = 0	x = 1
1	00	00(0)	01(1)
2	01	11(1)	01(1)
3	11	11(1)	10(0)
4	10	10(0)	10(0)

d)

q_1q_2 \ x	0	1
00	0	0
01	1	0
11	1	1
10	1	1

00	0	1
01	1	1
11	1	0
10	0	0

00	0	1
01	1	1
11	1	0
10	0	0

Figur 9.30 Syntes för seriefallet

Ur Karnaughdiagrammen erhålls:

$$q_1^+ = q_1 + q_2x'$$
$$q_2^+ = q_1'x + q_2x'$$
$$u = q_1'x + q_2x'$$

I ett allmänt fall skulle nu varje cell i det iterativa nätet beskrivas av ovanstående ekvationer. Emellertid kan i de flesta fall cellerna förenklas i något avseende, så också här. Det kombinatoriska nätet har bara en utsignal och det finns därför ingen anledning att utrusta varje cell med kombinatoriken för u-funktionen. Det är tillräckligt att den sista cellen (den N:te) producerar önskat u-värde. Cell N behöver däremot inte generera tillståndssignalerna q_1^+ och q_2^+, då efterföljande cell saknas. Genom insättning av begynnelsevillkoret $q_1 = q_2 = 0$ (starttillståndet är kodat 00) i de Booleska uttrycken för q_1^+ och q_2^+ kan cellerna i början av iterationskedjan förenklas.

Vi kan särskilja följande typer av celler:

cell 1: $(q_1 = q_2 = 0)$
$q_1^+ = 0$
$q_2^+ = x$

Förenklingen fortsätter i cell 2, eftersom här gäller att $q_1 = 0$.

cell 2: $(q_1 = 0)$
$q_1^+ = q_2 x'$
$q_2^+ = q_2 + x$

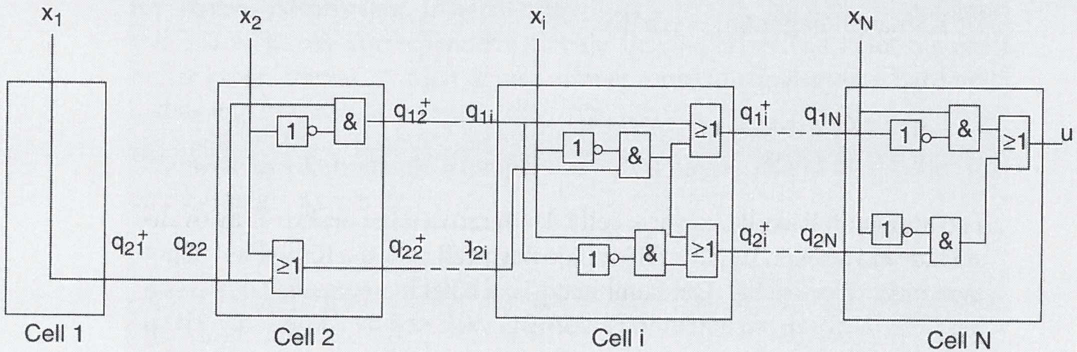
Nätets övriga celler t.o.m. den näst sista ges av de tidigare framtagna uttrycken. För cell 3 gäller visserligen $<q_1q_2> \neq <10>$, men detta resulterar inte i någon förenkling.

cell 3 t.o.m. (N–1):
$q_1^+ = q_1 + q_2 x'$
$q_2^+ = q_1' x + q_2 x'$

Den sista cellen är unik och ges av:

cell N:
$u = q_1' x + q_2 x'$

Nätets realisering visas i figur 9.31.



Figur 9.31 Nätets realisering

I exempel 9.9 har vi valt att använda oss av Mealy-modellen. Moore-modellen fungerar naturligtvis också, men är för iterativa kombinatoriska nät mindre lämplig. Moore-modellens utsignal beror, som bekant, endast av tillståndsvariablerna. Därför kommer, om ingen åtgärd vidtages, utsignalen från cell N att bli oberoende av x_N.

Användandet av en iterativ struktur ger alltså möjlighet att realisera kombinatoriska nät med ett stort antal insignaler N för en kostnad som stiger linjärt med N. Detta kan jämföras med en realisering enligt disjunktiv eller konjunktiv normalform (realisering med permanentminne) där kostnaden stiger ungefär med 2^N (= antalet positioner i minnesarean med N bitars adress och ordlängden 1).

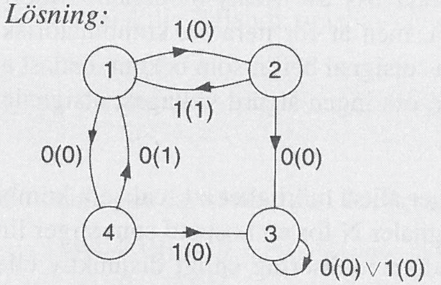
En allmän jämförelse mellan informationsbehandling utförd parallellt respektive seriellt visar i regel följande. Den parallella behandlingen i ett kombinatoriskt nät är snabb men kräver ett stort komponentuppbåd. Den seriella behandlingen är långsammare men kräver i gengäld färre komponenter. Valet mellan serie- och parallellbehandling är alltid avhängigt av det större sammanhanget. Om exempelvis insignalerna primärt levereras i serieform utgör naturligtvis sekvensnätets längre bearbetningstid inte någon nackdel medan å andra sidan en ren parallell behandling skulle kräva en serie-parallell omvandling före det kombinatoriska nätet. (Hur sådan omvandling utförs visas i kapitel 11.2.)

För nät vars begynnelsetillstånd omedelbart lämnas kan förenklingen av celler oftast drivas längre.

Exempel 9.10

Ett antal binära signaler $<x_1, x_2, \ldots, x_N>$ införs till ett kombinatoriskt nät för vars utsignal u gäller, att u = 1 omm varje grupp av konsekutiva nollor och varje grupp av konsekutiva ettor består av ett jämnt antal.

Lösning:



δ–λ	x = 0	x = 1
1	4(0)	2(0)
2	3(0)	1(1)
3	3(0)	3(0)
4	1(1)	3(0)

a	q_1q_2	x = 0	x = 1
1	00	10(0)	01(0)
2	01	11(0)	00(1)
3	11	11(0)	11(0)
4	10	00(1)	11(0)

q_1q_2 \ x	q_1^+: 0	q_1^+: 1	q_2^+: 0	q_2^+: 1	u: 0	u: 1
00	1	0	0	1	0	0
01	1	0	1	0	0	1
11	1	1	1	1	0	0
10	0	1	0	1	1	0

Figur 9.32 Syntes för seriefallet

Lösningsgången enligt figur 9.32 ger de Booleska uttrycken:

$$q_1^+ = q_1'x' + q_2x' + q_1x$$
$$q_2^+ = q_2'x + q_2x' + q_1x$$
$$u = q_1q_2'x' + q_1'q_2x$$

Cellerna förenklas därefter utgående från givet starttillstånd, här kodat 00.

cell 1:

Genom att sätta in $q_1 = q_2 = 0$ i uttrycken för q_1^+ och q_2^+ eller genom att utnyttja Karnaughdiagrammen, där vi vet att bara tillstånd 00 är möjligt, erhålls

$$q_1^+ = x'$$
$$q_2^+ = x$$

cell 2:

Nätet befinner sig nu antingen i tillstånd 01 eller i tillstånd 10. Utnyttjar vi Karnaughdiagrammen eller sätter in $q_2 = q_1'$ i de Booleska uttrycken får vi

$$q_1^+ = q_2^+ = q_1'x' + q_1x$$

cell 3 och alla följande udda celler:

De möjliga tillstånden är 00 eller 11 ($q_1 = q_2$), vilket ger

$$q_1^+ = q_1 + x'$$
$$q_2^+ = q_1 + x$$

cell 4 och alla följande jämna celler:

Nätet kan inte befinna sig i tillstånd 00 ($q_1 + q_2 = 1$) och cellerna förenklas till

$$q_1^+ = q_2^+ = q_1x + q_2x'$$

Utsignalen u ska endast bildas i sista cellen (cell N). Om antalet celler är udda blir naturligtvis alltid u = 0 och vi förutsätter därför att N är ett jämnt tal. Nätet kan då inte befinna sig i tillstånd 00 och u kan förenklas till

$$u = q_1'x + q_2'x'$$

Intressant att notera är att motsvarande sekvensnäts kombinatoriska funktionsuttryck inte ingår i en enda av det iterativa nätets celler. Funktionen blir visserligen den korrekta om så vore fallet, men nätet blir inte minimalt.

Näten i exempel 9.9 och 9.10 hade bestämda starttillstånd och p.g.a. starttillståndens kodning kunde celler förenklas på angivet sätt. Funktionellt var de dock identiska med den allmänna cellen (cell i) utgående från att bara vissa tillstånd var möjliga. Kan vi utifrån påverka tillståndsvariablerna till cell 1 ger vi följaktligen nätet olika starttillstånd och får därmed möjlighet att med styrvariabler ändra nätets funktion. Detta visar vi med ett exempel.

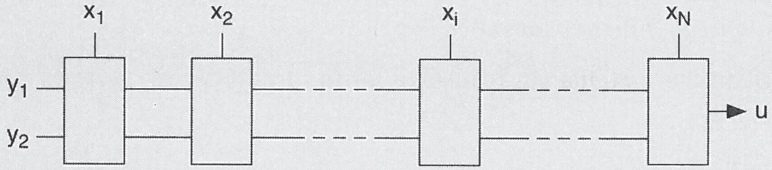
Exempel 9.11

Konstruera ett iterativt kombinatoriskt nät med N insignaler X = <x_1, x_2, …, x_N>, två styringångar y_1, y_2, och en utsignal u.

För $y_1y_2 = 00$ ska u = 1 omm antalet ettor i $X \geq 3$
För $y_1y_2 = 01$ ska u = 1 omm antalet ettor i $X \geq 2$
För $y_1y_2 = 10$ ska u = 1 omm antalet ettor i $X \geq 1$
För $y_1y_2 = 11$ ska u ovillkorligen = 1

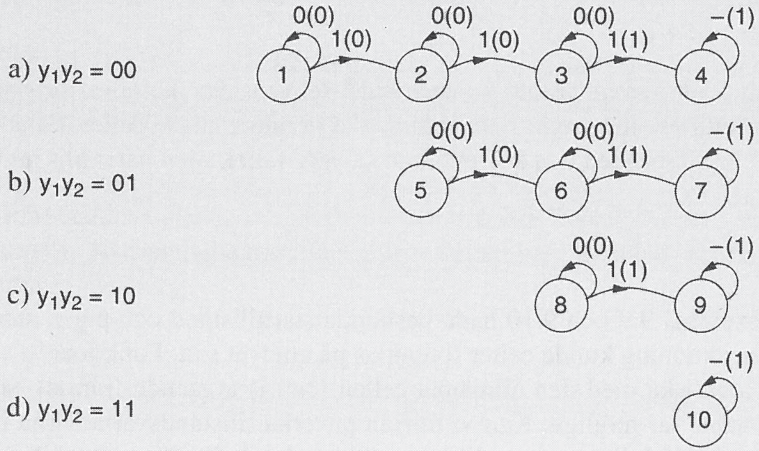
Lösning:

Uppgiften kan lösas på ett par olika sätt. Den önskade strukturen är dock den i figur 9.33, där y_1, y_2 är insignaler bara till cell 1. De definierar därmed nätets starttillstånd.



Figur 9.33 Nätets önskade struktur

För $y_1y_2 = 00$ ges önskad funktion av grafen i figur 9.34a) med starttillståndet 1. P.s.s. erhålls önskad funktion för $y_1y_2 = 01$ av grafen i figur b) med starttillståndet 5, för $y_1y_2 = 10$ och $y_1y_2 = 11$ av graferna i c) respektive d) med starttillstånden 8 respektive 10.

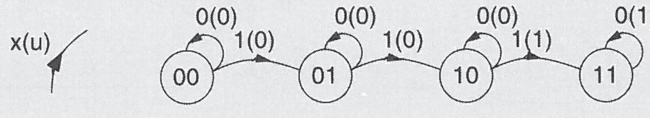


Figur 9.34 a) Nätets funktion med starttillståndet $y_1y_2 = 00$ b) $y_1y_2 = 01$ c) $y_1y_2 = 10$ d) $y_1y_2 = 11$

Genom att studera graferna (eller med formell minimering) finner man lätt att 2 ≡ 5, 3 ≡ 6 ≡ 8 och 4 ≡ 7 ≡ 9 ≡ 10. Den minimerade grafen blir således identisk med 9.34a) där

$Q_{start} = 1$ för $y_1y_2 = 00$
$Q_{start} = 2$ för $y_1y_2 = 01$
$Q_{start} = 3$ för $y_1y_2 = 10$
$Q_{start} = 4$ för $y_1y_2 = 11$

Lämpligt är att koda tillstånden i enlighet med y_1y_2, varvid lösningen fortskrider enligt figur 9.35. Cellerna 1 t.o.m. N–1 blir identiska och ges av ekvationerna för q_1^+ och q_2^+. Cell N är unik och beskrivs av ekvationen för u.



δ(λ)	x = 0	x = 1
00	00(0)	01(0)
01	01(0)	10(0)
10	10(0)	11(1)
11	11(1)	11(1)
q_1q_2	$q_1^+q_2^+(u)$	

q_1^+:

q_1q_2 \ x	0	1
00	0	0
01	0	1
11	1	1
10	1	1

q_2^+:

q_1q_2 \ x	0	1
00	0	1
01	1	0
11	1	1
10	0	1

u:

q_1q_2 \ x	0	1
00	0	0
01	0	0
11	1	1
10	0	1

$q_1^+ = q_1 + q_2x$
$q_2^+ = q_2x' + q_2x' + q_1x$ (alt q_1q_2)
$u = q_1q_2 + q_1x$

Figur 9.35 Lösning efter framtagning av minimal graf

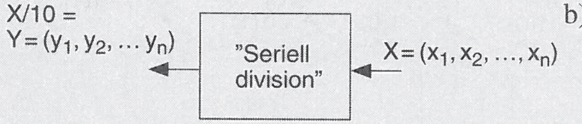
Som ett avslutande exempel på användning av sekvensnät i allmänhet och iterativa kombinatoriska nät i synnerhet ska vi lösa följande konstruktionsproblem.

Exempel 9.12

Konstruera ett kombinatoriskt nät som omvandlar ett binärt n-bitars heltal X till NBCD-kod.

Lösning:

Vi börjar med att betrakta det sekventiella fallet. Den första frågan gäller då om mest signifikant eller minst signifikant bit ska uppträda först. Med kännedom om den divisionsmetod för talomvandling som presenterades i kapitel 2 drar vi den slutsatsen att mest signifikant bit ska uppträda först. Vidare bör det vara möjligt att konstruera ett sekvensnät som med $X = \langle x_1, x_2, \ldots, x_n \rangle$ som seriell ingång på sin seriella utgång producerar $X/10 = Y = \langle y_1, y_2, \ldots, y_n \rangle$. Se figur 9.36a)

a)


b)

Q	x = 0	x = 1
0	(0) 0	(0) 1
1	(0) 2	(0) 3
2	(0) 4	(0) 5
3	(0) 6	(0) 7
4	(0) 8	(0) 9
5	(1) 0	(1) 1
6	(1) 2	(1) 3
7	(1) 4	(1) 5
8	(1) 6	(1) 7
9	(1) 8	(1) 9
Q	$(y)Q^+$	

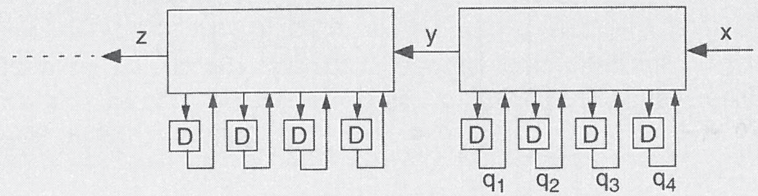
Figur 9.36 a) Sekvensnät för seriell division med 10
b) Tillståndstabell

Tillståndstabellen för sekvensnätet får utseendet enligt figur 9.36b). Den är framtagen med följande resonemang. Med Q = 0 och x = 1 går vi till tillstånd 1. Om i nästa intervall x = 1 har vi hittills noterat att talet x börjar med binärkombinationen 11, d.v.s. NBCD-talet 3, och vi går därför till Q = 3. Hittills har den influtna delen av X inte kommit upp i tio, varför y hela tiden sätts till 0. Om emellertid Q = 7 och x = 1 betyder detta att X börjar med 1111 = 15. Av denna orsak ger vi då y = 1 och $Q^+ = 5$. Som synes motsvarar y kvotsiffran och Q^+ resten i varje klockintervall.

Vid NBCD-kodning av Q får vi via Karnaughdiagram följande funktionsuttryck för kombinatoriken.

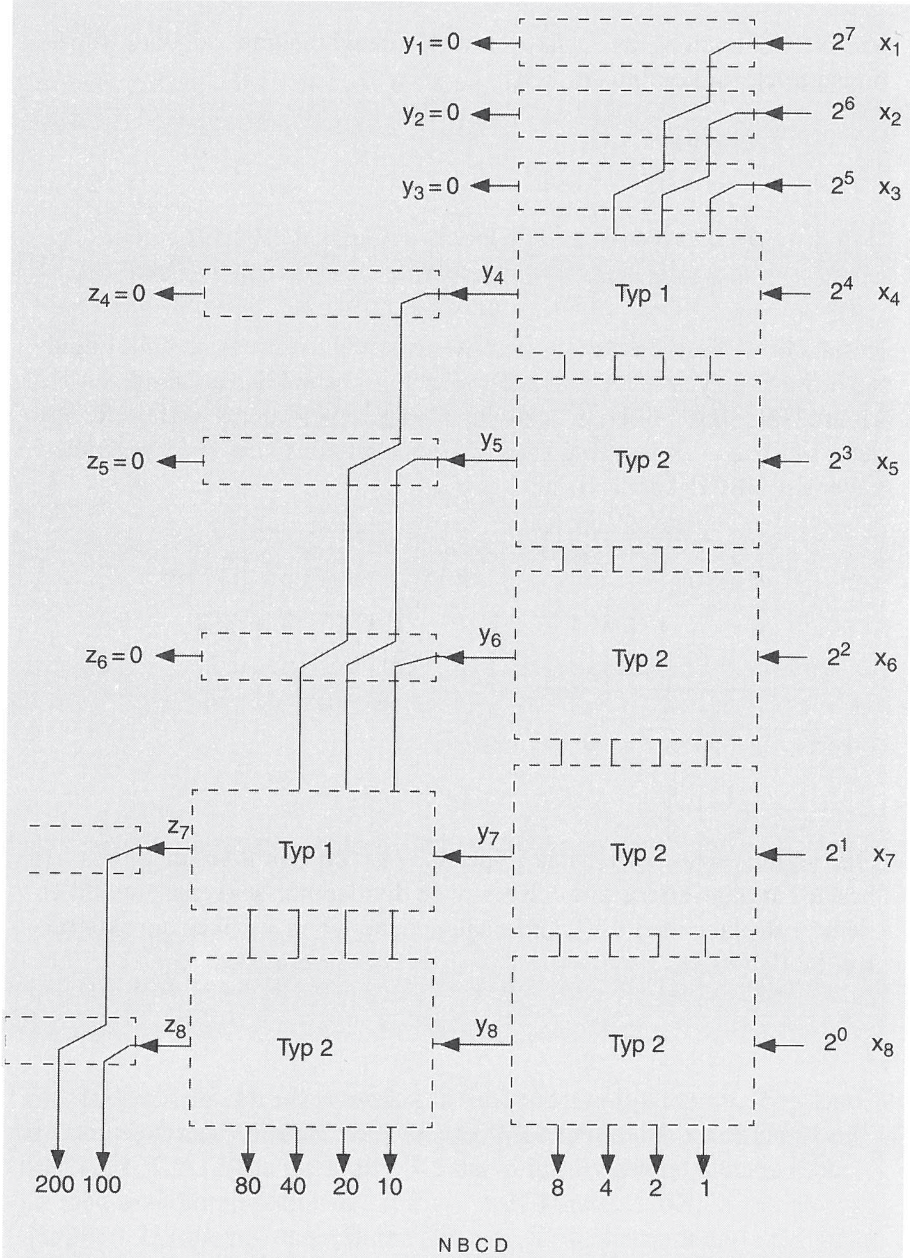
$$q_1^+ = q_2q_3'q_4' + q_1q_4$$
$$q_2^+ = q_3q_4 + q_2'q_3 + q_1q_4'$$
$$q_3^+ = q_1'q_2'q_4 + q_2q_3q_4'$$
$$q_4^+ = x$$
$$y = q_1 + q_2q_3 + q_2q_4$$

Resultatet så långt består i en iterativ struktur av sekvensnät enligt figur 9.37. Signalen y som representerar talet X/10 behandlas ju lämpligen på samma sätt som x för att generera z som representerar x/100 etc. På detta sätt har vi ett nät som efter n klockintervall i sina D-vippor innehåller det NBCD-kodade talet.



Figur 9.37 Nätets struktur i seriefallet

Omvandlingen av sekvensnät i figur 9.37 till ett rent kombinatoriskt nät består i att konvertera vart och ett av de dividerande sekvensnäten till en iterativ struktur med tidskoordinanten ersatt av en vertikal rumskoordinat. Se figur 9.38.



Figur 9.38 Det iterativa nätet

Iterationen börjar uppifrån (motsvarande $t = 0$ för sekvensnätet). Så länge $q_1 = q_2 = 0$ blir näten enbart kortslutningar, vilket kan verifieras i de ovan givna funktionsuttrycken.

$$q_1^+ = 0 \qquad q_2^+ = q_3$$
$$q_3^+ = q_4 \qquad q_4^+ = x$$

Först från och med nivån med x_4 som ingång blir funktionsuttryckens realisering ett verkligt nät. Fortfarande är emellertid $q_1 = 0$, varför nätet typ 1 realiserar

$$q_1^+ = q_2 q_3' q_4$$
$$q_2^+ = q_3 q_4 + q_2' q_3$$
$$q_3^+ = q_1' q_2' q_4 + q_2 q_3 q_4'$$
$$q_4^+ = x$$
$$u = q_2 q_3 + q_2 q_4$$

På nästa nivå är alla fyra tillståndsvariablerna utbildade, varför vi här har att med nätet typ 2 realisera den kompletta avbildningen enligt tidigare funktionsuttryck.

9.11 Formell definition av sekvensnät

Ett synkront sekvensnät A definieras med hjälp av fem storheter, d.v.s. A är en femtupel

$$A = \langle S_{in}, S_{ut}, S_Q, \delta, \lambda \rangle$$

där

S_{in} är en ändlig mängd av ingångsvärden (ingångsalfabet)
S_{ut} är en ändlig mängd av utgångsvärden (utgångsalfabet)
S_Q är en ändlig mängd av inre tillstånd (tillståndsalfabet)

Dessa tre mängder har följande element

$$S_{in} = \{X_1, X_2, \ldots, X_N\}$$
$$S_{ut} = \{U_1, U_2, \ldots, U_M\}$$
$$S_Q = \{Q_1, Q_2, \ldots, Q_P\}$$