

LABORATION **ET 22**
DIGITALTEKNIK 3

INNEHÅLL:

1. Avkodare/Multiplexer.
2. Register.
3. Adderare.
4. RAM-minne.

Handledare:

Namn: Nr: Labplats

Laborationen utförd den

Laborationen inlämnad den

Godkänd den av

I de föregående försöken har de grundläggande byggbitarna i digitaltekniken studerats. När dessa byggbitar skall sammansättas till en digital konstruktion brukar kretslösningen uppdelas i två huvudgrupper, kombinatoriska kretsar och sekvenskretsar. De kombinatoriska kretsarna styrs genom en kombination av insignaler. Grindar är exempel på sådana kretsar. Sekvenskretsar styrs också av en kombination av insignaler men insignalernas tidsföljd (sekvens) är viktig. Sekvenskretsar är därför beroende av tidigare förlopp varför de måste ha minne. De bistabila vipporna är exempel på sekvenskretsar.

1. AVKODARE/MULTIPLEXER.

Avkodare.

En avkodare är en kombinatorisk krets med flera utgångar, som omvandlar en kombination av ingångssignalerna till koden "en av alla". Detta innebär att endast en utgång kan ettställas för en viss ingångskombination. Vidare gäller i allmänhet att två olika ingångskombinationer ej kan ettställa samma utgång.

En avkodare med två ingångar som avkodar den binära formen av siffrorna 0, 1, 2, 3, kan schematiskt tecknas enligt Fig. 1.

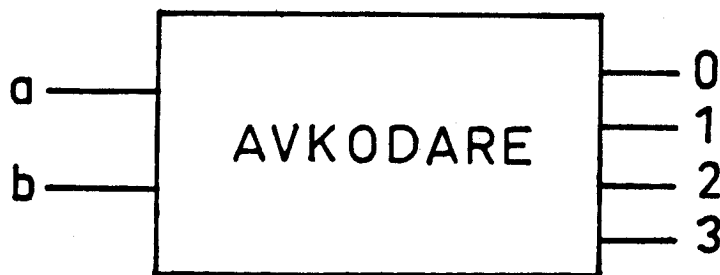


Fig. 1.

Avkodaren har två ingångar a och b på vilka man kan lägga de 4 olika kombinationerna:

$ba = 00, 01, 10$ och 11

Dessa kombinationer svarar mot de decimala siffrorna 0, 1, 2 och 3. Avkodaren har 4 utgångar som svarar mot dessa siffror. För en viss ingångskombination ska alltså endast en utgång ettställas. På a-ingången lägges den minst signifikanta binära siffran, dvs den högra biten i det binära talet. På b-ingången lägges den mest signifikanta biten, dvs den vänstra.

Denna avkodning kan utföras i en sk diodmatris. Fig. 2 visar avkodarens uppbyggnad. Vi förutsätter förutom hög representation även att inversen till ingångssignalerna finns tillgängliga.

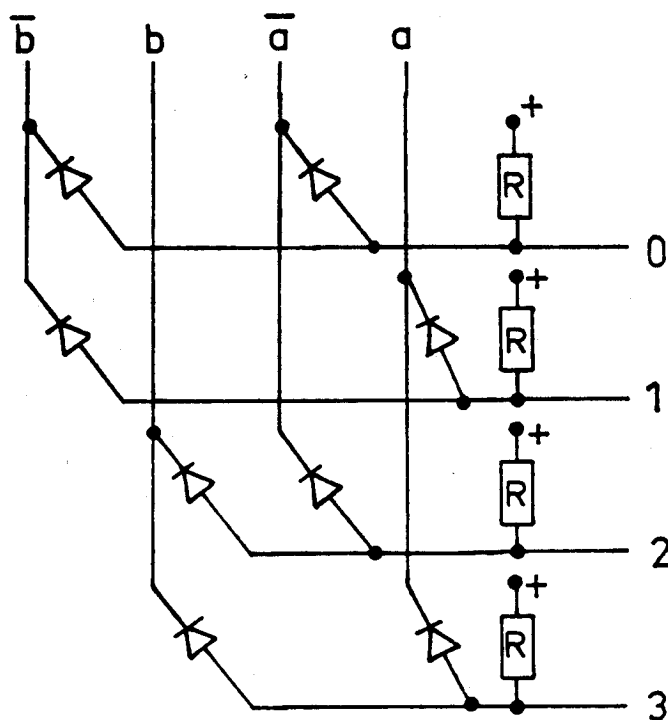


Fig. 2.

Vi ser ur figuren att villkoret för att utgång 0 ska ha hög potential (logisk etta) är att ingen ström går genom resistansen R .

Detta medför att både \bar{a} och \bar{b} måste vara ett, dvs $ba = 00$. För utgång 1 gäller att utgången är ettställd endast om $ba = 01$ osv.

Om vi lägger in ett visst binärt tal ba så kommer alltså utgången med motsvarande decimala numrering att bli ettställd.

Matrisavkodare är en enkel och billig lösning om ej alltför många ingångar användes. Avkodare kan emellertid även byggas upp med grindar. Vi visar i Fig. 3 en annan variant av realiseringen av avkodarblocket i Fig. 1.

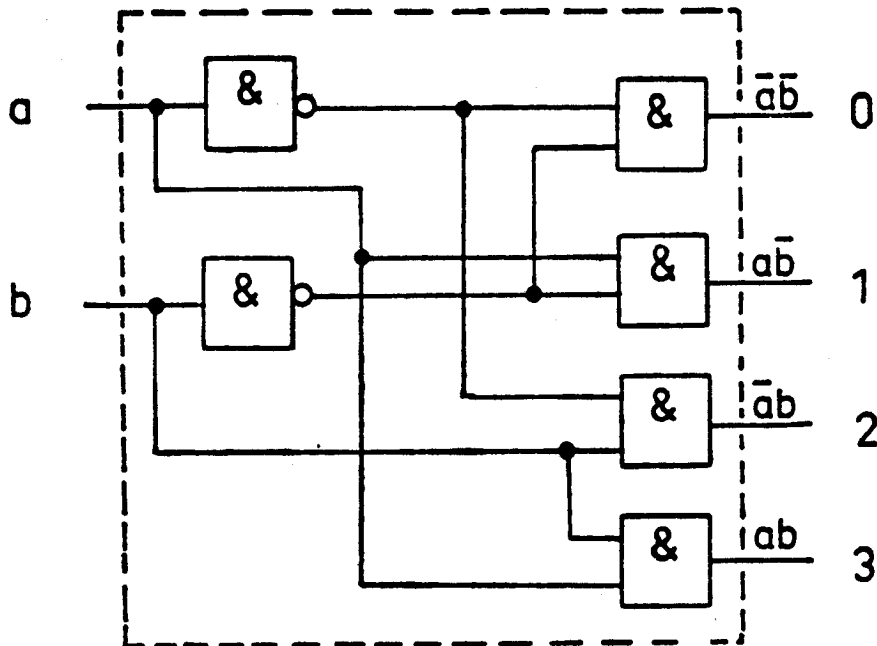


Fig. 3.

Avkodarens funktion framgår av följande resonemang.

Utgång 0 ettställs om $ba = 00$
 " 1 " " $ba = 01$
 " 2 " " $ba = 10$
 " 3 " " $ba = 11$

De fyra utgångskretsarna skall alltså ha följande logiska funktioner.

Utgång	Funktion
0	$\bar{a}\bar{b}$
1	$a\bar{b}$
2	$\bar{a}b$
3	ab

Dessa funktioner kan realiseras med varsin AND-grind som Fig. 3 visar. Vi ser att de funktioner utgångarna realiserar helt enkelt är de fyra olika mintermer som kan bildas av de två variablerna a och b.

Avkodaren ovan var alltså uppbyggd av enbart AND-grindar och inverterare (NAND-grindar). Generellt gäller att alla avkodare kan byggas upp av enbart AND-grindar och inverterare.

Vi skall nu konstruera en avkodare, som omvandlar binära tal av längden 3 bitar till deras decimala motsvarighet, dvs siffrorna 0-7. Vi kallar ingångarna a, b och c, där a står för bitpositionen längst till höger. De åtta utgångarna betecknas med siffrorna 0-7. Nedanstående tabell visar hur vi bestämmer utgångarnas logiska funktioner.

Utgång nr	Motsvarande binära tal cba	Motsvarande logiska utgångsfunktion
0	000	$\bar{a}\bar{b}\bar{c}$
1	001	$\bar{a}\bar{b}c$
2	010	$\bar{a}b\bar{c}$
3	011	$\bar{a}bc$
4	100	$a\bar{b}\bar{c}$
5	101	$a\bar{b}c$
6	110	$a\bar{b}c$
7	111	abc

Kretsen kan sedan relativt enkelt byggas upp med 8 stycken AND-grindar och 3 stycken inverterare, en för varje ingångsvariabel enligt Fig. 4.

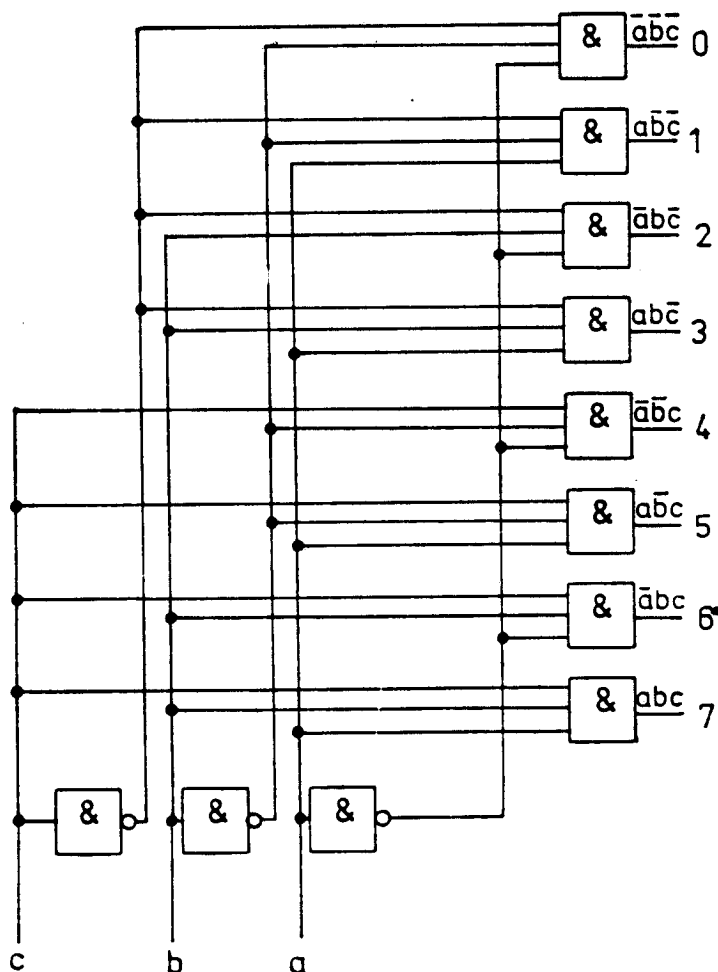


Fig. 4.

Den "direkta" avkodningstekniken som visades i Fig. 3 och Fig. 4 är ej så användbar då avkodarens storlek växer. Vid avkodning av 2-bitars tal erfordras $4 (=2^2)$ AND-grindar med 2 ingångar. Med 3-bitars tal krävdes $8 (=2^3)$ stycken 3-ingångars AND-grindar. Vid avkodning av exempelvis 8-bitars tal skulle det behövas $256 (=2^8)$ AND-grindar med 8 ingångar.

Man brukar i sådana fall använda sig av s k gruppavkodning, vilket innebär avkodning i flera etapper.

I exemplet ovan kan de åtta bitarna delas upp i två grupper om fyra stycken bitar. Dessa två grupper avkodas var för sig och kombineras sedan i ett avslutande steg. På så sätt kommer endast 4-ingångars AND-grindar att behövas. Fig. 5 visar principiellt förfarandet.

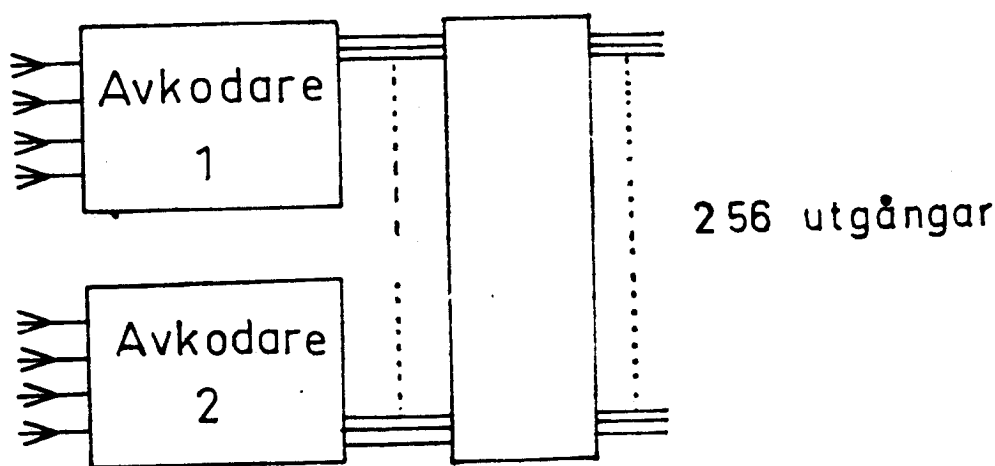


Fig. 5.

Avkodare för BCD-kod konstrueras på följande sätt. Avkodaren kommer att ha ingångarna a, b, c och d svarande mot BCD-kodens 4 bitar. Vi låter a beteckna kodens högra bit, osv. Avkodaren ska ha 10 utgångar svarande mot siffrorna 0-9.

Detta avkodningfall skiljer sig från de tidigare behandlade på så sätt att vi ej behöver avkoda alla de 16 kombinationer, som är möjliga med 4 ingångar. De 6 kombinationer som motsvarar talen 10-15 ingår ej i BCD-koden, då ju denna omfattar endast de decimala entals-siffrorna 0-9. Dessa 6 kombinationer behöver alltså avkodaren ej befatta sig med, och vi skall visa att detta förenklar konstruktionen.

Avkodaren kan beskrivas med 10 logiska funktioner, en för varje utgång. Vid bestämmandet av dessa är det lämpligt att använda Karnaughdiagrammet, då ju de kombinationer som inte ingår i BCD-koden kan införas som "don't care"-termer. Dessa är 1010, 1011, 1100, 1101, 1110 och 1111. (Obs ordningen dcba!)

Genom att alltså rita upp 10 stycken Karnaugh-diagram, ett för varje utgångsfunktion, kan vi nu bestämma de enklaste utgångsfunktionerna. Detta sker genom att göra så stora hoptagningar som möjligt med utnyttjande av \emptyset -termerna.

Vi skall som exempel visa detta genom att bestämma funktionerna för utgångarna 0, 7 och 9. Motsvarande logiska funktioner kallar vi Z_0 , Z_7 och Z_9 . Karnaugh-diagrammen återfinns i figur 6.

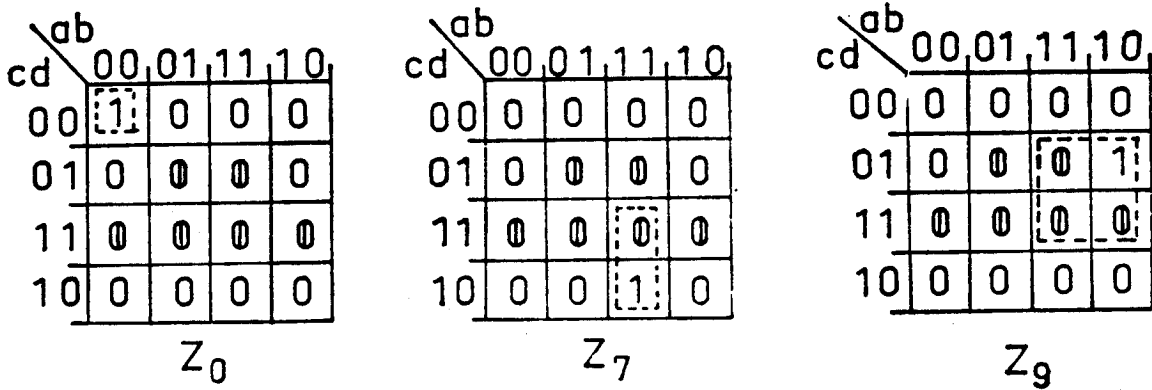


Fig. 6.

Genom att systematiskt undersöka alla de 10 utgångsfunktionerna erhålles:

$$\begin{array}{ll} Z_0 = \bar{a}\bar{b}\bar{c}\bar{d} & Z_1 = a\bar{b}\bar{c}\bar{d} \\ Z_2 = \bar{a}\bar{b}\bar{c} & Z_3 = a\bar{b}\bar{c} \\ Z_4 = \bar{a}\bar{b}c & Z_5 = a\bar{b}c \\ Z_6 = \bar{a}bc & Z_7 = abc \\ Z_8 = \bar{a}d & Z_9 = ad \end{array}$$

Vi får då kretsen i Fig. 7.

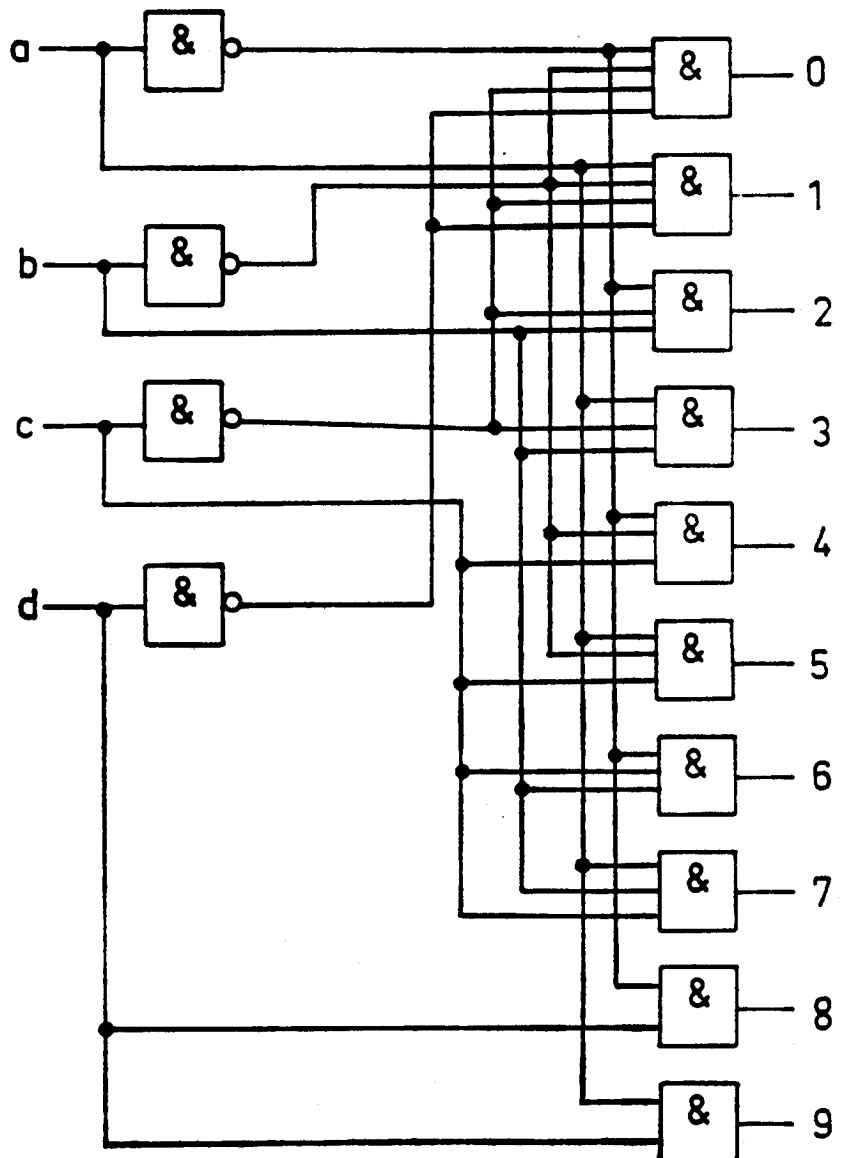


Fig. 7.. BCD-avkodare.

En s k BCD-7-segmentavkodare har 4 binära ingångar för BCD-kodade binära siffror men endast 7 utgångar. Utgångarna kopplas till en sifferindikator av segmenttyp, vilken indikerar de 10 decimala siffrorna 0, 1, 2 9. Segmentindikatorn har 7 st lysdioder A, B, C, D, E, F och G anordnade enligt Fig. 8. De decimala siffrorna erhålls genom att tända de lysdioder, vilka tillsammans bildar den sökta siffran. Se sammanställningstabellen i bilden.

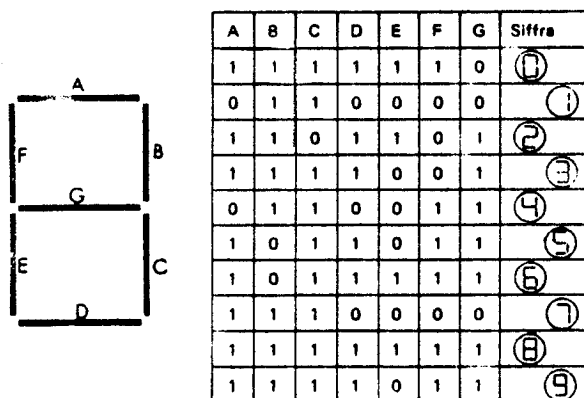


Fig. 8. Sjusegmentsdisplay.

BCD-7-segmentavkodaren kopplas till sifferindikator enligt Fig. 9. Transistorer, som klarar spänningen över lysdioden, ingår i avkodaren. Strömmen genom segmenten och transistoren är så stor att spänningen mellan kollektor-emitter är nästan noll medan spänningen över segmentet är ungefär +E. Transistorerna fungerar som en strömbrytare.

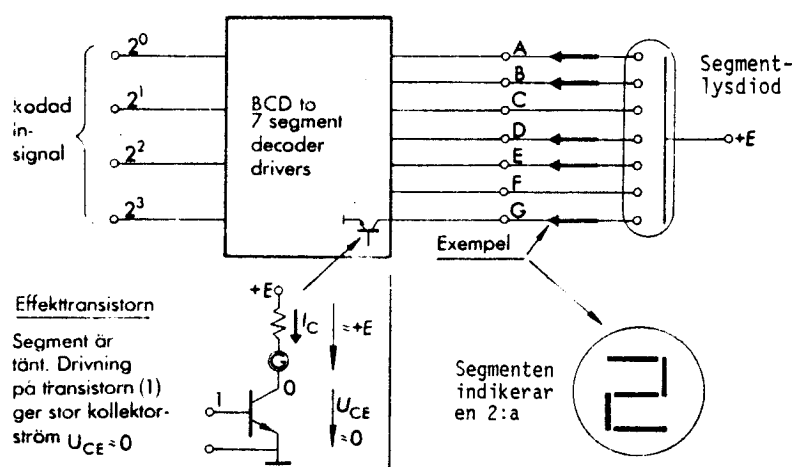


Fig. 9. BCD-7-segment-avkodare till en sjusegmentsdisplay.

Avkodaren tillverkas som en integrerad krets, vilken bl a innehåller en BCD-decimalomvandlare med 10 st NAND-grindar för de decimala siffrorna 0, 1, 2 ... 8, 9 samt 7 st NAND-grindar och effekttransistorer för sifferindikatorn 7 segment A, B och C.... G. Varje utgång på en decimal NAND-grind är kopplad till just de NAND-grindar och transistorer som behövs för att rätt segment skall tändas i sifferindikatorn.

UPPGIFT 1.

Du skall nu avkoda de binära talen i uppgift 11, ET21A och presentera resultatet i decimal form på en 7-segment-indikator. Använd avkodaren 9368 och display FND 500.

Avkodaren 9368 har fördelen att innehålla en konstantströmgenerator på utgångarna vilket innebär att inga strömbegränsningsmotstånd behövs till lysdioderna i indikatorn. 9368 innehåller även "latchar" på ingångarna. Ben nr 3 på 9368 betyder "latch enable" och lägges på noll i fortsättningen.

Koppla upp avkodare och indikator. Koppla därefter in utgångarna på 7490 till avkodaren och kontrollera att avkodaren ger rätt siffror på displayen.

Multiplexer (data selector).

En digital multiplexer är en kombinatorisk krets som med hjälp av en lämplig styrsignal väljer en av flera insignaler och låter den gå till utgången. En multiplexer utför således i princip samma funktion som en flerlägesomkopplare. Jämfört med avkodaren, som grenar ut ett fåtal insignaler till ett stort antal utsignaler, så sammanför multiplexern ett stort antal insignaler till ett fåtal utsignaler.

En 2-linje till 1-linje multiplexer motsvaras följaktligen av en enpolig tvåvägs omkopplare. (Fig. 10).

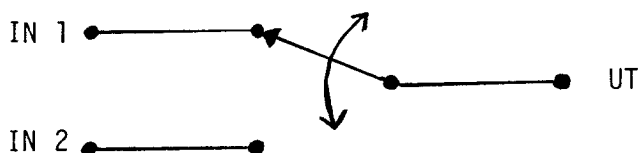


Fig. 10 .

En sådan multiplexer kan konstrueras med en kvadrupel tvåingångs NAND-krets 7400 (Fig.11).

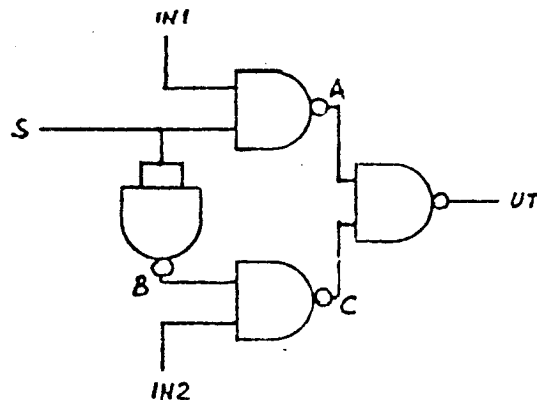


Fig. 11.

Sanningstabell.

S	IN1	IN2	A	B	C	UT
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	1	1	0
0	1	1	1	1	0	1
1	0	0	1	0	1	0
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	1	0	0	1	1

Ur tabellen framgår att då $S = 0$ blir $UT = IN2$ och då $S = 1$ blir $UT = IN1$.

Oftast köpes multiplexrar färdiga. Ett exempel är SN74157 som innehåller 4 st 2-linje till 1-linje. Den motsvaras av en fyrpolig tvåvägs omkopplare.

2. REGISTER.

Ett register i en digital konstruktion är avsett för lagring av information. Registret kan exempelvis hålla i minnet de tal som skall matas in för bearbetning och det resultat som beräkningarna gav. Informationen är kodad till ettor eller nollor efter lämplig kod. Ett register består av ett antal vippor, inbördes ordnade på ett visst sätt. Varje vippa är ett korttidsminne för ettor-nollor och kan endast lagra en bit varför man måste ha lika många vippor i registret som det största antal bitar man kommer att använda. Vanligen seriekopplar man ett antal vippor efter varandra (Fig.12).

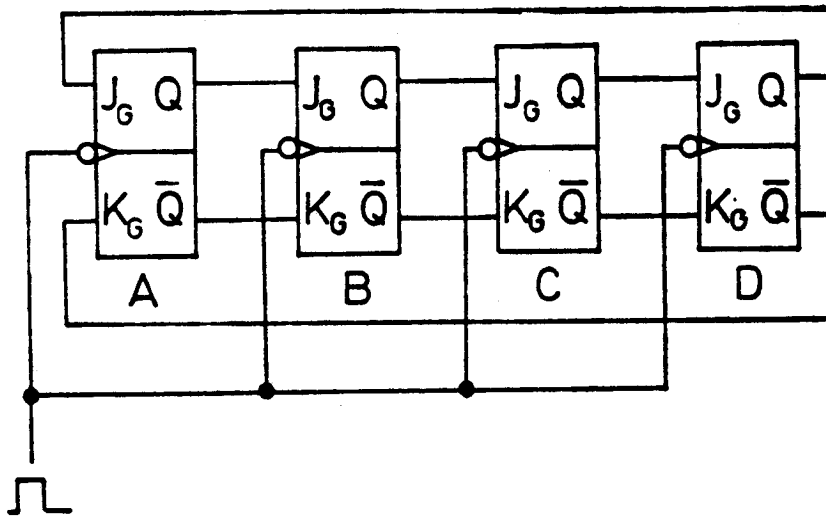


Fig. 12. 4-bits skiftregister.

Vi har här 4 vippor, A, B, C och D kopplade så att Q-utgången på vipporna A, B och C utgör J-ingången på efterföljande vippa. Dessutom gäller att \bar{Q} -utgångarna utgör K-ingångar på efterföljande vippa. D-vippans utgångar utgör på samma sätt ingångar till vippa A.

Vi antar exempelvis att vipporna från början har innehållet ABCD = 1001. Beteckningen ABCD = 1001 innebär då att $Q_A = 1$, $Q_B = 0$, $Q_C = 0$ och $Q_D = 1$. Vad händer då första pulsen kommer?

Vi vet att $Q_A = J_B$, $\bar{Q}_A = K_B$, $Q_B = J_C$, $\bar{Q}_B = K_C$ etc. (J_B och K_B betyder naturligtvis J- resp. K-ingången på vippa B osv).

Med antagandet ABCD = 1001 gäller då före den första pulsen:

$$J_A = Q_D = 1 \quad J_B = Q_A = 1 \quad J_C = Q_B = 0 \quad J_D = Q_C = 0$$

$$K_A = \bar{Q}_D = 0 \quad K_B = \bar{Q}_A = 0 \quad K_C = \bar{Q}_B = 1 \quad K_D = \bar{Q}_C = 1$$

Då erhåller man efter första pulsen:

$$J_A K_A = 10 \text{ ger } Q_A = 1$$

$$J_B K_B = 10 \text{ ger } Q_B = 1$$

$$J_C K_C = 01 \text{ ger } Q_C = 0$$

$$J_D K_D = 01 \text{ ger } Q_D = 0$$

Nu blir alltså ABCD = 1100.

Det vi har gjort är att vi har skiftat innehållet i vipporna ett steg åt höger. Dessutom har innehållet i vippan längst till höger hamnat i vippan längst till vänster.

Man inser nu att då nästa puls (eller skiftpuls, som den brukar kallas i detta sammanhang) kommer, ändras vippornas innehåll till ABCD = 0110.

Efter tredje skiftpulsen blir då vippornas innehåll ABCD = 0011 och efter den fjärde pulsen är vi tillbaka i utgångsläget nämligen ABCD = 1001.

Skiftregistret i Fig. 12 är ett sk återkopplat skiftregister, eftersom informationen i vippa längst till höger överföres till vippa längst till vänster. Det normala förfaringssättet med högerskiftande skiftregister är annars att innehållet i vippa längst till höger försvinner efter skiftningen, önskar man läsa ut talet 1001 ur registret för vidare bearbetning ger man alltså skiftregistret skiftpulser och på vippa D:s Q-utgång uppträder pulsföljden (Fig. 13 a).

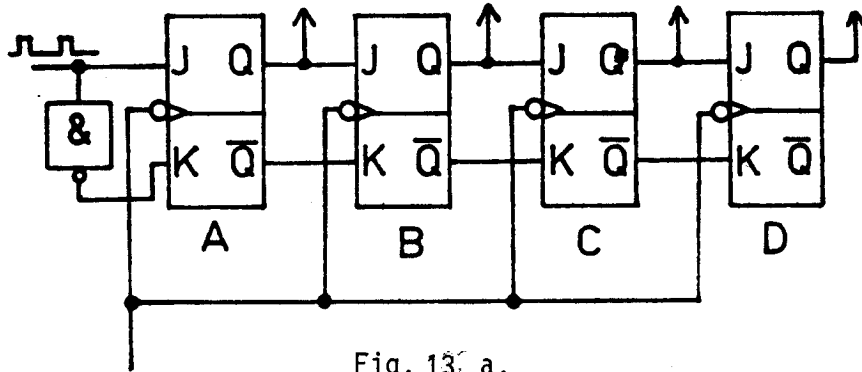


Fig. 13 a.

Vi antar som exempel att det binära talet som skall mottagas är 1010. Motsvarande binära pulståg visas i fig. 13 b. Observera också att den binära signalens pulser infaller samtidigt (synkront) med klockpulserna.

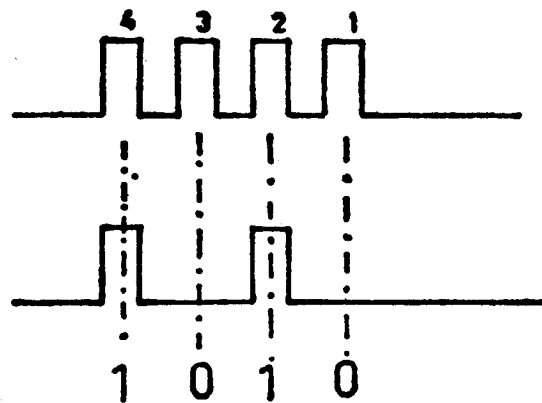


Fig. 13 b.

Då klockpuls 1 kommer, flyttas samtidigt biten längst till höger i den binära signalen till vippa A:s ingångar. Eftersom denna bit är en nolla kommer vippa A att förbli i nolläge efter första skiftpulsen. Samtidigt sker skiftning av det "gamla" innehållet i vippa A till B, från B till C och från C till D. Vippa D:s innehåll försvinner efter skiftningen. Eftersom alla vipporna var nollställda från början innehåller de fortfarande enbart nollor.

Då andra skiftpulsen inträffar, ligger en logisk etta på A-vippans J-ingång. Efter andra skiftningen innehåller vippa A en etta, medan de övriga innehåller nollor.

Genom att utvidga detta resonemang inser man att efter 4 skiftpulser ligger talet 1010 lagrat i vipporna. Tabellen nedan visar vippornas innehåll efter varje skiftpuls.

Antal skiftpulser	Vippornas innehåll			
	A	B	C	D
0	0	0	0	0→
1	<u>0</u>	0	0	0→
2	<u>1</u>	0	0	0→
3	<u>0</u>	1	0	0→
4	<u>1</u>	0	1	0

De understrukna siffrorna markerar de binära siffror som i tur och ordning skiftas in i registret. Pilarna längst till höger indikerar att vippa D:s innehåll försvinner efter varje skiftning. Nu ligger alltså talet 1010 i registret. Om man önskar läsa ut detta tal i serieform för vidare bearbetning, ger man skiftregistret skiftpulser och på vippa D:s Q-utgång uppträder då pulsföljden.

Skiftregister är synnerligen lämpade att mottaga och/eller utsända information i serieform. Skiftregister används också för omvandling från parallell- till serieform eller tvärtom. Register är en viktig beståndsdel i apparatur för binär addition och subtraktion liksom för multiplikation och division. Om ett tal skiftas ett steg åt vänster innebär detta en multiplikation med 2. Två steg motsvarar 4 gånger etc. Om talet skiftas ett steg åt höger betyder det en division med 2.

Register tillverkas numera endast i integrerad teknik i form av monolitiska planarkretsar eller med fälteffekttransistorer av MOS-typ. MOS-transistorerna har en enklare uppbyggnad än planartyperna.

Uppgift 2.

Undersökning av ett 8-bitars skiftregister 74198 (se TTL-BOKEN).

Koppla upp kretsen enligt följande:

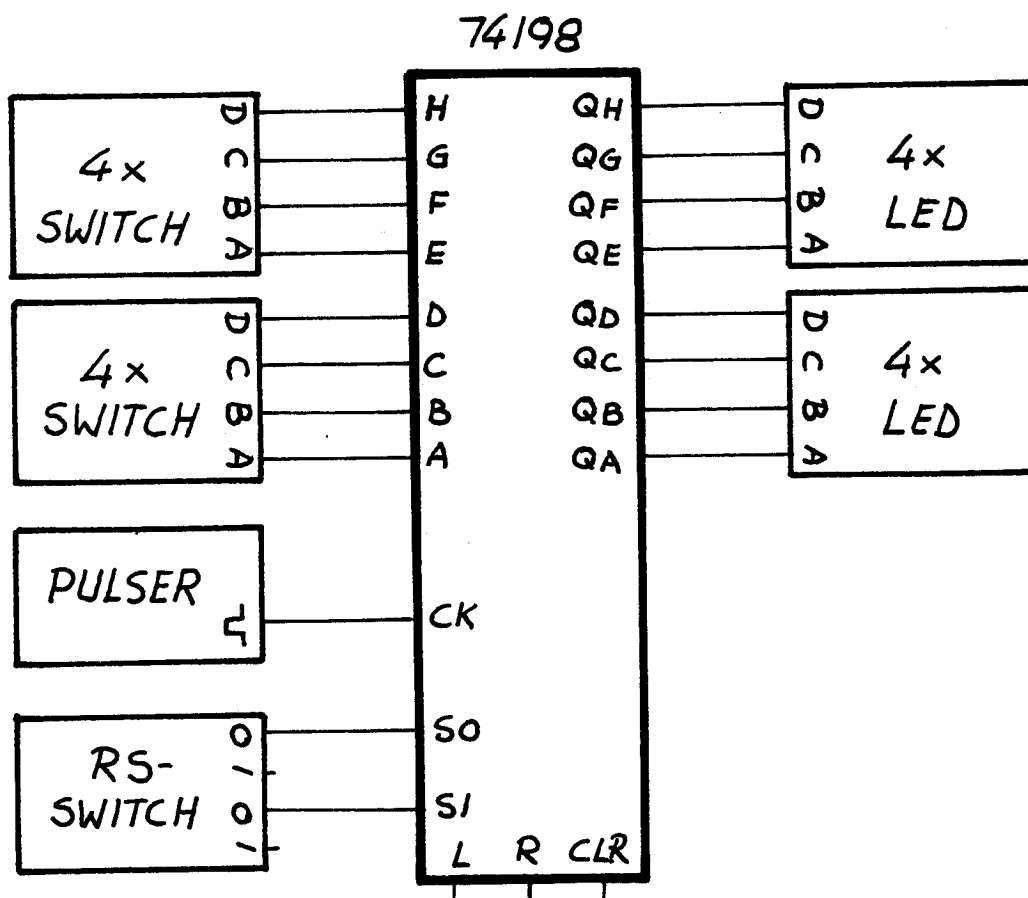


Fig. 21.

Ställ dataingångarna

HGFEDCBA = 1000 0001

Med S0 och S1 i 1-läge mata in en klockpuls.

På utgångarna erhålles

Gör vänsterskift.

S0 = S1 = L = R =

Efter 7 klockpulser visar utgångarna

Gör högerskift.

S0 = S1 = L = R =

Efter 6 klockpulser visar utgångarna

SHIFT LEFT INPUT och SHIFT RIGHT INPUT användes för

Förklara innebörden (numeriska) av 1 stegs vänsterskift

Dito högerskift

3. ADDERARE.

Vid addition av två binära tal gäller:

$$\begin{array}{l} 0 + 0 = 0 \quad 1 + 0 = 1 \\ 0 + 1 = 1 \quad 1 + 1 = 10 \end{array}$$

Addition av decimala tal ger en minnessiffra (carry) om summan är större än 9.

Addition av binära tal ger tydligen en minnessiffra om man adderar 1 + 1.

I tabellen visas binär och decimal addition av talen $A = 19_{10}$ och $B = 27_{10}$.

	Binary						Decimal	
	2^5	2^4	2^3	2^2	2^1	2^0	10^1	10^0
Carry-in	1	0	0	1	1	—	1	—
<i>A</i>	0	1	1	0	1	1	1	9
<i>B</i>	0	1	0	0	1	1	2	7
Sum	1	0	1	1	1	0	4	6

Observera att ingen minnessiffra kan förekomma i den minst signifikanta positionen. Bortsett från denna position måste en logisk krets kunna addera tre bitar i varje position, nämligen de två "originalbitarna" plus en eventuell minnessiffra.

En krets som kan utföra en addition av två bitpositioner och en inkommande minnessiffra och vars utgångar ger delsumman och utgående minnessiffra kallas heladderare (full adder). En halvadderare (half adder) är en krets som adderar två bitar. Den har utgångar för summa och minnessiffra men saknar ingång för inkommande minnessiffra. Den lämpar sig således för addition i den minst signifikanta positionen eftersom ingen minnessiffra kan förekomma i denna position. I Fig. 15 visas blockschemamässigt hur addition av två 4-bits tal ($a_3a_2a_1a_0$ och $b_3b_2b_1b_0$) sker med hjälp av fyra adderare. De olika minnessiffrorna betecknas med c . Resultatet av additionen visas av s -utgångarna där s står för summa. Bitarna $S_3S_2S_1S_0$ ger det korrekta resultatet.

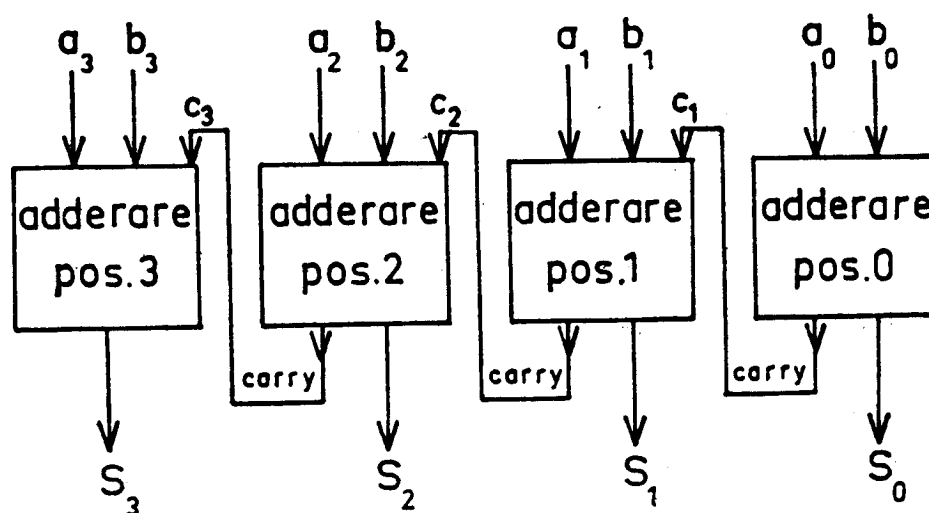


Fig. 15. Parallelladderare.

Halvadderaren.

En halvadderare kan addera två binära tal och på utgångarna presentera talens summa och minnessiffra. Den kan däremot inte ta emot och behandla information från en föregående adderande krets. Dess sanningtabell visas i Fig.16 och ur sanningstabellen framgår att

$$s = \bar{a}b + a\bar{b} = a \oplus b$$

$$c_{ut} = ab$$

a	b	s	c _{ut}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Fig.16 .

Den exkluderande (uteslutande) ELLER-grindens funktion motsvarar således summafunktionen. För att klara minnessiffran kopplas en OCH-grind parallellt med den exkl. ELLER-grinden (se Fig. 17).

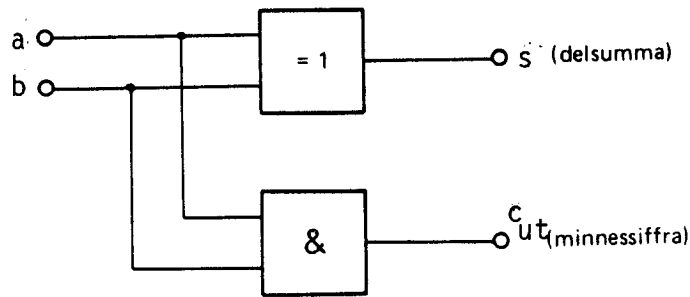


Fig. 17.

Heladderaren.

En heladderare skall kunna addera två tal a och b och till dessa en minnessiffra. Dessutom skall eventuellt uppkommande minnessiffra kunna tas ut.

a	b	c_{in}	s	c_{ut}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig. 18.

Heladderarens sanningstabell visas i Fig. 18. Ur sanningstabellen framgår att en summaterm på 1 uppstår för ett udda antal ettor. De tre variablerna ger tre kombinationer med en etta och två nollor och en kombination av tre ettor. Summan kan uttryckas som:

$$s = \bar{a}\bar{b}c_{in} + \bar{a}b\bar{c}_{in} + a\bar{b}\bar{c}_{in} + abc_{in}$$

Minnessiffran c_{ut} erhålles som:

$$c_{ut} = \bar{a}bc_{in} + a\bar{b}c_{in} + ab\bar{c}_{in} + abc_{in}$$

Uttrycket för c_{ut} kan förenklas med Karnaughdiagram (Fig. 19) till

$$c_{ut} = ab + ac_{in} + bc_{in}$$

När vi nu kan uttrycka s och c_{ut} i ingångsvariablerna a, b och c_{in} är det möjligt att realisera adderarblocken i Fig. 15 med ett rent kombinatoriskt nät.

		ab			
		00	01	11	10
c_{in}	0	0	0	1	0
	1	0	1	1	1

Fig. 19.

Av två halvadderare kan emellertid också en heladderare byggas upp. Fig. 20 visar ett blockschema för ett sådant system. Den vanligaste metoden att konstruera heladderare är just den ovan beskrivna. Man utför alltså heladdition genom två successiva halvadditioner.

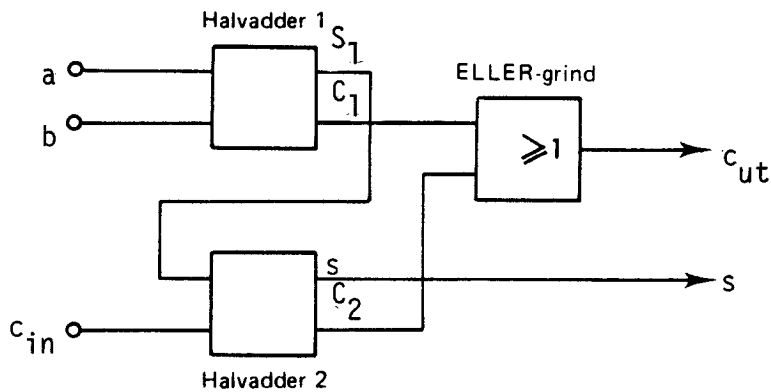


Fig. 20.

UPPGIFT 3.

Resonera igenom heladderarens funktion i Fig. 20 och fyll i tabellen.

a	b	c _{in}	s ₁	c ₁	c ₂	s	c _{ut}
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

UPPGIFT 4.

Koppla upp en halvadderare. Som OCH-grind används 2 st NAND-grindar (7400).

Rita först ett kopplingsschema nedan med logiska symboler.

Utför de logiska operationerna och fyll i sanningstabellen. Kom ihåg att "1"-ställa de ingångar som inte används.

a	b	s	c _{ut}

För addition av flersiffriga binära tal finns två olika metoder, serieaddition och parallelladdition. Serieaddition utförs på följande sätt:

De minst signifikanta siffrorna i de båda talen adderas i vanlig ordning först:

$$\begin{array}{r} a: \quad 1001 \\ b: \quad + 1101 \\ \hline \quad \quad 0 \\ c_{ut}: \quad 1 \end{array}$$

Summan av dessa två tal, i vårt exempel 0, läggs in i ett register. Eventuell minnessiffra lagras i ett minne för att i nästa delmoment adderas till summan av de näst minst signifikanta siffrorna, osv. Som register används ett skiftregister. Minnet som krävs för lagring av minnessiffrorna, utgörs av en bistabil vippa.

De siffror som kommer in i registret, skiftas ett steg åt höger för varje klockpuls. När den successiva additionen är utförd efter fyra klockpulser, innehåller registret talens summa. Denna summa kan lämpligen presenteras på en till registerutgången ansluten indikator. Notera att vid serieaddition åtgår endast en heladderare, oavsett antalet siffror i de tal som skall adderas. Fig. 22 visar blockschemat för en serieadderare.

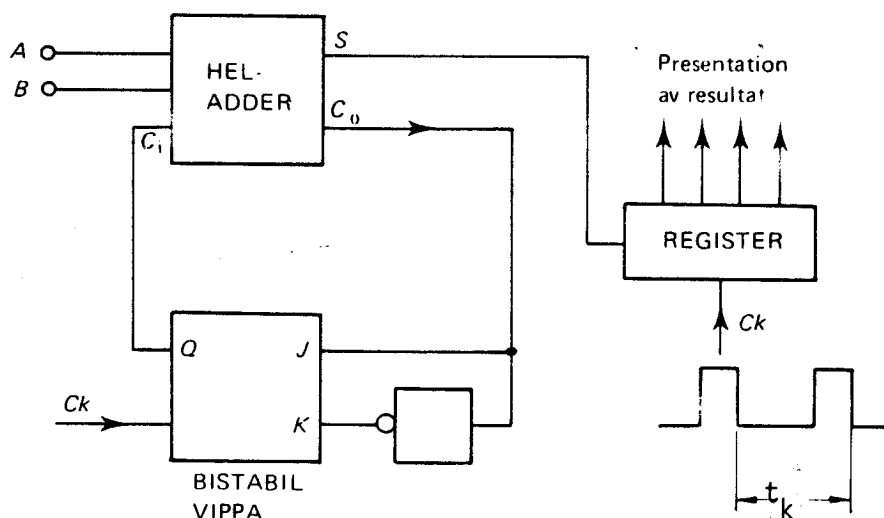


Fig. 22.

För varje addition åtgår i tid ett klockpulsintervall (t_k). Om de tal som skall adderas innehåller många siffror, kommer räkneoperationen att behöva motsvarande längre tid. Registerfunktionen måste anpassas till det antal siffror som de adderade talen innehåller. Fysiffriga tal kräver i princip register med fyra minnesbitar.

Vid parallelladdition kopplas som förut beskrivits ett antal heladderare parallellt med varandra (jfr Fig. 15). En heladderare åtgår per siffra.

I gengäld behövs inte något register. En komplett addition kräver tidsmässigt endast ett klockpulsintervall, varför parallelladderaren generellt blir snabbare än serieadderaren

4. RAM-MINNE.

En vippa är ett typiskt minneselement. Vi skall ge ett exempel på detta. Antag att vi i en krets vill lagra värdet av en logisk variabel a i ett visst ögonblick, för att efter en tid använda oss av detta värde för någon beräkning.

Lagring i ett minne brukar benämnas inskrivning, och då man plockar ut vippans innehåll för användning kallas detta utläsning. Skrivning och läsning kan utföras av kretsen i Fig. 23.

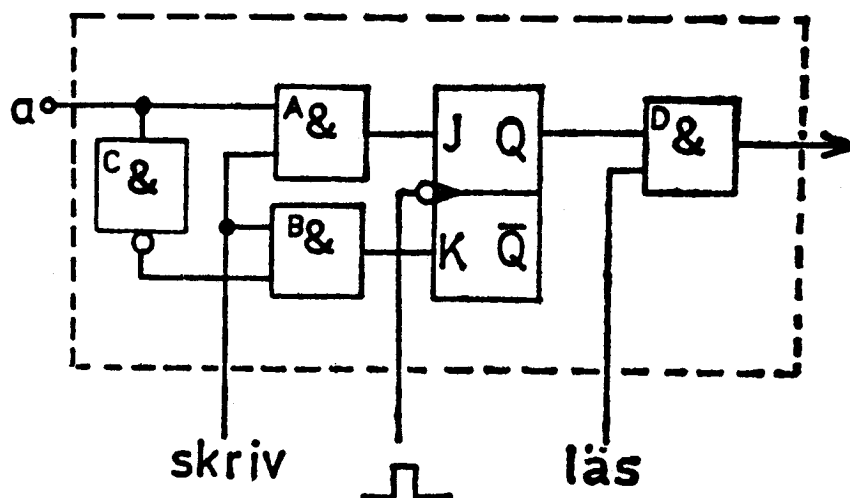


Fig. 23.

Ledningen märkt skriv ligger normalt på nollnivå. Detta medför att vippan ej påverkas av värdet av a , eftersom de bägge AND-grindarnas (A och B) utgångar ligger konstant på nollnivå.

Om vi nu vill skriva in värdet av variabeln a i vippan lägger vi en logisk etta på skrivningången. Om $a = 1$ kommer J-ingången att få ettnivå och K-ingången kommer därvid, tack vare inverteraren C, att få nollnivå.

Om $a = 0$ blir J-ingången nollställd och K-ingången ettställd. Om nu en puls kommer in på pulsingången, så antar vippans Q-utgång a -ingångens värde ty $a = 1$ ger $JK = 10$ och $a = 0$ ger $JK = 01$. Om vi nu åter lägger en nolla på skrivledningen kan vippan ej påverkas ytterligare vid eventuella pulser. Värdet på a kommer alltså att ligga lagrat ända tills vi gör nästa inskrivning genom att ettställa skrivningången.

Eftersom även läsledningen normalt ligger på nollnivå kommer minneskretsens utgång att normalt vara nollställd. Då man önskar "känna av" vippans innehåll ettställes läsningången och vippans innehåll, dvs Q-utgångens tillstånd, blir därmed även tillståndet på kretsens utgång efter grind D.

Fig. 24 visar en minnesmatris för 16 bitar med 16 vippor vilka placerats i en 4X och 4Y matris. (X_0Y_0 , X_1Y_1 , X_2Y_2 och X_3Y_3 (CD 2155). Den integrerade kretsen är placerad i en flat kapsel.

Den integrerade minnesmatrisen CD 2155 heter också RAM, eller RAM-minne i dagligt tal. RAM är ett sammandrag av orden Random Access Memory vilket betyder minne med konstant accesstid. Utvecklingen av tillverkningsmetoderna för halvledarelement har gått mot en standardisering med så få olika element som möjligt. RAM-minnet är ett sådant grundelement. Accesstiden för den visade CD 2155 är 6,5 ns. RAM-minnet har cellerna orienterade i X- och Y-riktning (Fig. 24).

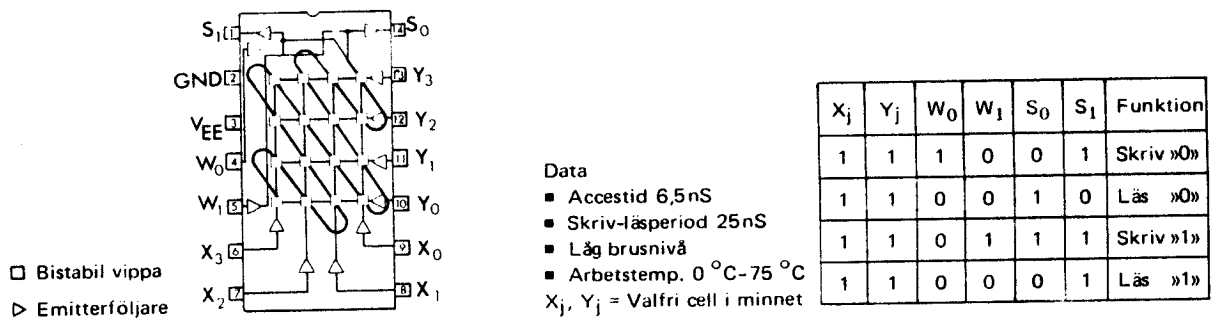


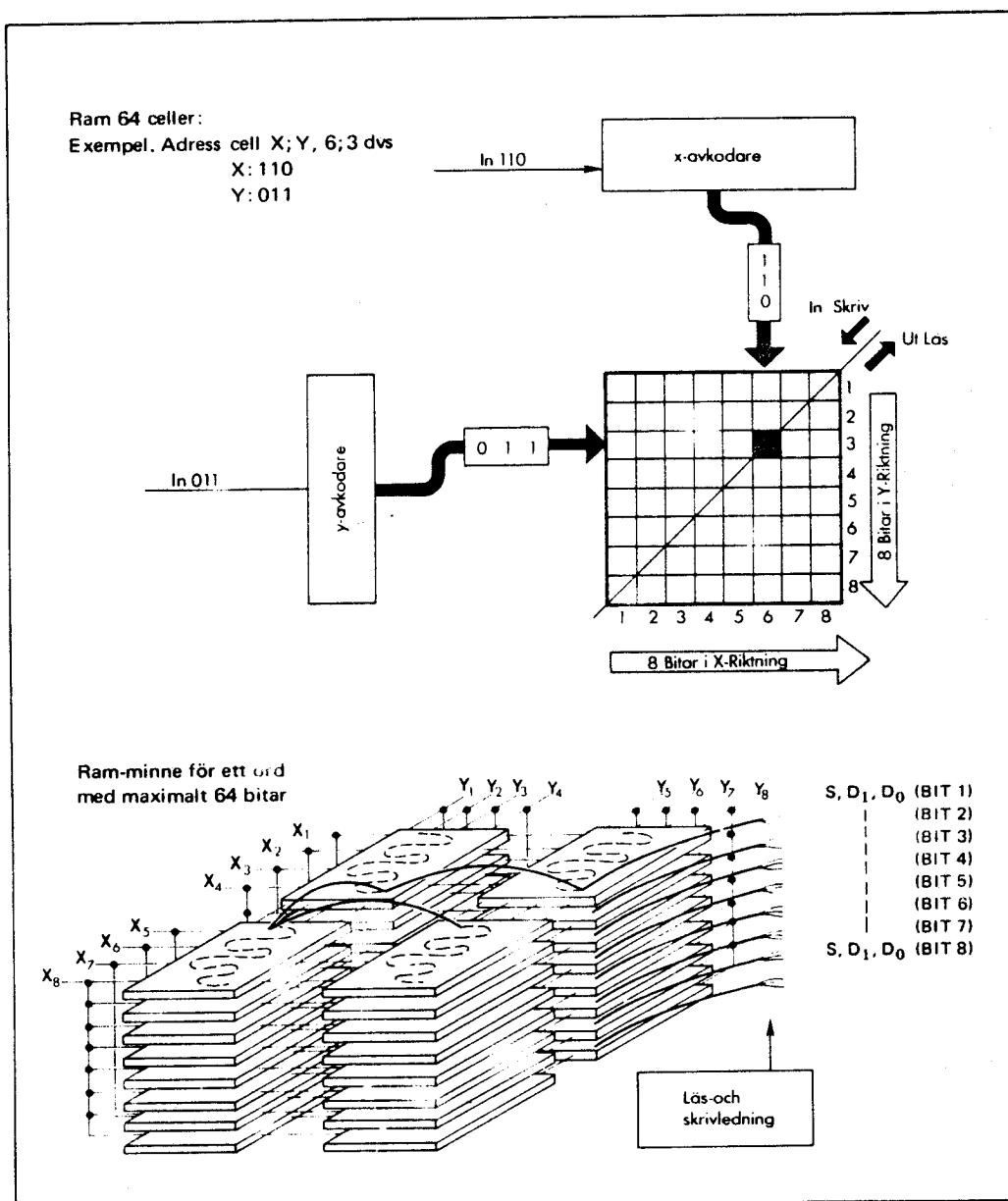
Fig.24 .

Adressering (bestämning) av en viss cell i minnet erhålls genom att ge de önskade X-Y-linjerna en logisk "1" och alla övriga linjer en "0". Med W_0 - och W_1 -ingångarna i läge "0" ger den valda minnescellen en ström till antingen läsledning "0" eller "1" beroende på vilken information som finns lagrad i den adresserade cellen.

Avläsningsströmmen är sedan förstärkt och pulsformad för att ge en logisk "1"-nivå till antingen utgång S_0 eller S_1 . Avläsning av en cell i matrisen förstör icke innehållet i minnescellen eftersom ingen ändring av minnesvippan uppstår. Man kan således läsa av matrisen hur många gånger som helst utan att innehållet ändras. Denna teknik kallas NDRO-operation (Non Destructive Read Out = icke förstörande avläsning).

Inskrivning av logisk "1" eller "0" göres genom att ge en logisk "1" till de önskade X-Y-linjerna och samtidigt ge en logisk "1" till W_0 eller W_1 -ingången. (Studera sanningstabellen). Man gör detta genom en adress-signal i X-riktningen och en i Y-riktningen från en X-avkodare och en Y-avkodare. RAM-minnet kan endast adresseras med en cell i taget. När adresseringen har aktiverat en cell i minnet, läser man av minnet eller skriver in en ny information i cellen längs läsledningen.

Man skall lägga märke till att en minnesmatris endast skriver in en bit (0 eller 1). För att få en RAM-matris att arbeta fordras en adressinformation i X-Y-riktning och en läs- eller skriv-information längs läsledningen. Vill man skriva in ett ord på 4,8 eller 16 bitar så parallellkopplar man så många RAM-minne som det antal bitar ordet innehåller. Man skriver in informationen, ordet, på samma plats i varje RAM-minne. Detta är möjligt genom parallellkopplingen. Samma adress erhålls till varje RAM-minne, och ordet läses ut parallellt med en bit från varje RAM-minne. Fig. 25 visar adresseringen av ett RAM på 8X och 8Y.



Adressering av ett ram-minne på 64 celler och 32 ram-matriser för ett ord på 64 bitar

Fig. 25 .

Adress-signalerna behöver endast ha 3 bitar ty 001 motsvarar 1, 111 motsvarar 7 och 000 motsvarar 8. Dessutom visar bilden 4 staplar minnen med 8 RAM-element i varje stapel. Varje RAM-minnesmatris är på 16 celler (bitar), 4X och 4Y, På de 32 RAM-matriserna kan man således skriva in ett ord med max 64 bitar.

Uppgift 6.

Undersökning av integrerat 64-bitars RAM-minne. DM8599/74189.

Minnet är organiserat i 16 ord à 4 bitar. Det är försett med inbyggd avkodning för adressingångarna vilket innebär att antalet adressingångar är fyra.

Antalet in- och utgångar är lika med antalet bitar i ordet, dvs fyra (se Fig. 26).

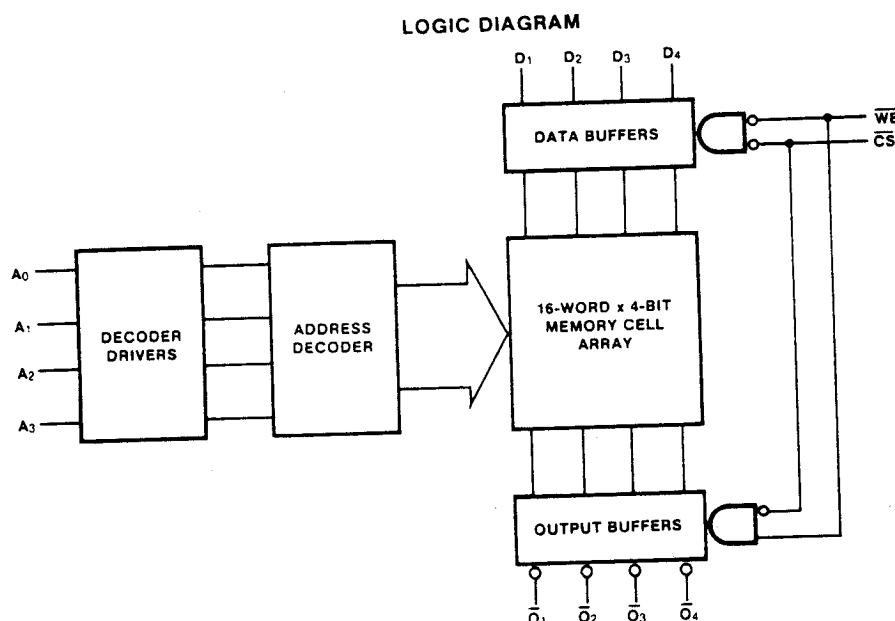
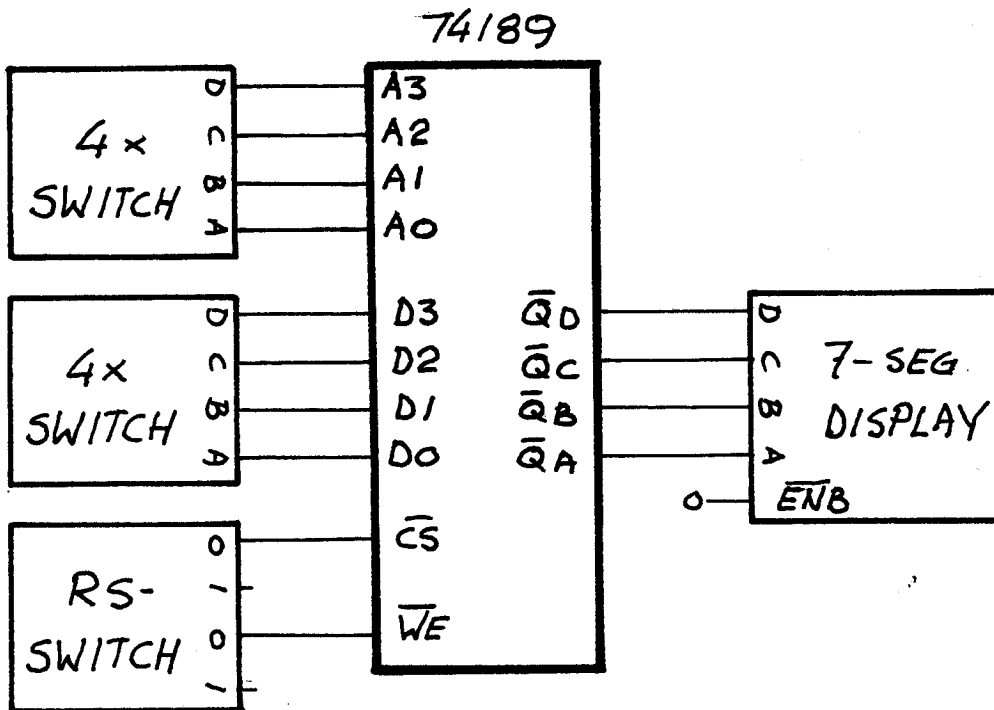


Fig. 26.

Funktioner.

Minnet är försett med två ENABLE-ingångar, MEMORY ENABLE (\overline{ME}) eller CHIP SELECT (\overline{CS}) och WRITE ENABLE (\overline{WE}). (ENABLE = sätta i stånd, möjliggöra) 1-ställning av \overline{CS} innebär att såväl skrivning i minnet som läsning av dess innehåll förhindras. 1-ställning av \overline{WE} innebär att skrivning i minnet förhindras.

Koppla upp enligt RAM-minnet enligt följande.



Nollställning av minnet.

$\overline{CS} = 0$ (minnet inkopplat)

$\overline{WE} = 1$ (skrivning förhindras)

Ställ de fyra data-ingångarna på 0, ty nollställningen innebär att samtliga adresser skall inskrivas med 0.

Ställ in adressen 0, dvs $A = B = C = D = 0$. Ställ kortvarigt $\overline{WE} = 0$. Härvid sker skrivning av datainformationen 0 i adress D.

Fortsätt med de resterande 15 adresserna och skriv in 0 i samtliga.

Nollställning av minnesinnehållet har nu utförts.

Skrivning i minnet.

$\overline{CS} = 0$ och $\overline{WE} = 1$

Välj önskad adress. Ställ in det data-ord som skall skrivas in på denna adress.

Ställ kortvarigt $\overline{WE} = 0$. Härvid skrivs data-ordet in i den valda adressen.

Läsning ur minnet.

När skrivning i minnet verställt (max. 16 ord om 4 bitar/ord), kan läsning av innehållet i respektive adress göras.

$$\bar{CS} = 0$$

$$\bar{WE} = 1$$

Ställ in önskad adress.

Adressernas innehåll kan nu läsas ut via 7-seg. DISPLAY'en.

OBSERVERA: \bar{WE} får inte 0-ställas under läsningen. Om så sker, inläses på den valda adressen det data-ord som för tillfället är inställt.

Skriv in i minnet enligt tabell.

Cell nr	Adressgång				Dataingång				
	D	B	C	A	Decim	D_D	D_C	D_B	D_A
0					15				
1					14				
2					13				
3					12				
4					11				
5					10				
6					9				
7					8				
8					7				
9					6				
10					5				
11					4				
12					3				
13					2				
14					1				
15					0				

Kontrollera att Du har skrivit in rätt genom att läsa ut minnesinnehållet i samtliga ord. Vid utläsning är $\bar{CS} = \dots\dots\dots$ och $\bar{WE} = \dots\dots\dots$

Ändra innehållet i cell nr 10 till 10 samt innehållet i cell nr 5 till 5. Beskriv tillvägagångssättet!

.....

54S/74S189 54LS/74LS189

64-BIT RANDOM ACCESS MEMORY (With 3-State Outputs)

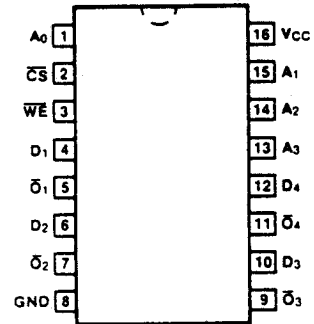
DESCRIPTION — The '189 is a high speed 64-bit RAM organized as a 16-word by 4-bit array. Address inputs are buffered to minimize loading and are fully decoded on-chip. The outputs are 3-state and are in the high impedance state whenever the Chip Select (\overline{CS}) input is HIGH. The outputs are active only in the Read mode and the output data is the complement of the stored data.

- 3-STATE OUTPUTS FOR DATA BUS APPLICATIONS
- BUFFERED INPUTS MINIMIZE LOADING
- ADDRESS DECODING ON-CHIP
- DIODE CLAMPED INPUTS MINIMIZE RINGING

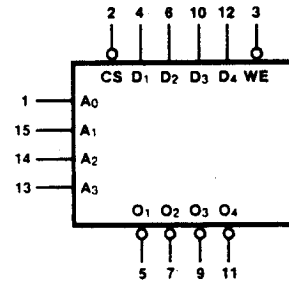
ORDERING CODE: See Section 9

PKGS	PIN OUT	COMMERCIAL GRADE	MILITARY GRADE	PKG TYPE
		$V_{CC} = +5.0\text{ V} \pm 5\%$, $T_A = 0^\circ\text{C to } +70^\circ\text{C}$	$V_{CC} = +5.0\text{ V} \pm 10\%$, $T_A = -55^\circ\text{C to } +125^\circ\text{C}$	
Plastic DIP (P)	A	74S189PC, 74LS189PC		9B
Ceramic DIP (D)	A	74S189DC, 74LS189DC	54S189DM, 54LS189DM	6B
Flatpak (F)	A	74S189FC, 74LS189FC	54S189FM, 54LS189FM	4L

CONNECTION DIAGRAM PINOUT A



LOGIC SYMBOL



V_{CC} = Pin 16
 GND = Pin 8

INPUT LOADING/FAN-OUT: See Section 3 for U.L. definitions

PIN NAMES	DESCRIPTION	54/74S (U.L.) HIGH/LOW	54/74LS (U.L.) HIGH/LOW
$A_0 - A_3$	Address Inputs	0.63/0.16	0.5/0.013
\overline{CS}	Chip Select Input (Active LOW)	0.63/0.16	0.5/0.013
\overline{WE}	Write Enable Input (Active LOW)	0.63/0.16	0.5/0.013
$D_1 - D_4$	Data Inputs	0.63/0.16	0.5/0.013
$\overline{O}_1 - \overline{O}_4$	Inverted Data Outputs	162/10 (50)	10/10 (5.0)

FUNCTION TABLE

INPUTS		OPERATION	CONDITION OF OUTPUTS
\overline{CS}	\overline{WE}		
L	L	Write	High Impedance
L	H	Read	Complement of Stored Data
H	X	Inhibit	High Impedance

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

Beträffande 3-state.

Både adress- och dataledningar och vissa kontrollsignaler är ofta av tri-state (3-state) typ dvs förutom den höga och låga nivån kan utgången fränkopplas (inta ett högimpedanstillstånd). Fränkopplingen av en utgång sker med en extra styringång ofta kallad "enable" eller "tristate". Tristate-principen kan tillämpas på alla typer av grindar. "Tristate" innebär ju endast att en utgång fränkopplas och därmed ej påverkar den på utgången anslutna signalledningen. Princip och grafisk symbol visas i Fig. 27.

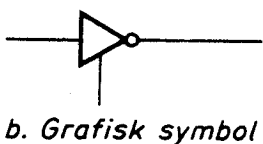
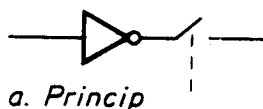


Fig. 27.