

ABC PIO MANUAL



SCANDIA **METRIC** AB

BANVAKTSVAGEN 20, BOX 1307, 171 25 SOLNA, TEL 08/82 04 00



***** INNEHÅLL *****

Sida

- 1 Introduktion
- 2 Tekniska data
- 3 Beskrivning
- 4 Stiftsplacering
- 5 Kortval, byglingar
- 6 Kommandon, kontroll
- 7 Kommandon, data
- 8 Inkoppling av digital våg
- 9 Drivprogram digital våg
- 10 Färdiga drivprogram
- 11 Installation
- 12 Placeringsschema
- 13 Kopplingsschema

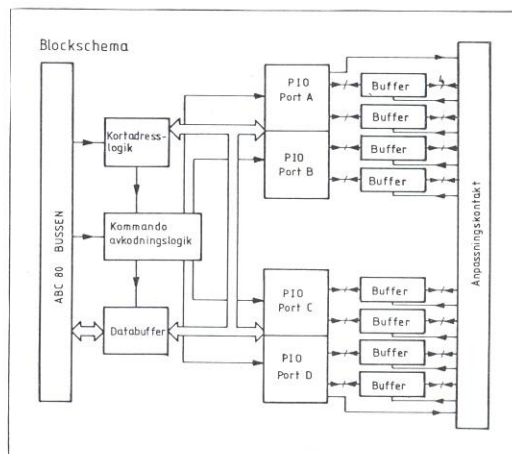
A B C P I O

ABC PIO är ett anpassningskort för parallella digitala signaler om 32 bitar. Dessa har TTL-nivå, vilket innebär att spänningen < 0.8 V ger en logisk nolla "0" och 2.4V - 5.5V ger en logisk etta "1". Kortet kan valfritt användas för ut- eller inmatning av TTL-signaler i grupper om fyra bitar. All datahantering sker i grupper om 8 bitar, men i programmet kan man behandla enskilda bitar.

----- ANVÄNDNINGSSOMRÅDE -----

ABC PIO kan användas i bl.a. följande applikationer:

- === Digitala mätinstrument och vågar med parallell BCD-anslutning
- === Remsläsare och remsstans
- === Skrivare med parallellformat
- === Statuskontroll av signalnivåer, t.ex. larmsignaler
- === Styrning av digital utrustning, t.ex. multiplexer, stegmotorer
- === Styrning av digitala nivåer, t.ex. reläfunktioner
- === Hantering av externa A-D och D-A-omvandlare



----- UPPBYGGNAD -----

ABC PIO är uppbyggd kring två parallellkretsar (Z80 PIO) som har 2x8 bitar var. Dubbelriktade buffertkretsar sitter mellan PIO:n och kontakten mot omgivningen. Vidare ingår logik för avkodning av kortadress och de olika kommandona, samt en 8 bitars databuffer. Buffertriktningen styrs genom byglingar på anpassningskontakten. Programmet skall först initiera respektive port för in eller ut-riktning. Denna riktning skall vara samma som byglingen för respektive buffer. Därefter kan man sända och ta emot data, till och från omgivningen.

----- GENERELLA DRIVROUTINER -----

Till ABC PIO finns det färdiga generella drivrutiner som underlättar programmeringen. Dessa drivrutiner, som hanteras från BASIC, är av speciell betydelse när det krävs snabba avläsningar.

Exempel på generella drivrutiner är:

- == Parallell-BCD-rutin för olika antal siffror
- == Remsläsar-rutin

TEKNISKA DATA ABC PIO

=== ALLMÄNT ===

Spänningsmatning : +5V
 Strömförsörjning : Min 420 mA (vid alla buffrar inåt)
 : Ström tillkommer beroende på extern last
 Storlek : Europakort 100*160 mm
 Bussanslutning : Till ABC-80-bussen
 Kortadresser : 6 byglingar ger 64 möjliga adresser
 Kontaktdon : 96-pin Europadon, DIN 41612

=== IN/UT-GÅNGAR ===

Portar : 4 st om 8 bitar
 Riktning : Grupper om 4 bitar väljs ut eller in

Signalnivåer : TTL-signaler
 Logisk nolla : < 0.8 V
 Logisk etta : 2.4 V till 5.5 V

Ström- ut
 till signaljord : Max 15 mA
 till + 5 V : Max 24 mA
 Extern belastning : Min 200 ohm, resistiv last
 Ström- in : 20 uA vid "1", 200 uA vid "0"
 Drivbuffer : Bidirektionell buffer typ 74LS243

Överföringshastighet
 Max i BASIC : 1 ms /Port
 Max i maskinspråk : 10 us /Port

=== KOMMANDON I BASIC ===

Datakommandon

OUT 1,C : Kortadress C
 OUT 0,D : Data ut port A
 OUT 32,D : Data ut port B
 OUT 64,D : Data ut port C
 OUT 96,D : Data ut port D
 D=INP(0) : Data in port A
 D=INP(32) : Data in port B
 D=INP(64) : Data in port C
 D=INP(96) : Data in port D

Kontrollkommandon

OUT 128,K : Kontroll ut port A
 OUT 160,K : Kontroll ut port B
 OUT 192,K : Kontroll ut port C
 OUT 224,K : Kontroll ut port D

K=15 : Utmode

K=79 : Inmode

K=207 : In/ut-mode, följt av riktningsmönster
 K=15 : 4 höga bitar ut, 4 låga bitar in

K=207 : In/ut-mode, följt av riktningsmönster
 K=240 : 4 höga bitar in, 4 låga bitar ut

***** BESKRIVNING *****

ABC PIO anpassar TTL-kompatibel utrustning till ABC 80. Kortet är uppbyggd kring två Zilog Z80 PIO-kretsar med totalt 32 bitar. Dessa är grupperade på 4 portar om 8 bitar var, kallade för port A, B, C och D. Två dubbelriktade buffrar finns mellan varje port på PIO:n och anpassningskontakten. Drivbuffrarna kan ställas i valfri riktning ut eller in. Riktningen bestäms av bygling på anpassningskontakten. Detta innebär att 8 bitar på en port kan vara inåtriktade, utåtriktade eller så kan 4 bitar vara inåtriktade och 4 bitar utåtriktade.

I följande text beskrivs en buffer med portbokstav och Låg för de fyra lägre bitarna samt Hög för de fyra högre bitarna (AHög=port A, bit 4 till 7).

ABC PIO kan användas för många olika typer av digital anpassning. Anpassningen utförs i två steg:

1. Anslutning av TTL-signaler från digital utrustning till ABC PIO.
2. Programmering av en drivrutin för denna anslutning.

Anslutningen utförs så att man lätt kan ta emot eller sända data, utan allt för omfattande konverteringar i drivprogrammet. Om man har 8 binära dataledningar från t.ex. en A-D-omvandlare eller till en parallell skrivare, kopplas dessa till samma port t.ex. port A, på anpassningskontakten. Kontrollsignalerna i ovanstående exempel kopplas till nästa port.

Om man har ett instrument med sex parallella BCD-siffror och diverse in- och utgående kontrollsignaler, kopplas dessa lämpligen på följande sätt:

Port AHög: Kontrollsignaler in
 Port Alåg: Första siffra (MSD)
 Port BHög-CLåg: Siffra 2 till 5
 Port Dhög: Siffra 6 (LSD)
 Port DLåg: Kontrollsignaler ut

Alla portar utom DLåg bygglas för in-riktning på anpassningskontakten. De inåtriktade data och kontrollerledningar som inte används, bör kopplas till jord.

Programmeringen av ABC PIO kan antingen göras helt i BASIC eller i maskinspråk med hjälp av assembler. Man kan också använda färdiga drivprogram (se sid 10). Kommandona i BASIC beskrivs på sidan 6 och 7.

***** STIFTSPLACERING *****

Nedanstående tabell visar hur signalerna är placerade på stiften till anpassningskontakten. Två lika grupper finns till respektive PIO-krets om 16 bitar. Varje grupp innehåller 16 dataledningar, signaljord, riktningsval av data, datapuls samt +5V.

Stiften i rad B och C med samma nummer är sammankopplade. Anpassningskontakten på ABC PIO är ett Europadon 96/64-stifts kretskortsdon, vilket innebär att A- och C-raderna är försedda med stift.

SIGNAL	STIFT	STIFT	SIGNAL
Data A1	A01	BC01	Data A0
Data A3	A02	BC02	Data A2
Data A5	A03	BC03	Data A4
Data A7	A04	BC04	Data A6
Data B1	A05	BC05	Data B0
Data B3	A06	BC06	Data B2
Data B5	A07	BC07	Data B4
Data B7	A08	BC08	Data B6
Signaljord	A09	BC09	Signaljord
Signaljord	A10	BC10	Signaljord
Ut/in-ALåg	A11	BC11	Ut/in-AHög
Ut/in-BLåg	A12	BC12	Ut/in-BHög
A-puls	A13	BC13	B-puls
Ledig	A14	BC14	Ledig
+5V	A15	BC15	+5V
Ledig	A16	BC16	Ledig
Data C1	A17	BC17	Data C0
Data C3	A18	BC18	Data C2
Data C5	A19	BC19	Data C4
Data C7	A20	BC20	Data C6
Data D1	A21	BC21	Data D0
Data D3	A22	BC22	Data D2
Data D5	A23	BC23	Data D4
Data D7	A24	BC24	Data D6
Signaljord	A25	BC25	Signaljord
Signaljord	A26	BC26	Signaljord
Ut/in-CLåg	A27	BC27	Ut/in-CHög
Ut/in-DLåg	A28	BC28	Ut/in-DHög
C-puls	A29	BC29	D-puls
Ledig	A30	BC30	Ledig
+5V	A31	BC31	+5V
Ledig	A32	BC32	Ledig

Data A0 - D7 är respektive signal in eller ut för porten.

Ut/in-ALåg - DHög är riktningsväljare för respektive buffer. Normalt är buffern utåtriktad när denna ledning är fri och blir inåtriktad när ledningen kopplas till signaljord. T.ex. visar ALåg att det gäller buffern för de fyra lägre bitarna på A-porten.

A-puls - D-puls ger en puls ut, som är 0.3 usek. lång, för varje ny data som skrivs eller läses från porten. Detta gäller vid datakommandon när porten är ställd i inmode eller utmode (se sida 6).

+5V är spänning ut.

***** KORTVAL,BYGLINGAR *****

== KORTVAL==

ABC PIO rekommenderas att i första hand ha kortadress 2,3,4 och 5. Kortval görs på ABC PIO med hjälp av 6 byglingar i position 8 mitt på kortet (se placeringsschema sida 12). Byglingarna är märkta med siffrorna 0 till 5 för motsvarande bit. Där bygling saknas, erhålles en binär vikt enligt tabellen nedan. Summan av vikterna utgör kortadressen. Är kortadressen 5 skall bygling saknas i bygling 0 och 2, medan övriga skall ha en bygling. Nedanstående tabell ger förhållandet mellan bit och dess vikt:

BITNR	0	1	2	3	4	5
VIKT	1	2	4	8	16	32

Kortvalet testas med kommandot OUT 1,A ,där A är kortadressen. Om kortadressen överensstämmer tänds lysdioden, som visar att detta kort är aktivt.

==PULSSIGNALER==

De fyra byglingarna mellan position 10 och 11 (se sida 12) märkta A, B, C och D används av pulsutgångarna A-puls till D-puls. Utsignalen är normalt hög med en låg puls. Om man byglar för respektive puls blir signalen normalt låg med en hög puls. Puls finns endast på port som är ställd i inmode eller utmode (se sida 6).

==TERMINERINGSNÄT==

Vid busskontakten till ABC-80 finns utrymme för termineringsnät, bestående av 8 st kondensatorer (680 pF) och en SIL-krets med 8 motstånd (220 ohm). Dessa komponenter skall monteras dit när ABC PIO används i enkortslådan eller när termineringsnät saknas av annan orsak. Termineringsnät finns bl.a. i EX8, FD2, FD2U, och trekortschassit. Nätet har till uppgift att anpassa signalerna på bussen, så att reflektioner och andra störningar minskas.

Från ABC 80 initierar man kortet, styr och läser av det med speciella kommandon. Man kan dela upp kommandona i två grupper;

1 : Kontrollkommandon för styrning av kortet
2 : Datakommandon för dataöverföring via kortet
Här nedan följer beskrivning av kontrollkommandona och på nästa sida finns beskrivning av datakommandona.

==== KONTROLLKOMMANDON ====

Man börjar alltid med att ge kontrollkommandon för att aktivera kortet, samt att initiera PIO-kretsarna för samma riktning som buffrarna är byglade till.

==KORTVAL==

OUT 1,A Kortval för att aktivera kort med adress A, där A är 0 - 63.

==PIO-KONTROLL==

OUT 128,C Kontroll av PORT A.
OUT 160,C Kontroll av PORT B.
OUT 192,C Kontroll av PORT C.
OUT 224,C Kontroll av PORT D.

C markerar i vilken mode som porten skall arbeta. I nedanstående text är "KPORT" ett av ovanstående portnummer, alltså 128, 160, 192 eller 224.

=UTMODE=

OUT KPORT,15 Ställ port i ut-riktning.
Ingen bygling för riktning på kortkontakten.

=INMODE=

OUT KPORT,79 Ställ port i in-riktning.
Bygling av ut/in-KLåg och -KHög till GND.

=IN/UTMODE=

OUT KPORT,207
OUT KPORT,15 Port ställs med 4 höga bitar ut och 4 låga bitar in.
Bygling av ut/in-KLåg till GND.

OUT KPORT,207
OUT KPORT,240 Port ställs med 4 höga bitar in och 4 låga bitar ut.
Bygling av ut/in-KHög till GND.

OUT KPORT,207
OUT KPORT,255 Motsvarar INMODE.

OUT KPORT,207
OUT KPORT,0 Motsvarar UTMODE.

=EXEMPEL=

Ex.1 Ställ port C i utmode: Bygla inte stift A28 och B28 till GND.
Kommando blir: OUT 192,15

Ex.2 Ställ port A i inmode: Bygla stift A11 - A10 och B11 - B10.
Kommando blir: OUT 128,79

Ex.3 Ställ port DLåg ut och DHög in: Bygla stift B29 - B26.
Kommando blir: OUT 224,207
OUT 224,240

Ex.4 Utför Ex.3 i en sats.
Kommando blir: OUT224,207,224,240

==== DATAKOMMANDON ====

Med datakommandona kommunicerar man med den anslutna utrustningen. För att dessa kommandon skall fungera, måste man dessförinnan ha kopplat in de digitala signalerna till anslutningskontakten och initierat ABC PIO med kontrollkommandon enligt föregående sida. Vid styrning eller läsning av en enskild bit, kan man skala respektive bit med de logiska operationerna och bitens vikt (se ABC om mätatorsystem 4.1).

Här visas sambandet mellan bitnummer och dess vikt:

BITNR	0	1	2	3	4	5	6	7
VIKT	1	2	4	8	16	32	64	128

I nedanstående kommandon är D det data som sänds ut till porten eller tas emot från porten. Om variabler och konstanter följs av %, blir programmet både kortare och snabbare.

==DATAKOMMANDON==

=UTDATA=

OUT 0,D Data ut till PORT A.
 OUT 32,D Data ut till PORT B.
 OUT 64,D Data ut till PORT C.
 OUT 96,D Data ut till PORT D.

=INDATA=

D=INP(0) Data in från PORT A.
 D=INP(32) Data in från PORT B.
 D=INP(64) Data in från PORT C.
 D=INP(96) Data in från PORT D.

=UT/INDATA=

Om man läser av en utåtriktad port med INP, får man samma data som vid föregående laddning med OUT till den porten. Vid blandad ut/in-hantering av en port används ovanstående kommandon. De bitar som laddas ut till en inåtriktad port har ingen funktion, medan de bitar som hämtas in från en utåtriktad port, är samma som vid föregående laddning dit.

=EXEMPEL=

Nedanstående exempel förutsätter att Ex.1 till Ex.3 är utförda.

Ex.5 Ettställ enbart bit 1 och 7 till port C

Kommando blir: OUT 64%,130%

Ex.6 Ta in data från port A till variabeln D%

Kommando blir: D%=INP(0%)

Ex.7 Läs av ett BCD-värde på port D Högt till variabeln B%

B%=(INP(96%) AND 240%)/16%

Ex.8 Ettställ bit 4 på port C utan att ändra övriga bitar

Kommando blir: OUT 64%,(INP(64%) OR 16%)

Ex.9 Byt nivå på bit 0 på port C utan att ändra övriga bitar

Kommando blir: OUT 64%,(INP(64%) XOR 1%)

Ex.10 Läs av två BCD-siffror från port A till variabeln B%

Kommando blir: B%=(INP(0%)AND 240%)/16%*10+(INP(0%)AND 15%)

Nedanstående exempel visar hur man kopplar in en digital våg med 6 parallella BCD-siffror till ABC-PIO och hur tillhörande drivprogram ser ut.

===== KOPPLINGSBESKRIVNING =====					
V Å G #	A	B	C	-	P I O
Anm.	Stift#	Stift	B i t	Anmärkning	Adress
READY	47 #	A4	A7	READY	Port A Data : 0 Kontr:128
ERROR	45 #	C4	A6	ERROR	
SIGN	43 #	A3	A5	SIGN	
	#	C3, A10	A4	FlagA	
8x10E5	30 #	A2	A3	8 Digit 1	
4x10E5	29 #	C2	A2	4 (MSD)	
2x10E5	28 #	A1	A1	2	Port B Data : 32 Kontr:160
1x10E5	27 #	C1	A0	1	
	#				
8x10E4	25 #	A8	B7	8 Digit 2	
4x10E4	24 #	C8	B6	4	
2x10E4	23 #	A7	B5	2	
1x10E4	22 #	C7	B4	1	Port C Data : 64 Kontr:192
8x10E3	20 #	A6	B3	8 Digit 3	
4x10E3	19 #	C6	B2	4	
2x10E3	18 #	A5	B1	2	
1x10E3	17 #	C5	B0	1	
	#				
8x10E2	15 #	A20	C7	8 Digit 4	Port D Data : 96 Kontr:224
4x10E2	14 #	C20	C6	4	
2x10E2	13 #	A19	C5	2	
1x10E2	12 #	C19	C4	1	
8x10E1	10 #	A18	C3	8 Digit 5	
4x10E1	9 #	C18	C2	4	
2x10E1	8 #	A17	C1	2	Port D Data : 96 Kontr:224
1x10E1	7 #	C17	C0	1	
	#				
8x10E0	5 #	A24	D7	8 Digit 6	
4x10E0	4 #	C24	D6	4 (LSD)	
2x10E0	3 #	A23	D5	2	
1x10E0	2 #	C23	D4	1	Port D Data : 96 Kontr:224
REQ. IN	48 #	A22	D3	UT REQ.	
	#	C22	D2	UT 4	
	#	A21	D1	UT 2	
	#	C21	D0	UT 1	
	#				
GND	50 #	A10, C10	GND		
	#	C11, C10	Ut/in-AH	IN Port AHög	
	#	A11, A10	Ut/in-AL	IN Port ALåg	
	#	C12, C10	Ut/in-BH	IN Port BHög	
	#	A12, A10	Ut/in-BL	IN Port BLåg	
	#	C27, C26	Ut/in-CH	IN Port CHög	
	#	A27, A26	Ut/in-CL	IN Port CLåg	
	#	C28, C26	Ut/in-DH	IN Port DHög	
	#	A28	Ut/in-DL	UT Port DLåg, ingen bygling	
	#				

Inkopplingen består av 6 BCD-siffror med tillhörande kontrollsignaler. Handskakningen sker genom att UTREQ. sätts hög varpå klarsignalen READY skall gå hög. Om vågen är överlastad är ERROR låg och därmed data felaktig. FLAG A är en extern flaggmarkör som inte används i detta program. UT1, UT2 och UT4 är tre binära utgångar förutom UTREQ. Dessa kan användas till extern tarering av våg, styrning av provväxlare, kanalval och dylikt. I detta program används de inte. De ingångar som inte används, skall kopplas till signaljord GND.

===== DRIVPROGRAM DIGITAL VÅG =====

Här nedan följer ett program för avläsning av vågen i ovanstående inkoppling. Först initieras ABC PIO och sedan utförs handskakning med UTREQ och READY. Därefter avläses alla BCD-siffror till en ASCII-sträng, avskalas och lämnas sedan till huvudrutinen.

```

10 REM ***** LÄSNING AV VÅG *****
20 REM DATUM SIGN
30 GOSUB 1000 : REM Initiering
40 GOSUB 1130 : REM Ta in ett mätvärde
50 ONERRORGOTO 90
60 V=VAL(V$) : REM Omvandla sträng till värde
70 PRINT V
80 GOTO 40
90 PRINT "ÖVERLAST"
110 GOTO 40
1000 REM ##### VÅGLÄS-SUBROUTIN #####
1010 REM ===== VÅGINITIERING =====
1020 REM UTFÖRS EN GÅNG FÖRE LÄSVÅG
1030 REM INGEN IN-ELLER UT-VARIABEL
1040 REM INGEN INTERN-VARIABEL ANVÄNDS
1050 REM =====
1060 OUT 1%,2% : REM Kortval
1070 OUT 128%,79% : REM Inmode PORT A
1080 OUT 160%,79% : REM Inmode PORT B
1090 OUT 192%,79% : REM Inmode PORT C
1100 OUT 224%,207%,224%,240% : REM Inmode DLåg, Utmode DHög
1110 RETURN
1120 REM ===== LÄSVÅG =====
1130 REM AVLÄS BCD-VÅG PÅ KORT 2
1140 REM INGEN INVARIABEL
1150 REM UTVARIABEL V$=VÅGVÄRDE
1160 REM OM V$="E"+värde ÄR VÅGEN ÖVERLASTAD
1170 REM INTERNVARIABLER V%,V1%,V2%
1180 REM =====
1190 OUT 1%,2% : REM Kortval
1200 OUT 96%,8% : REM Request UT på bit D3
1210 V$=""
1220 V1%=INP(0%) : REM Läs status
1230 IF (V1% AND 64%)=0% THEN 1250 : REM Errormärke
1240 IF (V1% AND 128%)=0% THEN 1220 : REM Vänta Ready
1250 FOR V2%=0% TO 96% STEP 32% : REM Läsloop fyra portar
1260 V%=INP(V2%) : REM läs en port
1270 V$=V$+CHR$((V% AND 240%)/16%+48%) : REM Bilda ASCII av BCD-Hög
1280 V$=V$+CHR$((V% AND 15%)+48%) : REM Bilda ASCII av BCD-Låg
1290 NEXT V2%
1300 OUT 96%,0% : REM Ingen mer REQUEST UT
1310 V%=MID$(V$,2%,6%) : REM Ta ut data
1320 IF (V1% AND 32%)=0% THEN V$="+"+V$ ELSE V$="-"+V$ : REM tecken
1330 IF (V1% AND 64%)=0% THEN V$="E"+V$ ELSE V$=" "+V$ : REM errormärke
1340 RETURN

```

Den utgående strängen ser ut på följande sätt:

Pos: 1 : 2 : 3 : 4 : 5 : 6 : 7 : 8 :
 Data:Err :+/- :BCD1 :BCD2 :BCD3 :BCD4 :BCD5 :BCD6 :

+/- = Tecken på data

Err = Felmärke "E" om ERROR är aktiv låg eller " " om ERROR är hög.

BCD1 - BCD6 = Decimala siffror

===== FÄRDIGA DRIVPROGRAM =====

Istället för ovanstående drivrutin kan man använda färdiga drivrutiner. Dessa är gjorda med olika inriktningar, men en och samma drivrutin kan användas på olika sätt genom det optionsval som anges vid öppningen av rutinen.

Man läser in drivrutinen före sitt huvudprogram, varvid den reserverar ett eget utrymme i RAM.

Följande rutin, som ingår i BCD-drivrutinprogrammet och kallas för BC6: , läser av på samma sätt som föregående drivrutin:

```
10 REM ***** VÅGLÄS BC6: *****
20 REM 800903 HMSCM
30 OPEN "BC6:2AN" ASFILE 1 : REM Öppna drivrutin med kod 2AN
40 ONERRORGOTO 80 : REM hoppa till 80 om överlast
50 INPUT #1,V : REM Ta in ett mätvärde
60 PRINT V : REM Skriv vikt
70 GOTO 50
80 PRINT "ÖVERLAST"
90 GOTO 40
```

Begreppet "2AN" i rad 30 är uppgifter till drivrutinen som visar hur den skall arbeta.

2 anger kortadress på ABC PIO.

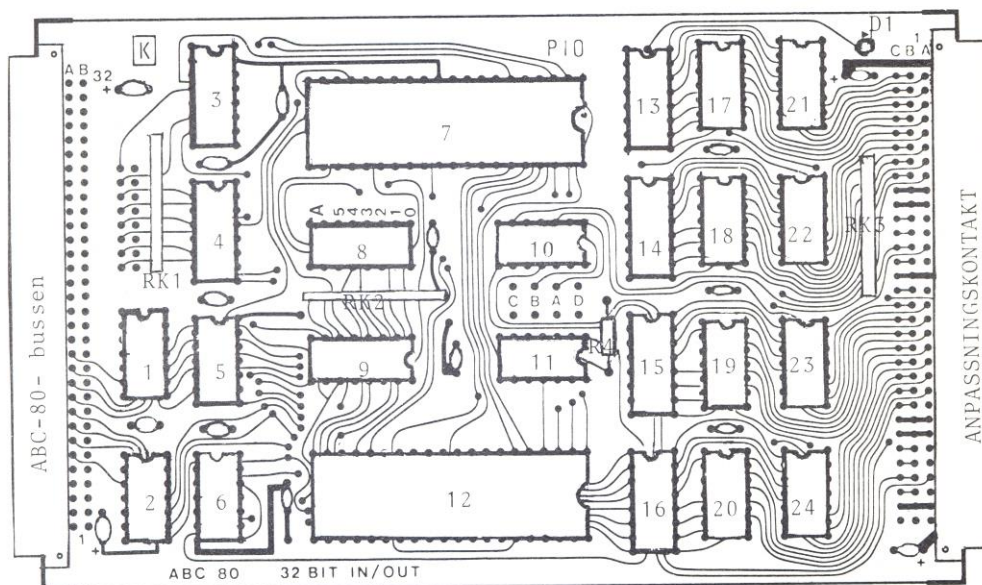
A anger option A av fyra möjliga optioner till BC6:

N är nivåkod för handskakningssignalerna, tecken och felmarkering.

Dessa optioner kan utläsas ur dokumentationen till BCD-drivprogrammet.

1. Gör en kopplingsbeskrivning som även kan fungera som underlag vid tillverkning av anslutningskabel (se sid 8) och gör iordning anslutningskabel.
2. Stäng av strömmen till alla ingående enheter, även utrustningen som skall kopplas in.
3. Koppla in anslutningskabel mellan ABC PIO och utrustningen.
4. Sätt ABC PIO på en kortplats i FD2, FD2U, EX8 eller EX1.
5. Sätt på strömmen till alla ingående enheter.
6. Testa om kortvalet är riktigt med följande kommando:
OUT 1,C där C är kortadressen.
Lysdioden skall då tändas.
7. Testa utgående TTL-signal på t.ex. port D bit 0 med följande kommando:
Ingen bygling skall finnas mellan A29 och signaljord GND.
OUT 224,15 för att initiera port D utåt.
OUT 96,0 för att sätta stift C21 låg, vilket är bit 0 på port D.
OUT 96,1 för att sätta stift C21 hög.
8. Testa ingående TTL-signal på t.ex. port A bit 0 med följande kommando:
Bygling skall finnas mellan A11 och signaljord GND.
OUT 128,79 för att initiera port A.
PRINT INP(0) AND 1 att skriva ut nivån på stift C1.
9. Följande program sätter ut ett allt högre värde på alla portar. Den lägsta biten på varje port byter oftast nivå, medan allt högre bitar halverar nivåbytet tidsmässigt.
10 REM ***** POUTBIT *****HMSCM*****
20 OUT 1%,2% : REM Kortval
30 FOR A%=128% TO 224% STEP 32% : REM Initiera portar ut
40 OUT A%,15% : REM Port utmode
50 NEXT A%
60 FOR A%=0% TO 255% : REM Sätt data ut
70 OUT 0%,A%,32%,A%,64%,A%,96%,A%
80 NEXT A%
90 GOTO 60
10. Följande program visar bitmönstret på alla portar, när dessa är inåtriktade.
10 REM ***** PIOINBIT *****HMSCM*****
20 ; CHR\$(12);
30 ; " BITTEST PÅ ABC PIO INGÅNGAR"
40 OUT 1%,2% : REM Kortval
50 FOR I%=128% TO 224% STEP 32% : REM Initiera portar
60 OUT I%,79% : REM Sätt portar i inmode
70 NEXT I%
80 ; ; ; "PORT A PORT B PORT C PORT D"
90 ; ; ; "76543210 76543210 76543210 76543210"
100 ; CUR(6,0); : REM Posision av bitmönster
110 FOR P%=0% TO 96% STEP 32% : REM Huvudloop mätning
120 A%=INP(P%) : REM Avläs port
130 J%=128%
140 FOR I%=7% TO 0% STEP -1% : REM Skriv bitmönster
150 IF A% AND J% THEN ; "1"; ELSE ; "0";
160 J%=J%/2%
170 NEXT I%
180 ; " "
190 NEXT P%
200 GOTO 100

*** PLACERINGSSCHEMA ***



Komponenter

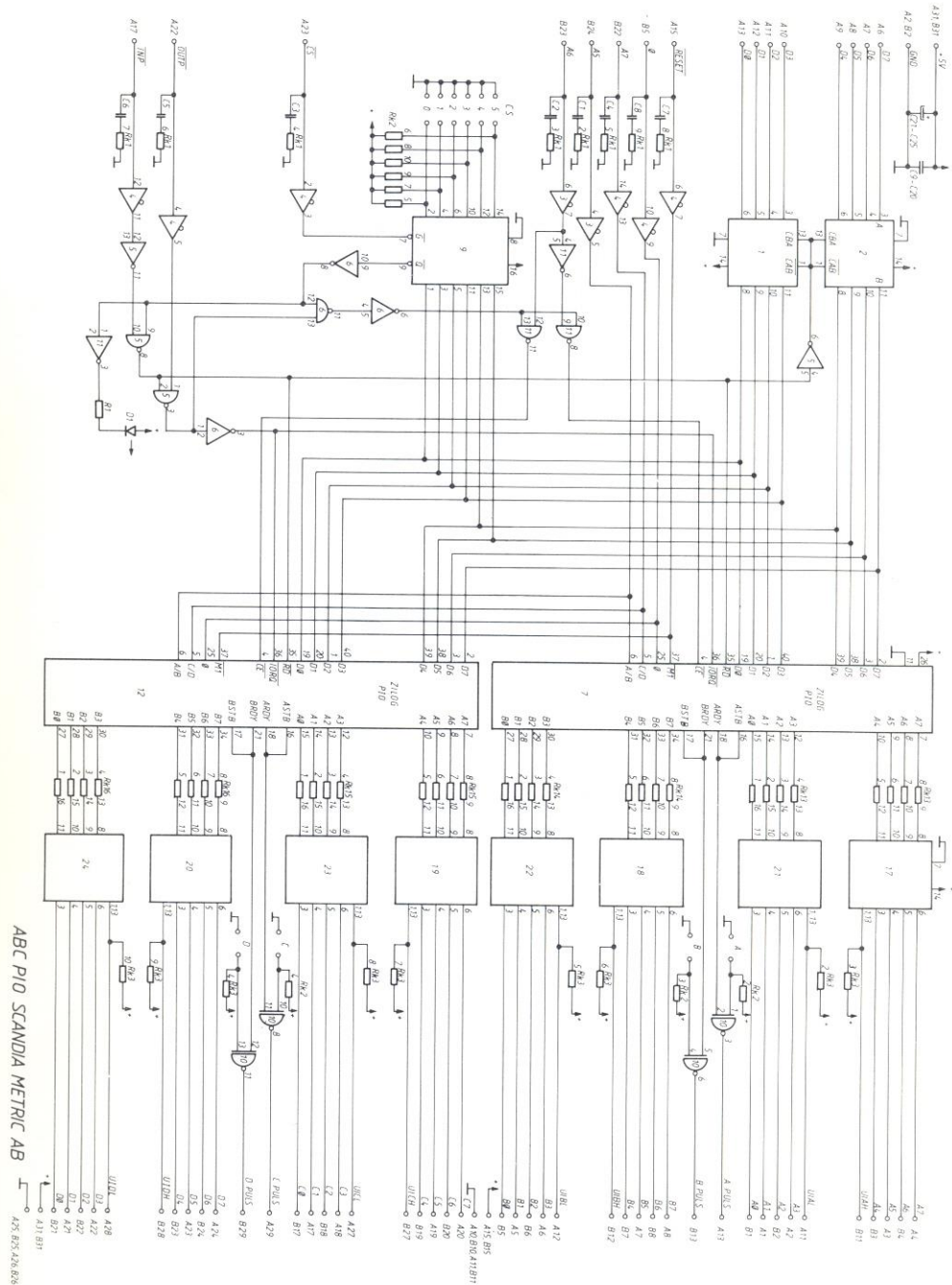
Position Krets

1 - 2	74LS243
3 - 4	74LS367
5 - 6	74LS00
7	Z80 PIO
8	KORTVAL
9	8131
10	74LS86
11	74LS00
12	Z80 PIO
13-16	898-3 8*1 Kohm
17-24	74LS243

(C1-C8)	680 pF
C9-C20	22 nF
C21-C25	6.8 uF

(RK1)	9*220 ohm
RK2-RK3	9*1 Kohm
R4	270 ohm
D1	Lysdiod

ABC 80 - BUSS-KONTAKT



TTL - ANSLUTNINGSKONTAKT