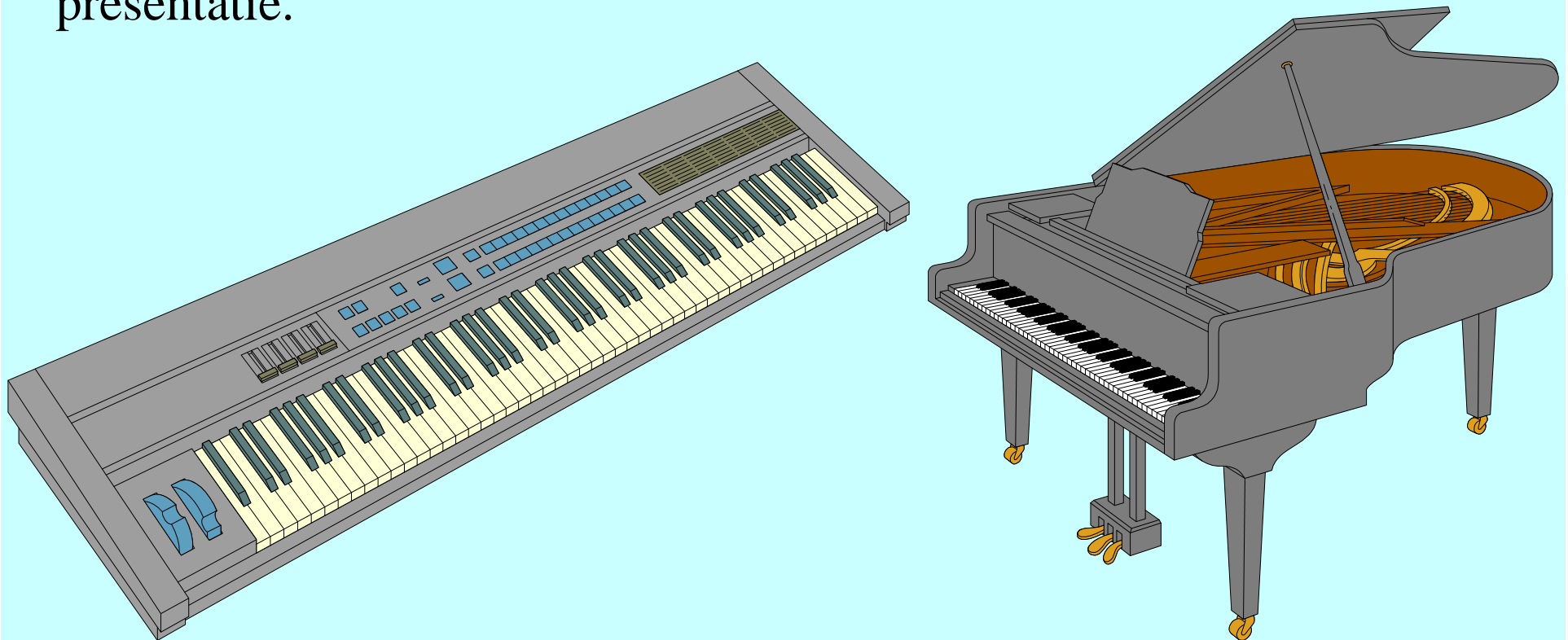


**piano**

Door A.J. Lugtigheid

# voorwoord

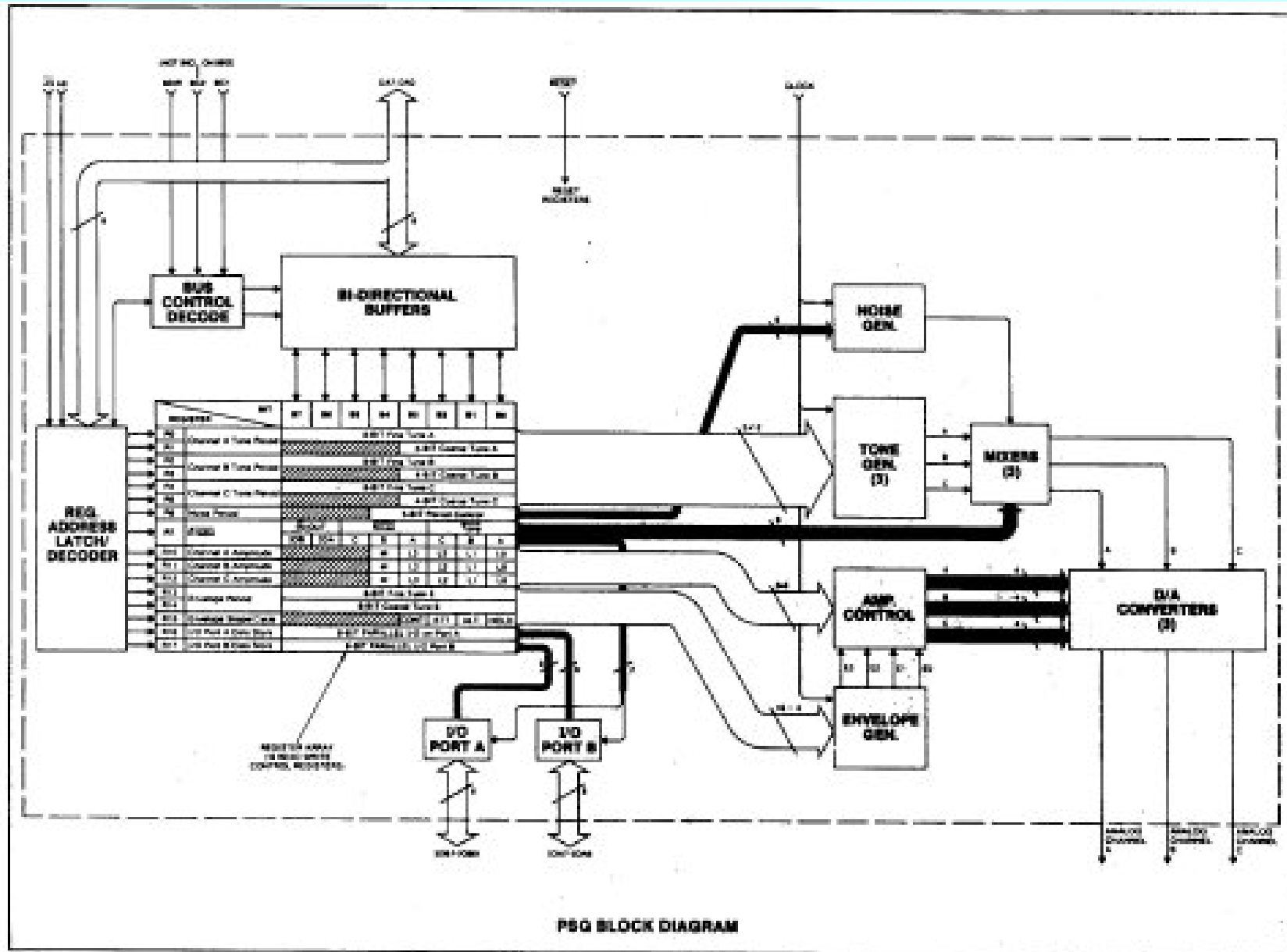
Ik heb lang geleden een fataal verkeersongeluk gehad, en ik had graag op een piano gespeeld. Het ging soms wat moeilijk en daarom dacht ik dat ik het beter kon automatiseren, en daarover gaat deze presentatie.



# AY3-8910A

IK merkte dat het pianospelen gaat met tonen, die je een bepaalde tijd aanhoudt, en die tonen bepalen bepalen de melodie. Ik dacht: “IK ga een monofone synthesizer maken”, want mijn ademhaling was wat ontregeld, en als je er dan bij zingt dan gaat het beter. Ik keek dus eerst in een folder van TANDY en daar zag ik het genoemde IC staan. Het kan 3 tonen tegelijk voortbrengen, maar ik gebruik er maar 1, gewoon wat programmeerbare delers die een klokpuls omlaag brengen en dat stuurde ik aan met de printerpoort van een van mijn computers. Aan de uitgang had ik een audio-versterker met een luidspreker

# Programmable sound generator

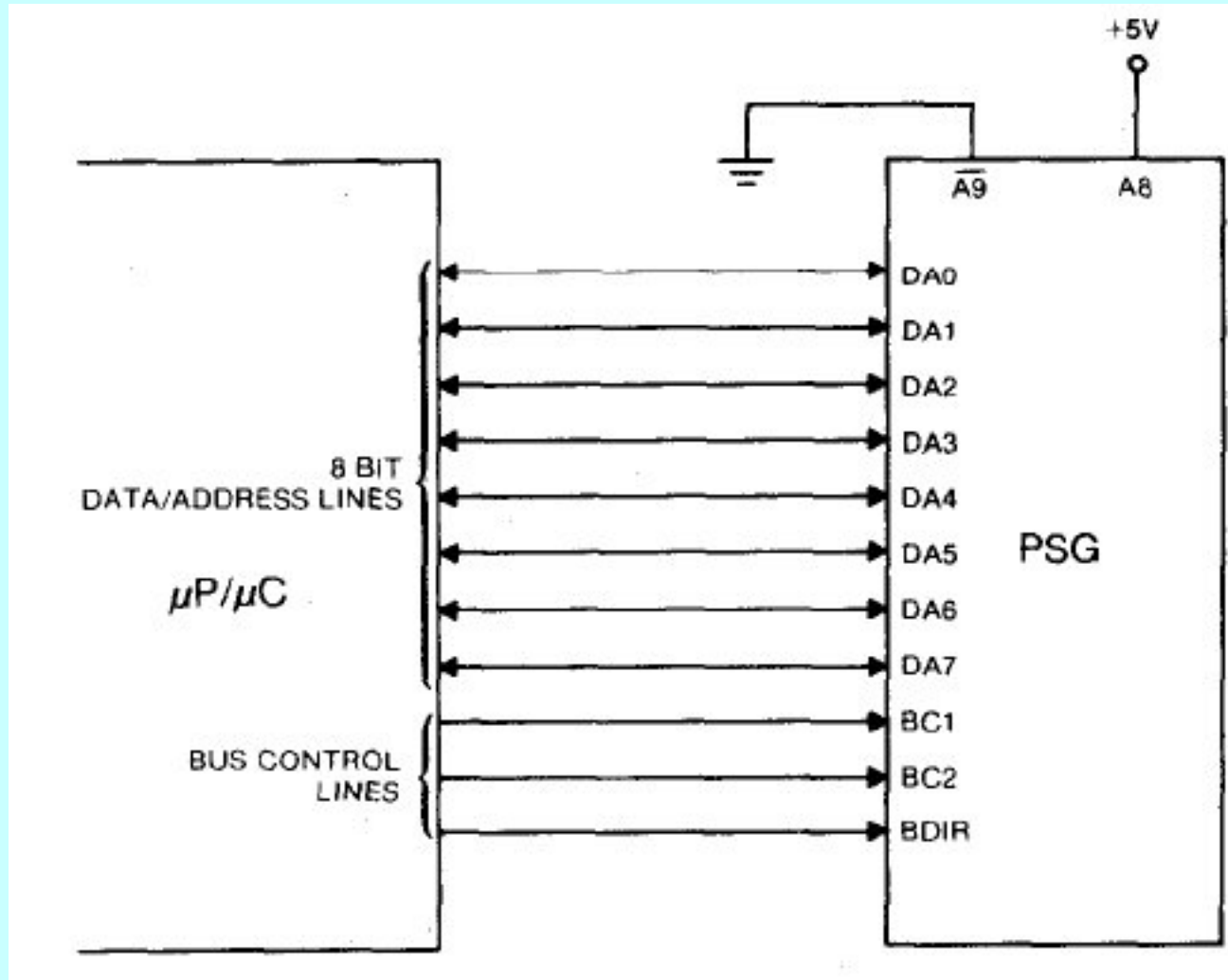


Dit is het inwendige van het IC, zo functioneert het.

# De registers

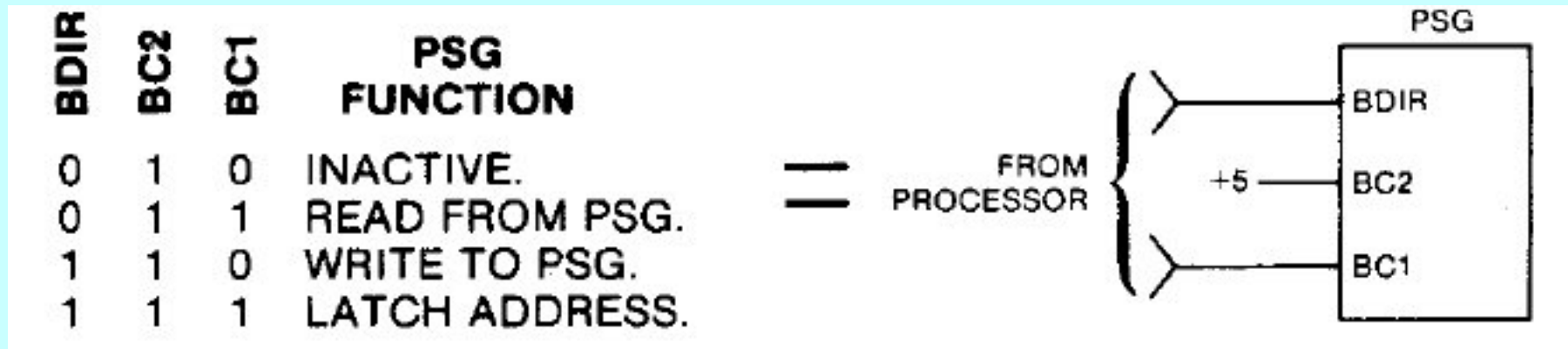
REGISTER		BIT	B7	B6	B5	B4	B3	B2	B1	B0	
R0	Channel A Tone Period	8-BIT Fine Tune A									
R1							4-BIT Coarse Tune A				
R2	Channel B Tone Period	8-BIT Fine Tune B									
R3							4-BIT Coarse Tune B				
R4	Channel C Tone Period	8-BIT Fine Tune C									
R5							4-BIT Coarse Tune C				
R6	Noise Period						5-BIT Period Control				
R7	Enable	IN/OUT			Noise			Tone			
		IOB	IOA	C	B	A	C	B	A		
R10	Channel A Amplitude						M	L3	L2	L1	L0
R11	Channel B Amplitude						M	L3	L2	L1	L0
R12	Channel C Amplitude						M	L3	L2	L1	L0
R13	Envelope Period	8-BIT Fine Tune E									
R14		8-BIT Coarse Tune E									
R15	Envelope Shape/Cycle						CONT.	ATT.	ALT.	HOLD	
R16	I/O Port A Data Store	8-BIT PARALLEL I/O on Port A									
R17	I/O Port B Data Store	8-BIT PARALLEL I/O Port B									

# INTERFACING 1



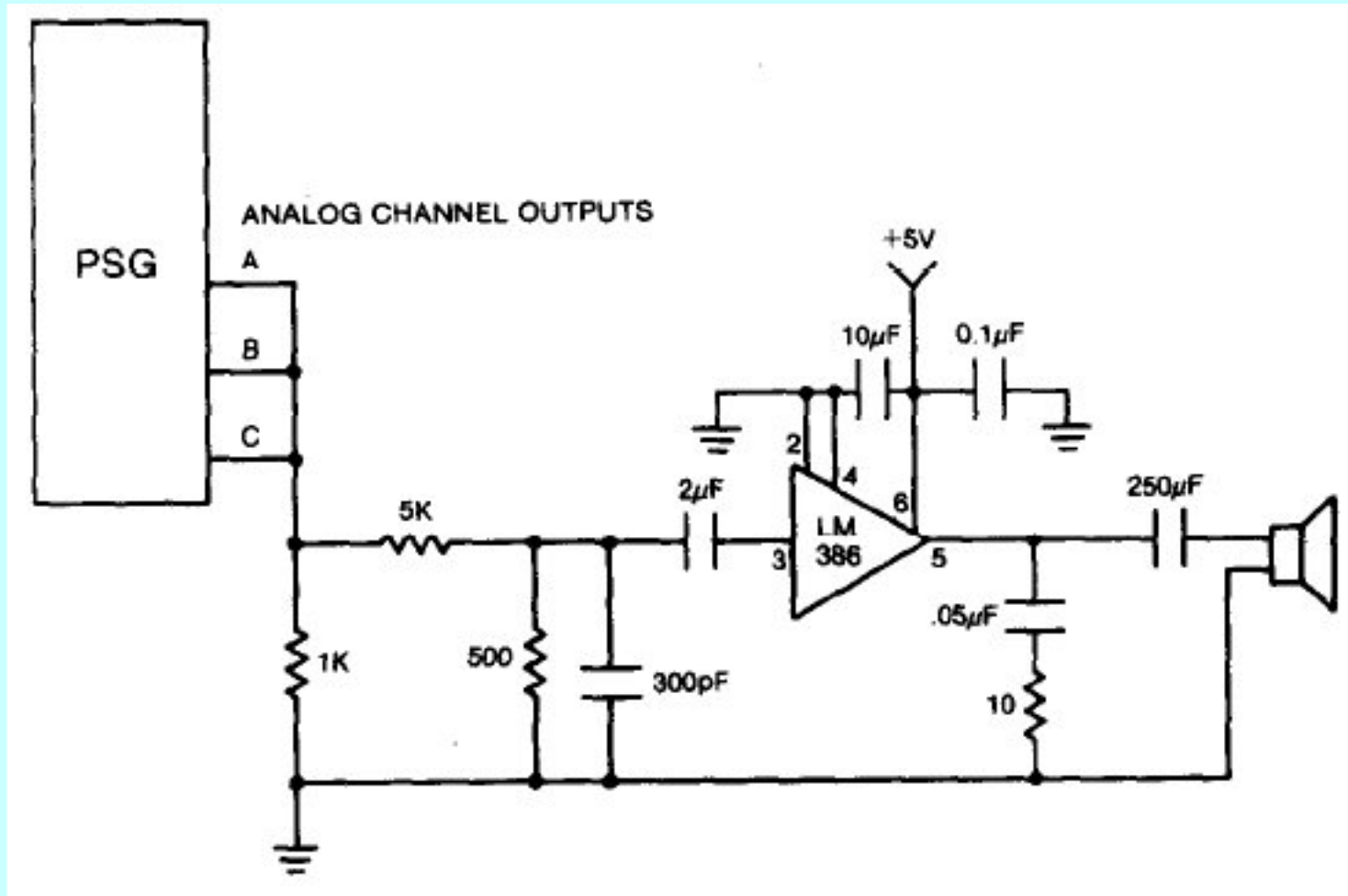
Zo gaat dat, gewoon wat draden naar de chip

# INTERFACING 2



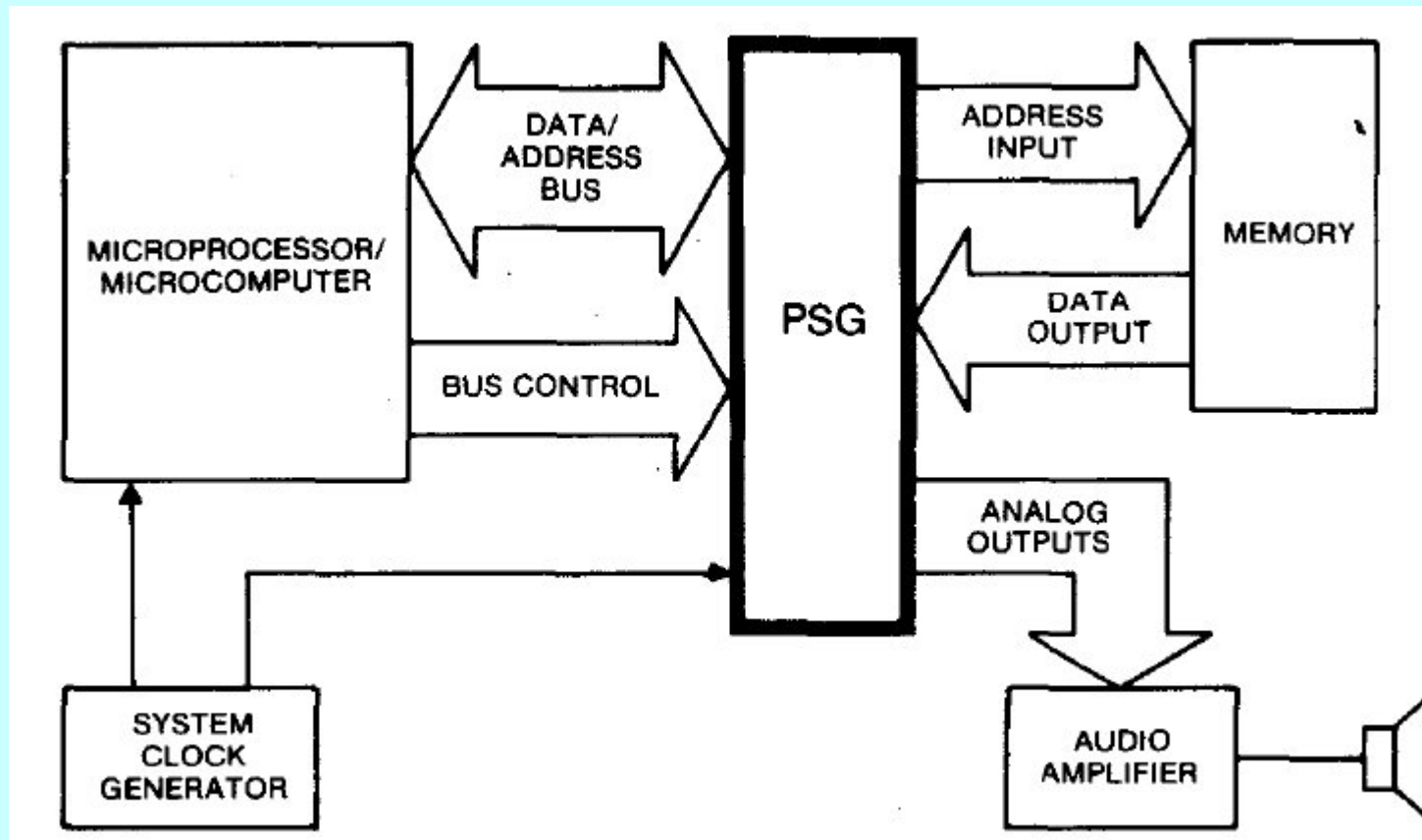
Zo wordt de PSG bestuurd, eerst wordt er **inactive** gegeven, dan wordt er data geplaatst op de databus (het register adres) en dan wordt er **latch** adres gegeven en daarna **inactive**. U heeft nu een register geselecteerd, en U kunt nu de data in dat register veranderen. Om de data te veranderen zet U eerst de data op de databus, dan geeft U **write to PSG** en daarna **inactive**. Om de data uit te lezen geeft U **read from PSG** en daarna **inactive**.

# INTERFACING 3



Dit bedoelen ze, een kleine audio-versterker, die een kleine luidspreker aanstuurd.

# INTERFACING 4



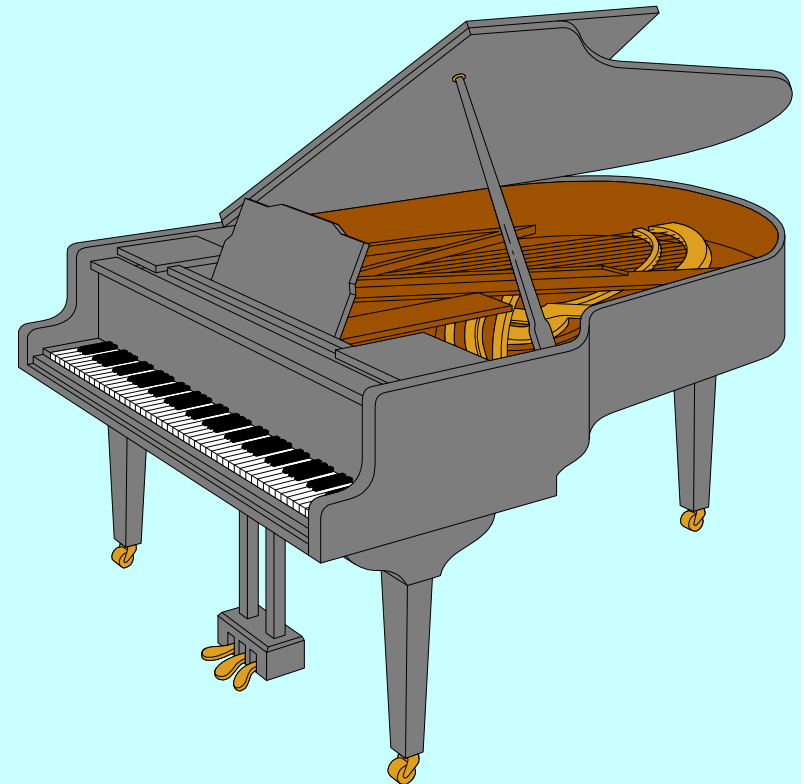
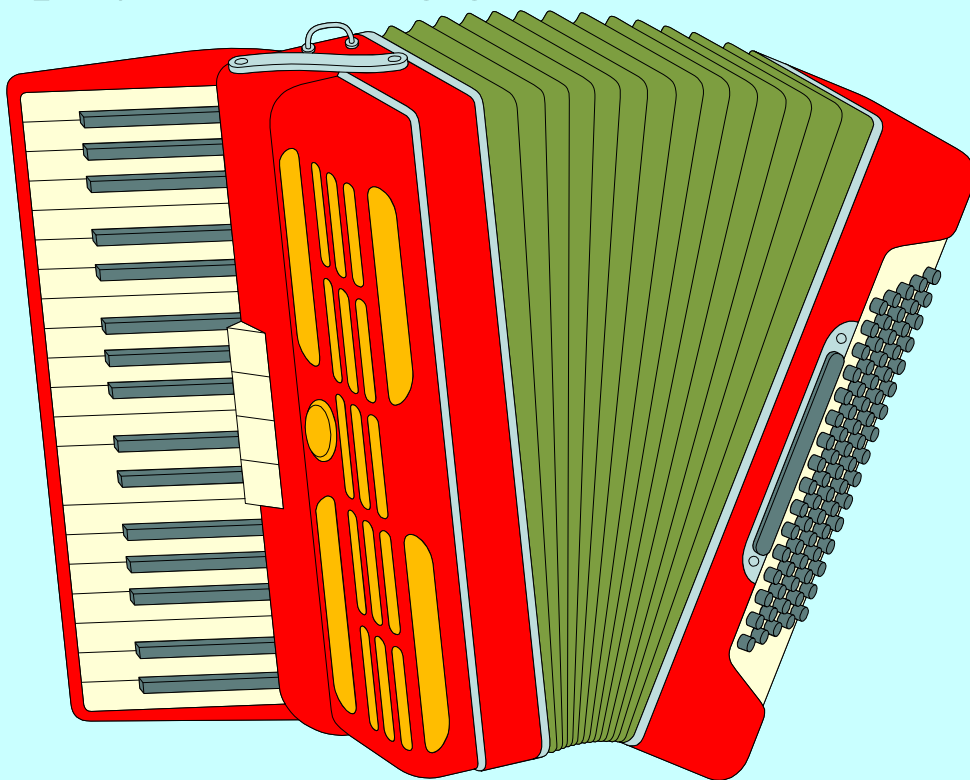
Zo had de fabrikant gedacht dat hun product gebruikt zou worden, maar ik heb soms aan de extra I/O solid state relais gehangen, die wat lampen aanstuurden.

# De software

Toon

Tijdsduur

De software bestaat uit records, en elk record bestaat uit 2 velden. Het eerste veld bestaat uit een waarde die overeenkomt met de gewenste toon, en het tweede veld bepaalt de tijdsduur. Dit was een project van lang geleden, nu zou ik het anders doen.



# Het record

Het record bestaat uit twee velden, de toon en de tijdsduur. Een heel belangrijke eigenschap is het aantal record's, hiervoor gebruik ik het eerste veld van record0.

<b>Aantal</b>	
<b>Toon</b>	<b>Tijdsduur</b>
<b>Toon</b>	<b>Tijdsduur</b>
<b>Toon</b>	<b>Tijdsduur</b>
<b>Toon</b>	<b>Tijdsduur</b>
<b>Toon</b>	<b>Tijdsduur</b>
<b>Toon</b>	<b>Tijdsduur</b>

Ik weet nog goed, in een van mijn prototypen had ik de sterkteregelaar vergeten. Dat is wat lastig als ze vragen “kan het wat zachter?”, oeps dat was ik vergeten.

Het grote voordeel van het op die manier organiseren van de data is dat je het naar een file kan schrijven en bewaren.



# Oktaaf 1 & 2

<b>NOTE</b>	<b>OCTAVE</b>	<b>IDEAL FREQUENCY</b>	<b>ACTUAL FREQUENCY</b>	<b>12-BIT REGISTER VALUE IN OCTAL</b>			
C	1	32.703	32.698	6	5	3	5
C#	1	34.648	34.653	6	2	3	4
D	1	36.708	36.712	5	7	4	7
D#	1	38.891	38.895	5	4	7	4
E	1	41.203	41.201	5	2	3	3
F	1	43.654	43.662	5	0	0	2
F#	1	46.249	46.243	4	5	6	3
G	1	48.999	48.997	4	3	5	3
G#	1	51.913	51.908	4	1	5	3
A	1	55.000	54.995	3	7	6	2
A#	1	58.270	58.261	3	6	0	0
B	1	61.735	61.733	3	4	2	4
C	2	65.406	65.416	3	2	5	6
C#	2	69.296	69.307	3	1	1	6
D	2	73.416	73.399	2	7	6	4
D#	2	77.782	77.789	2	6	3	6
E	2	82.406	82.432	2	5	1	5
F	2	87.308	87.323	2	4	0	1
F#	2	92.498	92.523	2	2	7	1
G	2	97.998	98.037	2	1	6	5
G#	2	103.826	103.863	2	0	6	5
A	2	110.000	109.991	1	7	7	1
A#	2	116.540	116.522	1	7	0	0
B	2	123.470	123.467	1	6	1	2

# OCTAAF 3 & 4

C	3	130.812	130.831	1	5	2	7
C#	3	138.592	138.613	1	4	4	7
D	3	146.832	146.799	1	3	7	2
D#	3	155.564	155.578	1	3	1	7
E	3	164.812	164.743	1	2	4	7
F	3	174.616	174.510	1	2	0	1
F#	3	184.996	184.894	1	1	3	5
G	3	195.996	195.903	1	0	7	3
G#	3	207.652	207.534	1	0	3	3
A	3	220.000	220.198	0	7	7	4
A#	3	233.080	233.043	0	7	4	0
B	3	246.940	246.933	0	7	0	5
C	4	261.624	261.357	0	6	5	4
C#	4	277.184	276.883	0	6	2	4
D	4	293.664	293.598	0	5	7	5
D#	4	311.128	310.724	0	5	5	0
E	4	329.624	329.973	0	5	2	3
F	4	349.232	349.565	0	5	0	0
F#	4	369.992	370.400	0	4	5	6
G	4	391.992	392.494	0	4	3	5
G#	4	415.304	415.839	0	4	1	5
A	4	440.000	440.397	0	3	7	6
A#	4	466.160	466.087	0	3	6	0
B	4	493.880	494.959	0	3	4	2

# OCTAAF 5 & 6

NOTE	OCTAVE	IDEAL FREQUENCY	ACTUAL FREQUENCY	12-BIT REGISTER VALUE IN OCTAL			
C	5	523.248	522.714	0	3	2	6
C#	5	554.368	553.766	0	3	1	2
D	5	587.328	588.741	0	2	7	6
D#	5	622.256	621.449	0	2	6	4
E	5	659.248	658.005	0	2	5	2
F	5	698.464	699.130	0	2	4	0
F#	5	739.984	740.800	0	2	2	7
G	5	783.984	782.243	0	2	1	7
G#	5	830.608	828.598	0	2	0	7
A	5	880.000	880.794	0	1	7	7
A#	5	932.320	932.173	0	1	7	0
B	5	987.760	989.918	0	1	6	1
C	6	1046.496	1045.428	0	1	5	3
C#	6	1108.736	1107.532	0	1	4	5
D	6	1174.656	1177.482	0	1	3	7
D#	6	1244.512	1242.898	0	1	3	2
E	6	1318.496	1316.009	0	1	2	5
F	6	1396.928	1398.260	0	1	2	0
F#	6	1479.968	1471.852	0	1	1	4
G	6	1567.968	1575.504	0	1	0	7
G#	6	1661.216	1669.564	0	1	0	3
A	7	1760.000	1747.825	0	1	0	0
A#	7	1864.640	1864.346	0	0	7	4
B	6	1975.520	1962.470	0	0	7	1

# OCTAAF 7 & 8

C	7	2092.992	2110.581	0	0	6	5
C#	7	2217.472	2237.216	0	0	6	2
D	7	2349.312	2330.433	0	0	6	0
D#	7	2489.024	2485.795	0	0	5	5
E	7	2636.992	2663.352	0	0	5	2
F	7	2793.856	2796.520	0	0	5	0
F#	7	2959.936	2943.705	0	0	4	6
G	7	3135.936	3107.244	0	0	4	4
G#	7	3322.432	3290.023	0	0	4	2
A	7	3520.000	3495.649	0	0	4	0
A#	7	3729.280	3728.693	0	0	3	6
B	7	3951.040	3995.028	0	0	3	4
C	8	4185.984	4142.992	0	0	3	3
C#	8	4434.944	4474.431	0	0	3	1
D	8	4698.624	4660.866	0	0	3	0
D#	8	4978.048	5084.581	0	0	2	6
E	8	5273.984	5326.704	0	0	2	5
F	8	5587.712	5593.039	0	0	2	4
F#	8	5919.872	5887.410	0	0	2	3
G	8	6271.872	6214.488	0	0	2	2
G#	8	6644.864	6580.046	0	0	2	1
A	8	7040.000	6991.299	0	0	2	0
A#	8	7458.560	7457.385	0	0	1	7
B	8	7902.080	7990.056	0	0	1	6

# sturing

Fig. 36 CLOCK AND BUS SIGNAL TIMING

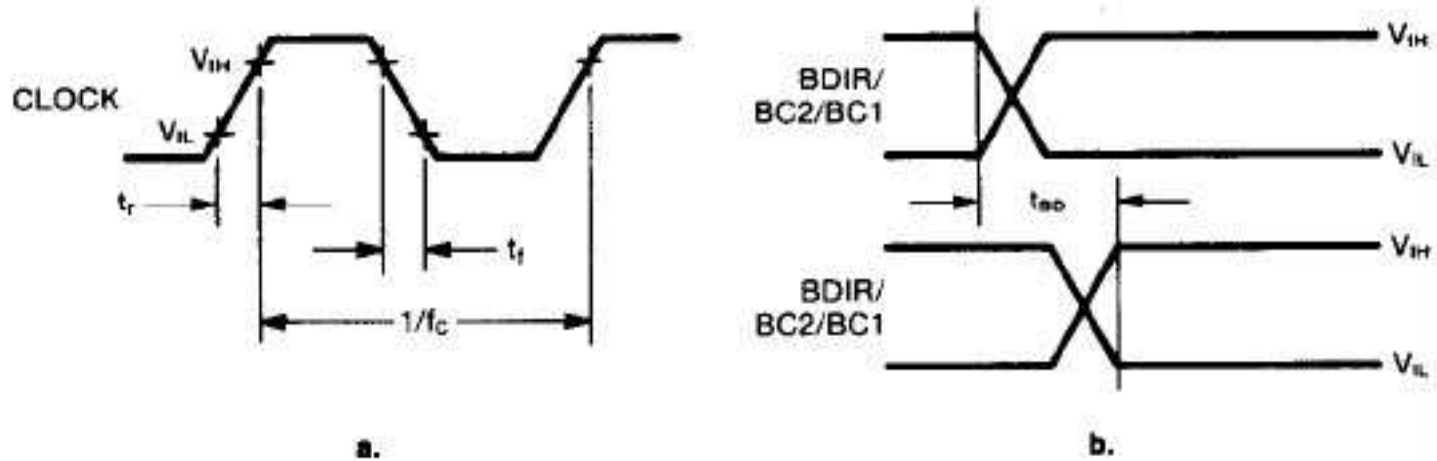
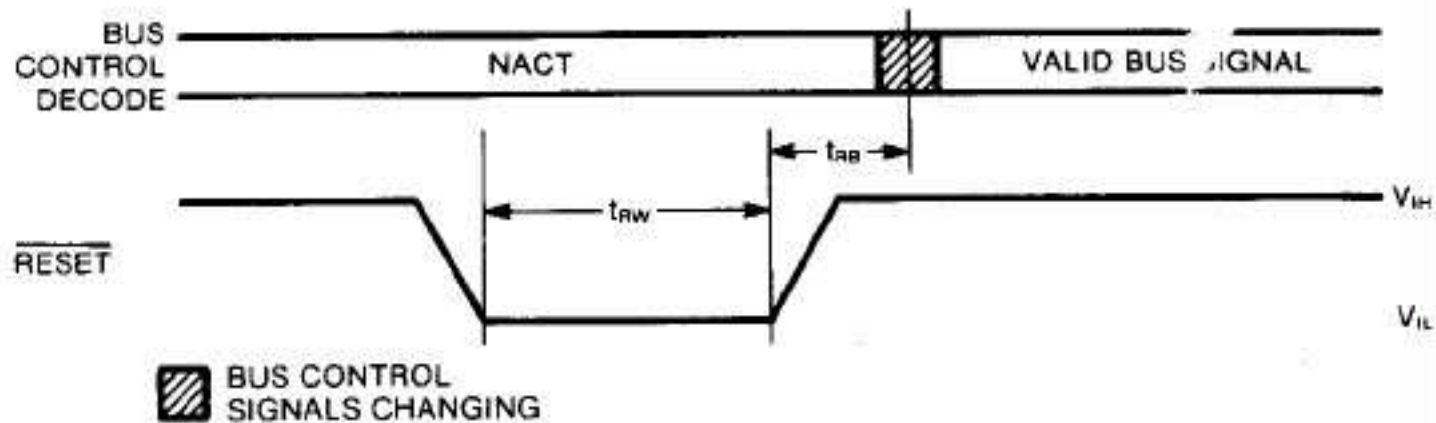
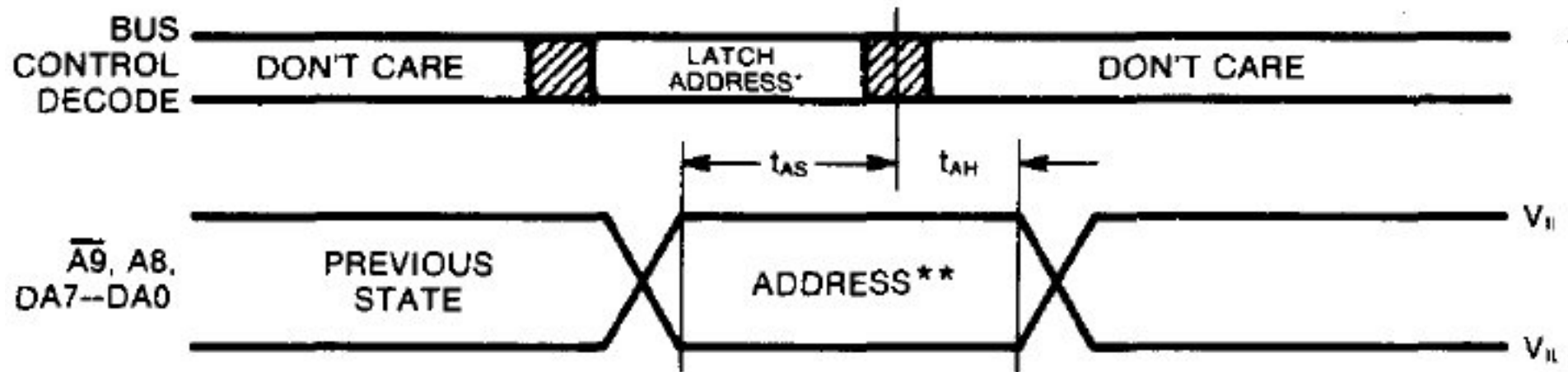


Fig. 37 RESET TIMING




# De aansturing

## LATCH ADDRESS TIMING



 BUS CONTROL SIGNALS CHANGING

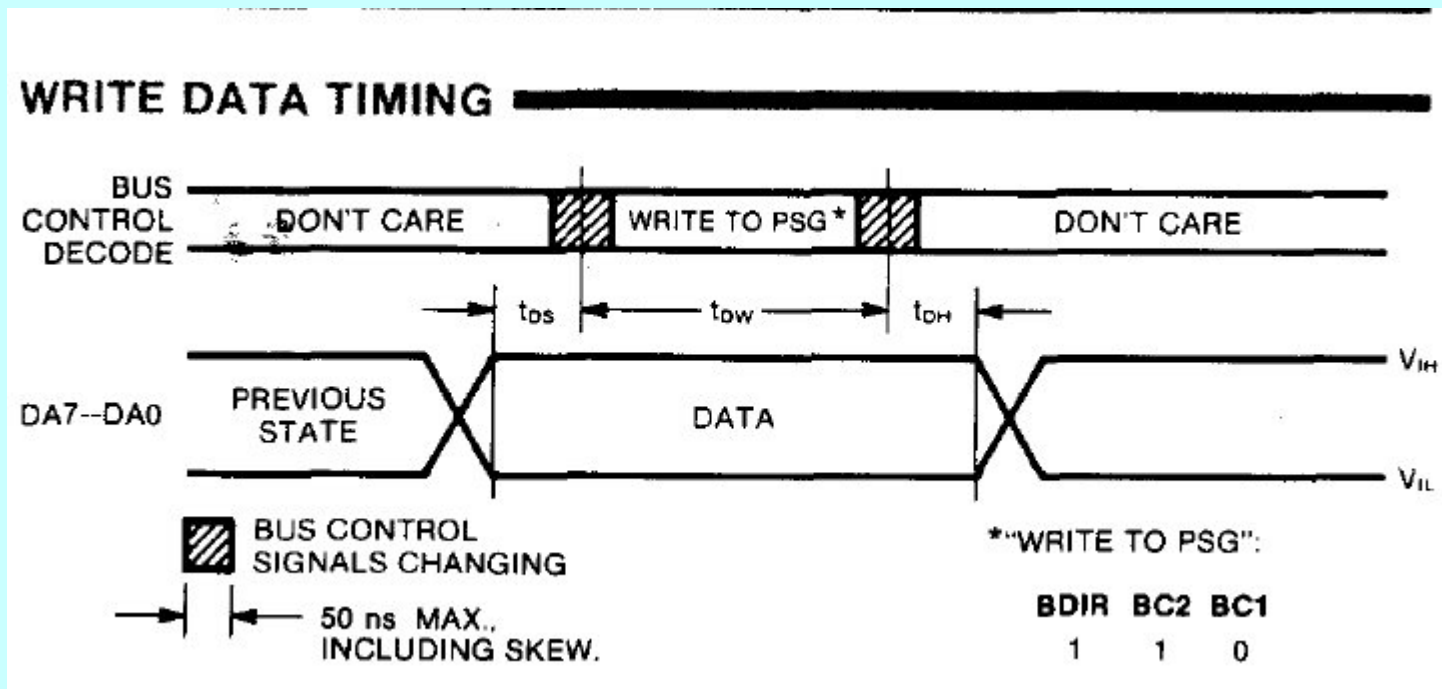
 50 ns MAX., INCLUDING SKEW.

\*\*REFER TO PARAGRAPH 2.1.1 FOR A DESCRIPTION OF "VALID" PSG ADDRESSING.

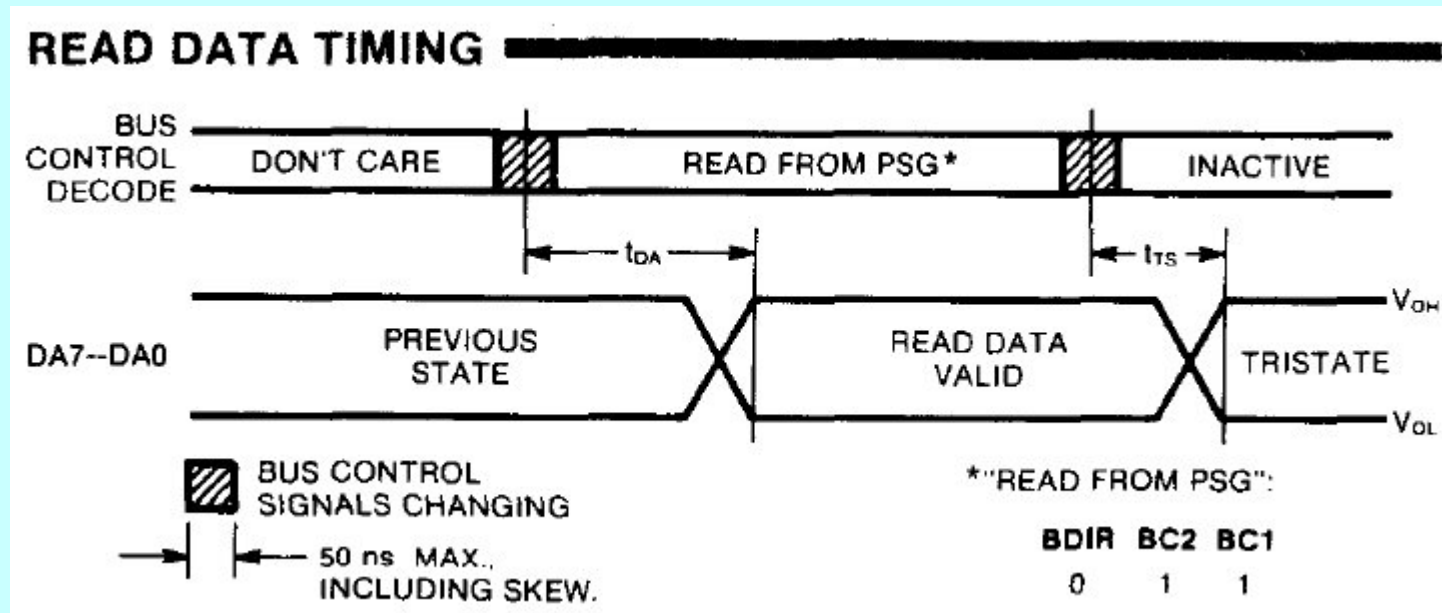
\*ANY COMBINATION OF BDIR, BC2, BC1 WHICH DECODE "LATCH ADDRESS":

	BDIR	BC2	BC1
	0	0	1
or	1	0	0
or	1	1	1

# Meer aansturing



# Nog meer aansturing



Characteristic	Sym	Min.	Typ.*	Max.	Units	Conditions
<b>Clock Input</b>						
Frequency	$f_c$	1.0	—	2.0	MHz	} Fig. 36
Rise time	$t_r$	—	—	50	ns	
Fall time	$t_f$	—	—	50	ns	
Duty Cycle	—	25	50	75	%	
<b>Bus Signals (BDIR, BC2, BC1)</b>						
Associative Delay Time	$t_{BD}$	—	—	50	ns	} Fig. 37
<b>Reset</b>						
Reset Pulse Width	$t_{RW}$	500	—	—	ns	} Fig. 38
Reset to Bus Control Delay Time	$t_{RB}$	100	—	—	ns	
<b>A9, A8, DA7--DA0 (Address Mode)</b>						
Address Setup Time	$t_{AS}$	400	—	—	ns	} Fig. 39
Address Hold Time	$t_{AH}$	100	—	—	ns	
<b>DA7--DA0 (Write Mode)</b>						
Write Data Pulse Width	$t_{DW}$	500	—	10,000	ns	} Fig. 40
Write Data Setup Time	$t_{DS}$	50	—	—	ns	
Write Data Hold Time	$t_{DH}$	100	—	—	ns	
<b>DA7--DA0 (Read Mode)</b>						
Read Data Access Time	$t_{DA}$	—	250	500	ns	} Fig. 40
<b>DA7--DA0 (Inactive Mode)</b>						
Tristate Delay Time	$t_{TS}$	—	100	200	ns	