PSG Manual

The PSG (Programmable Sound Generator) contains three tone generators and one noise generator. Each of the tone and noise generators can be distributed left and right, enabling a pseudo stereo effect to be generated. (See "System Control Port".)

Control of the PSG itself, which is described below, is performed by means of the write operation to I/O area 7FH.

The basic clock is 3.579545 MHz. The data to be sent from the CPU is immediately latched in the PSG, hence there is no need for a wait. The sound output goes OFF in the case of a power-on reset. Design the software so that the output goes OFF at the beginning of the program as well.

[1] Tone generator

Each tone generator consists of a frequency setting section (programmable counter) and a level setting section (programmable attenuator).

(1) Method of calculating the 10-bit frequency division ratio n

At the frequency setting section, the basic clock is frequency-divided to 1/32. This is further frequency divided by the tone counter set by the 10 bits F9 (MSB: top bit) to F0 (LSB: bottom bit).

Consequently, the basic clock frequency is divided by 32, then the desired frequency can be output by setting the value obtained by dividing the frequency-divided clock by the desired frequency in F9 to F0.

$$n = N/(32 \times f)$$

Where n = 10-bit frequency division ratio (F9 to F0)
N = Basic clock
f = Desired frequency

(2) Tone frequency setting

Set the 10-bit frequency division ratio (F9 to F0) in the tone counter in order to obtain the desired frequency. The 1st and 2nd bytes are identified by means of the top bit.

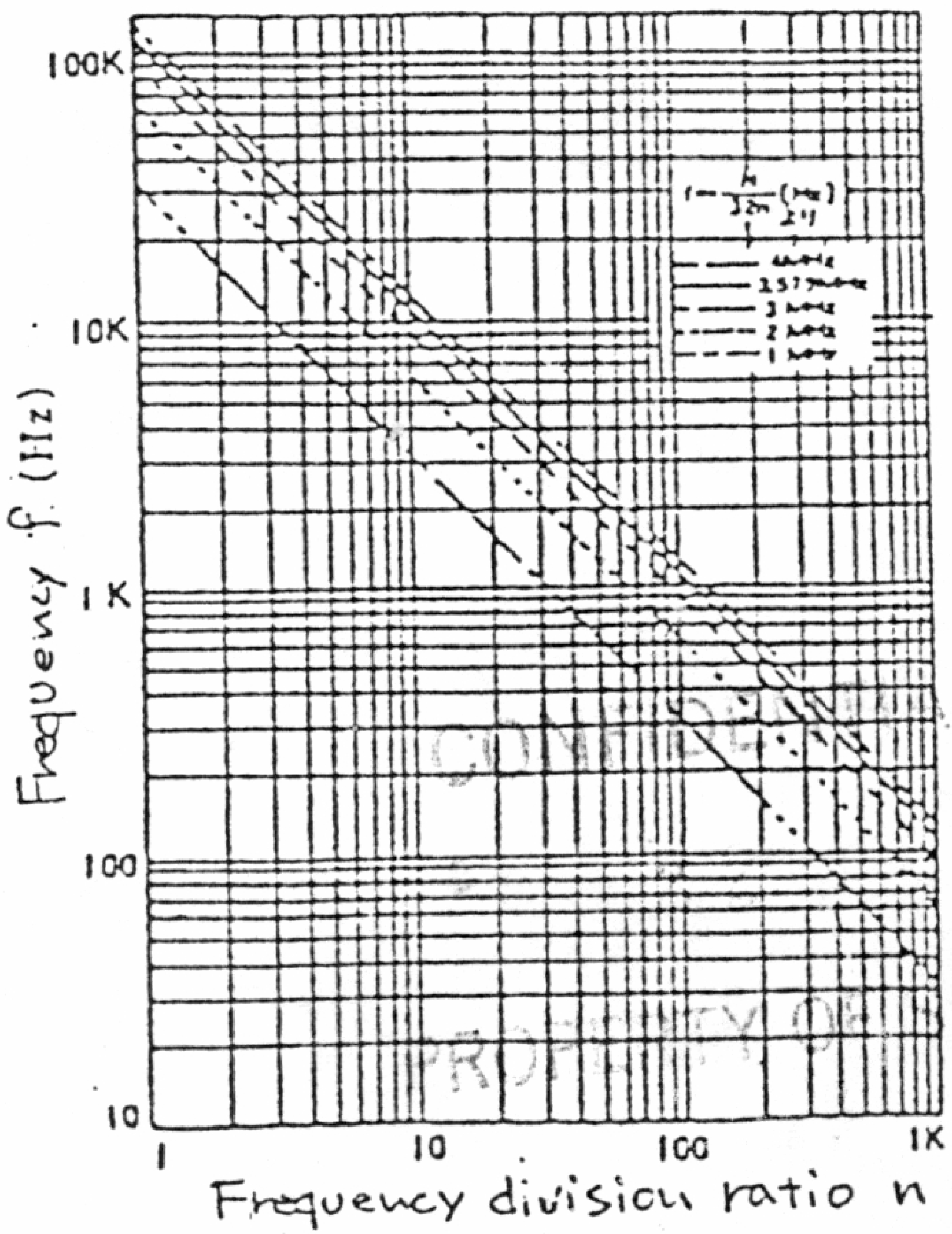
1st byte

D7	D6	D5	D4	D3	D2	D1	D0
*	REG. ADDR.			n			
1	R2	R1	R0	F3	F2	F1	F0

2nd byte

D7	D6	D5	D4	D3	D2	D1	D0
*		n					
O	×	F9	F8	F7	F6	F5	F4

R2	R1	R0	Control register allocation	1st
O	O	O	Tone generator 1	8×
O	1	O	Tone generator 2	A×
1	O	O	Tone generator 3	C×



Frequency division ratio with respect to the output frequency

(3) Example of frequency setting

Consider an example in which the basic clock frequency is 3.579545 MHz and the desired frequency of 440 Hz is output from TONE 1. (corresponding to "A" on the musical scale)

a. Calculation of frequency division ratio n

$$n = N/(32 \times f)$$
$$= 3579545/(32 \times 440)$$
$$\fallingdotseq 254.229$$

n is a 10-bit integer, hence the nearest integral value is 254.

Consequently, the frequency actually output is

$$f = N/(32 \times n)$$
$$= 3579545/(32 \times 254)$$
$$\fallingdotseq 440.397 \text{ (Hz)}$$

Here, the pitch error $\triangle C$ is obtained according to the following equation.

$$\triangle C = \{(f' - f)/f\}/(\sqrt[1200]{2}-1)$$
$$\fallingdotseq \{(440.397-440)/440\}/(\sqrt[1200]{2}-1)$$
$$\fallingdotseq (0.397/440)/0.000578 \fallingdotseq 1.56$$

f: True frequency f': Actual frequency $\sqrt[1200]{2}$

b. Data sent to PSG

$$n = 254$$
$$= 0011111110B$$

1st byte

*	REG. ADDR.			n			
1	R2	R1	R0	F3	F2	F1	F0

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	1	1	0

2nd byte

*		n					
0	×	F9	F8	F7	F6	F5	F4

D7	D6	D5	D4	D3	D2	D1	D0
0	×	0	0	1	1	1	1

(4) Tone level setting

The frequency set by the tone generator is sent to the level setting section where the volume level is set. The level setting section is a programmable attenuator which enables the volume level to be set in 16 steps from 0 dB to OFF according to a 4-bit attenuation value.

1st byte only

D7	D6	D5	D4	D3	D2	D1	D0

*	REG. ADDR.			ATT. DATA			
1	R2	R1	R0	A3	A2	A1	A0

R2	R1	R0	Control register allocation	HEX
0	0	1	Tone 1 attenuation	9×
0	1	1	Tone 2 attenuation	B×
1	0	1	Tone 3 attenuation	D×

Attenuation

[db]	A3 A2 A1 A0	HEX
0	0 0 0 0	×0
2	0 0 0 1	×1
4	0 0 1 0	×2
6	0 0 1 1	×3
8	0 1 0 0	×4
10	0 1 0 1	×5
12	0 1 1 0	×6
14	0 1 1 1	×7

[db]	A3 A2 A1 A0	HEX
16	1 0 0 0	×8
18	1 0 0 1	×9
20	1 0 1 0	×A
22	1 0 1 1	×B
24	1 1 0 0	×C
26	1 1 0 1	×D
28	1 1 1 0	×E
OFF	1 1 1 1	×F

[2] Noise generator

The noise generator consists of a noise generator circuit and a level setting section. The source of the noise supplied from the noise generator circuit is a shift register with EX-OR feedback. Each time the noise control register changes, the shift register is cleared.

The shift clock of this shift register is determined by four modes that are in turn determined by NF0 and NF1. If NF0 = NF1 = 0, for example, the shift clock becomes (N/32)/16. In this case, if FB = 0, this shift clock will be frequency-divided by 16, resulting in synchronous noise of a frequency of N/(32 x 16 x 16). If FB = 1, the shift register will be driven by this shift clock with EX-OR feedback, resulting in the generation of white noise.

(1) Noise generator circuit control

1st byte only

D7	D6	D5	D4	D3	D2	D1	D0
*	REG. ADDR.					SHIFT	
1	1	1	0	×	FB	NF1	NF0

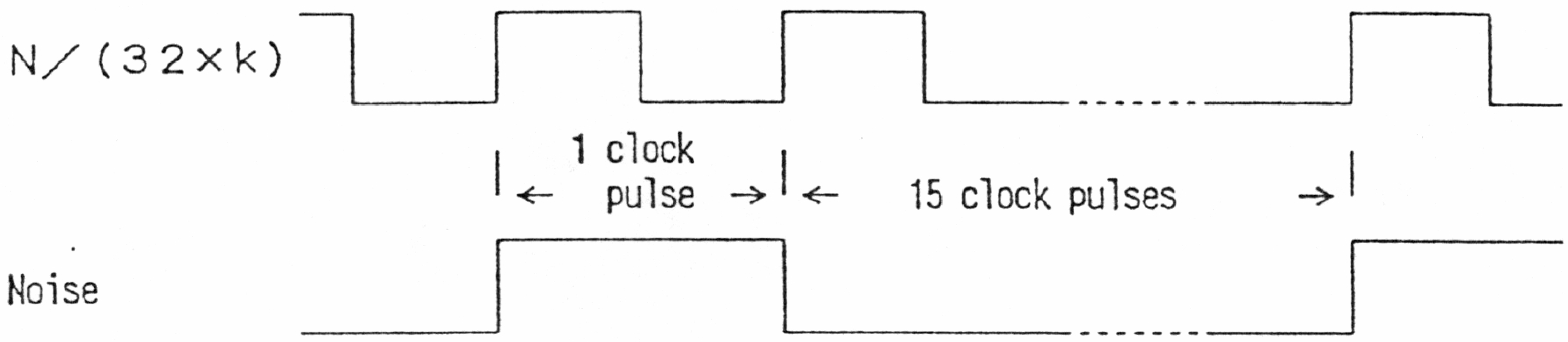
FB	Noise Generation
0	Synchronous Noise
1	White noise

NF1	NF0	Shift clock	k
0	0	(N/32)/k	16
0	1	(N/32)/k	32
1	0	(N/32)/k	64
1	1	Tone generator 3	

N : Basic clock

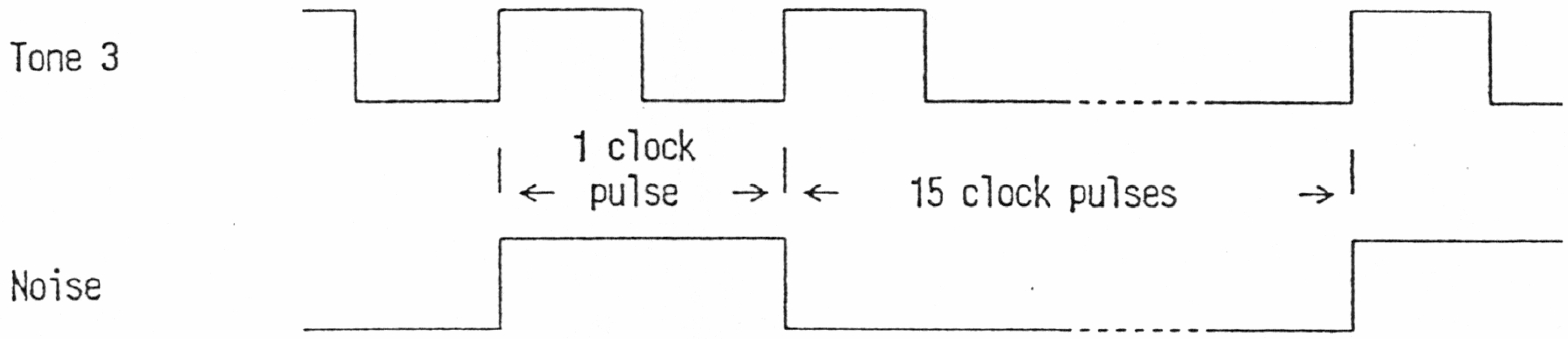
←At this time, the tone of the noise can be varied continuously

a. Synchronous noise (FB = 0)
When ONFO = NF1 = Value other than 1



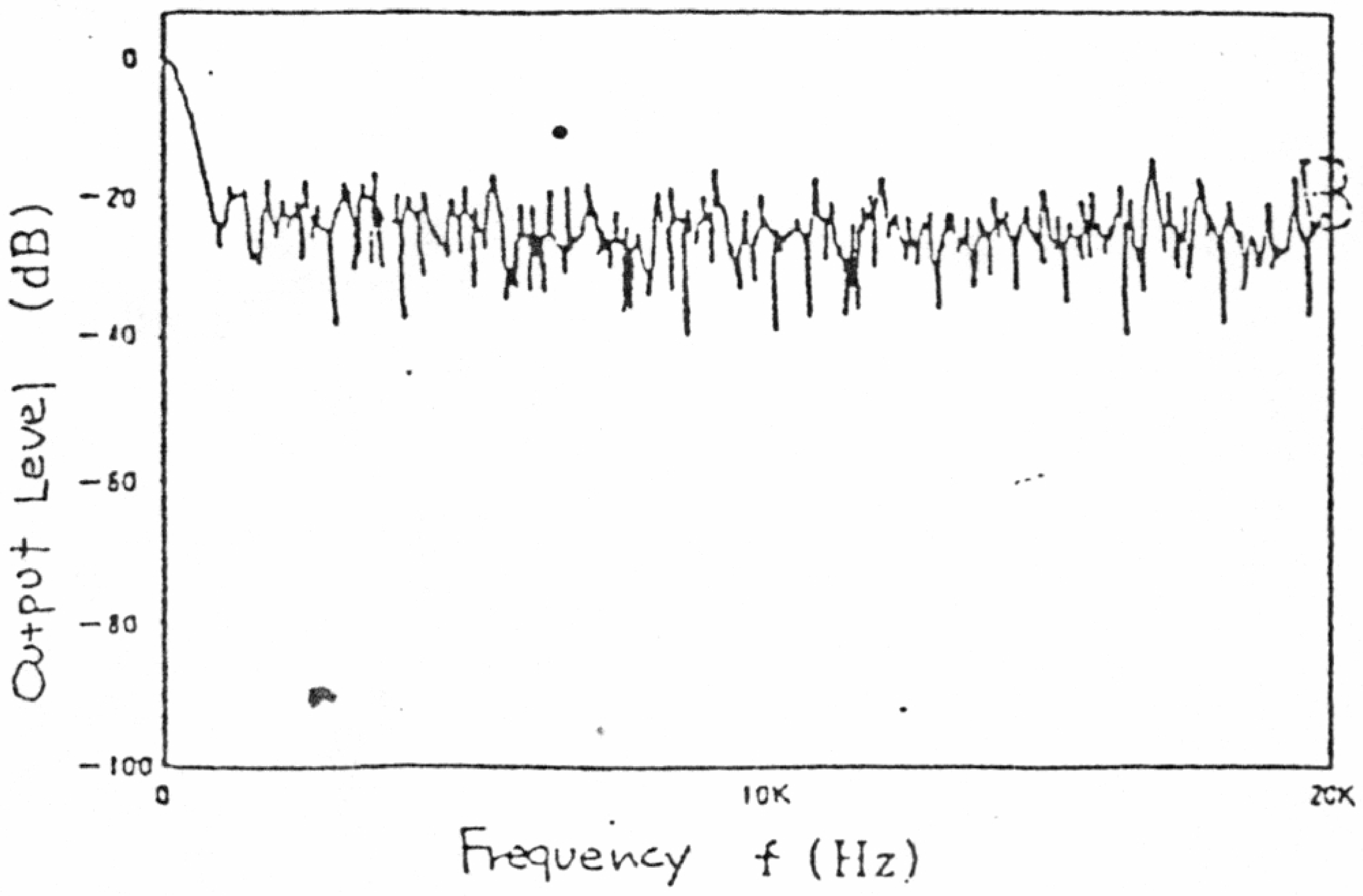
* The noise frequency is (N/(32 x k))/16 = N/(32 x k x 16)

When NFO = NF1 = 1 (Control by tone 3)



* The noise frequency is (frequency TONE 3)/16 = N/(32 x n x 16)

b.White Noise (FB = 1)
Spectrum when NFO = 0 NF1 = 1 n=1



(2) Noise level setting
1st byte only

D7	D6	D5	D4	D3	D2	D1	D0

*	REG. ADDR.			ATT. DATA			
1	1	1	1	A3	A2	A1	A0

* The attenuation control is the same as for "Tone".

[3] Register address feed

PSG uses the three bits R2 to R0 of the 1st byte to judge which control register the data has been sent from.

R2	R1	R0	Control register allocation	1st
0	0	0	Tone 1 frequency division ratio	8×
0	0	1	Tone 1 attenuation	9×
0	1	0	Tone 2 frequency division ratio	A×
0	1	1	Tone 2 attenuation	B×
1	0	0	Tone 3 frequency division ratio	C×
1	0	1	Tone 3 attenuation	D×
1	1	0	Noise generator circuit control	E×
1	1	1	noise attenuation	F×

[4] Correlation between the sound elements and PSG

Sound element	Physical element	Correlation with PSG
Pitch of sound	Frequency	[1] - (2) Tone frequency setting, [2] - (1) Noise generator circuit control
Tone	Hermonic components	This is mainly related to wave length. In the tone generation mode, the PSG can output three frequencies simultaneously from only a 50% duty pulse waveform. Consequently, by combining attenuation control with this mode, the harmonic components can be controlled. In the synchronous noise mode, a 6.25% duty pulse waveform.
Strength of tone	Amplitude	As described in [1] - (3) and [2] - (2), the attenuation of the three tones and noise can be controlled by 4-bit data.
Way in which sound is emitted.	Envelope	The wave length at left can be realized by using external data to control each attenuation. This can be done in the range where the 4-bit attenuation data is rewritten and envelope sequence control performed at each step. The tone and noise frequencies can be controlled by the range in which component control can be performed.

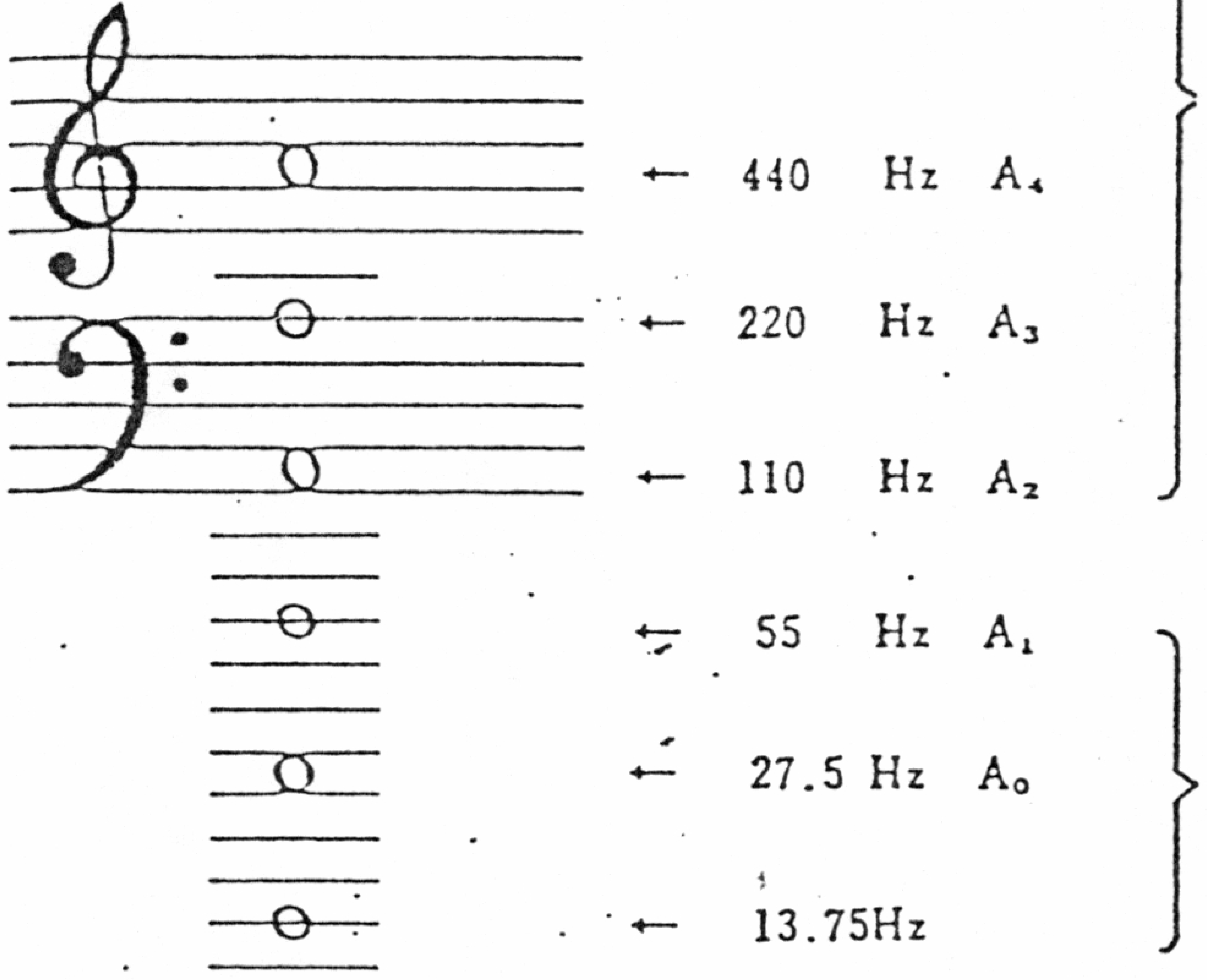
(next page)
Relation between musical interval and frequency division ratio for scale divided equally into 12 parts (basic clock: 3.579545 MHz)

Musical interval	Frequency division ratio	HEX		PSG output [Hz]	Actual frequency [Hz]
		1st	2nd		
A 2	1017	X9	3F	109.991	110.000
A#2	960	X0	3C	116.522	116.541
B 2	906	XA	38	123.467	123.471
C 3	855	X7	35	130.832	130.813
C#3	807	X7	32	138.613	138.591
D 3	762	XA	2F	146.799	146.832
D#3	719	XF	2C	155.578	155.563
E 3	679	X7	2A	164.744	164.814
F 3	641	X1	28	174.510	174.614
F#3	605	XD	25	184.894	184.997
G 3	571	XB	23	195.904	195.998
G#3	539	XB	21	207.534	207.652
A 3	508	XC	1F	220.199	220.000
A#3	480	X0	1E	233.044	233.082
B 3	453	X5	1C	246.934	246.942
C 4	428	XC	1A	261.357	261.626
C#4	404	X4	19	276.884	277.183
D 4	381	XD	17	293.598	293.665
D#4	360	X8	16	310.725	311.127
E 4	339	X3	15	329.973	329.628
F 4	320	X0	14	349.565	349.228
F#4	302	XE	12	370.400	369.994
G 4	285	XD	11	392.495	391.995
G#4	269	XD	10	415.840	415.305
A 4	254	XE	0F	440.397	440.000
A#4	240	X0	0F	466.087	466.164
B 4	226	X2	0E	494.960	493.883
C 5	214	X6	0D	522.715	523.251
C#5	202	XA	0C	553.767	554.365
D 5	190	XE	0B	588.742	587.330
D#5	180	X4	0B	621.450	622.254
E 5	170	XA	0A	658.005	659.255
F 5	160	X0	0A	699.131	698.456
F#5	151	X7	09	740.801	739.989
G 5	143	XF	08	782.244	783.991
G#5	135	X7	08	828.600	830.609
A 5	127	XF	07	880.795	880.000
A#5	120	X8	07	932.174	932.328
B 5	113	X1	07	989.920	987.767
C 6	107	XB	06	1045.429	1046.502
C#6	101	X5	06	1107.534	1108.731
D 6	95	XF	05	1177.484	1174.659
D#6	90	XA	05	1242.899	1244.508
E 6	85	X5	05	1316.011	1318.510
F 6	80	X0	05	1398.262	1396.913
F#6	76	XC	04	1471.854	1479.978
G 6	71	X7	04	1575.506	1567.982
G#6	67	X3	04	1669.566	1661.219
A 6	64	X0	04	1747.827	1760.000
A#6	60	XC	03	1864.349	1864.655
B 6	57	X9	03	1962.473	1975.533

* f = 3579545/(32 x n)   n: 10-bit frequency division   Maximum frequency is n = 1
* In the previous table, musical interval was calculated on the basis of 440 Hz as concert pitch.
* Regarding HEX, the X part is as follows:

Tone 1 → 8
Tone 2 → A
Tone 3 → C
1st is the 1st byte
2nd is the 2nd byte

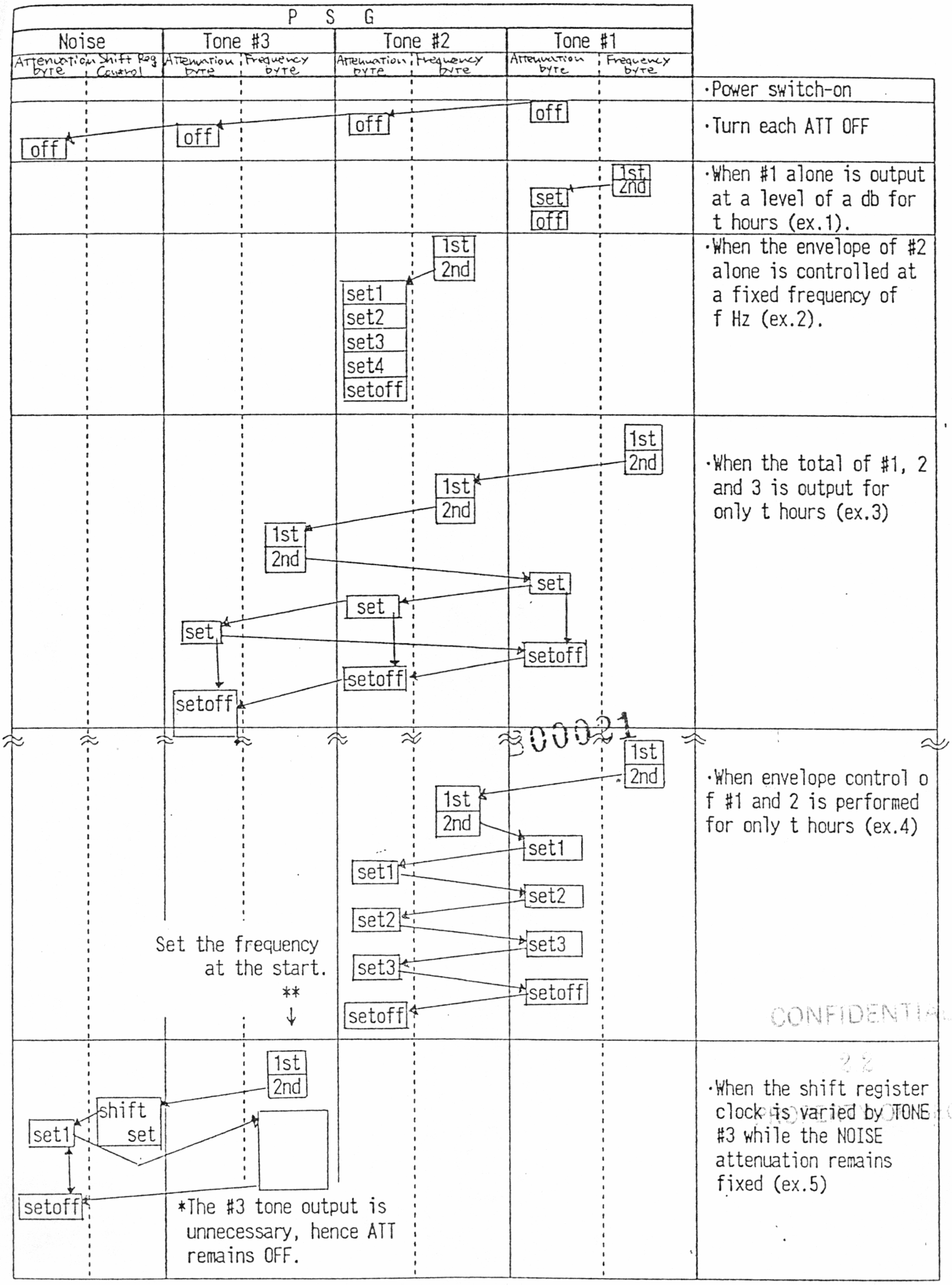
The upper limit is 3579545/32·1



TONE 1 to 3
Duty 50%

Base tone in synchronous noise mode
Duty 6.25%

Data input timming chart



·Power switch-on

·Turn each ATT OFF

·When #1 alone is output at a level of a db for t hours (ex.1).

·When the envelope of #2 alone is controlled at a fixed frequency of f Hz (ex.2).

·When the total of #1, 2 and 3 is output for only t hours (ex.3)

·When envelope control of #1 and 2 is performed for only t hours (ex.4)

·When the shift register clock is varied by TONE #3 while the NOISE attenuation remains fixed (ex.5)

*The #3 tone output is unnecessary, hence ATT remains OFF.

Flow of data control timing (t is controlled by an external processor.)

If a frequency of no greater than that generated by the tone generator is output, the outputs shown in the table below will be obtained due to the synchronous noise mode of the noise generator section. (Basic clock: 3.579545 MHz)

Musical interval	Frequency division ratio	HEX (TONE3)		PSG output [Hz]	Actual frequency [Hz]
		1st	2nd		
C 0	428	CC	1A	16.335	16.352
C# 0	404	C4	19	17.305	17.324
D 0	381	CD	17	18.350	18.354
D# 0	360	C8	16	19.420	19.445
E 0	339	C3	15	20.623	20.602
F 0	320	C0	14	21.848	21.827
F# 0	302	CE	12	23.150	23.125
G 0	285	CD	11	24.531	24.500
G# 0	269	CD	10	25.990	25.957
A 0	254	CE	0F	27.525	27.500
A# 0	240	C0	0F	29.130	29.135
B 0	226	C2	0E	30.935	30.868
C 1	214	C6	0D	32.670	32.703
C# 1	202	CA	0C	34.610	34.648
D 1	190	CE	0B	36.796	36.708
D# 1	180	C4	0B	38.841	38.891
E 1	170	CA	0A	41.125	41.203
F 1	160	C0	0A	43.696	43.654
F# 1	151	C7	09	46.300	46.249
G 1	143	CF	08	48.890	48.999
G# 1	135	C7	08	51.787	51.913
A 1	127	CF	07	55.050	55.000
A# 1	120	C8	07	58.261	58.270
B 1	113	C1	07	61.870	61.735
C 2	107	CB	06	65.339	65.406
C# 2	101	C5	06	69.221	69.296
D 2	95	CF	05	73.593	73.416
D# 2	90	CA	05	77.681	77.782
E 2	85	C5	05	82.251	82.407
F 2	80	C0	05	87.391	87.307
F# 2	76	CC	04	91.991	92.499
G 2	71	C7	04	98.469	97.999
G# 2	67	C3	04	104.348	103.826

* Set E3H for control of the noise generator circuit (synchronous noise mode).
f = TONE 3 frequency/16 = 3579545/(32 x n x16)

PSG manual END

plementary description for manual

System control port

③ I/O port 02H (Read/Write)

* Normally, be sure to make the status of this port "1". When using this value, first, set the status to "1" after the first NMI is generated, then subsequently set it to "0". If you fail to do this, the next NMI will not be generated.

Addition:

An NMI is enabled after the execution of one command from when the port is set to "0". This is to prevent the NMI from becoming active once again in the NMI routine. Normally, therefore, perform the following processing.

```
        NMI:
            .
            .
            .
            .
            LD      A,11XXXXXXB
            OUT     (002H),A
            LD      A,01XXXXXXB
            OUT     (002H),A
            RETN                    ; An NMI is enabled after this command.
```

B. System control port

⑥ I/O port 05H (Read/Write)

Mode setting for serial communications

* INT

* There is no need to perform an operation such as that of I/O port 02H.

Addition

When using this function (serial communications NMI), set NINT of I/O port 02H to the disable state ("1"). An NMI will be generated at the fall of the pulse at the NMI terminal.
If, however, a serial communications NMI is generated, the NMI terminal will go LOW, preventing the next NMI from becoming active. The NMI terminal is made HIGH as a result of reading the data of I/O port 04H, so read the data each time an NMI is generated. If it is conceivable that the NMI terminal may already be LOW at the start of the communications, perform a "dummy" read operation onc e.

B. System control port

⑦ I/O port 06H (Write Only)

Left-right distribution of sound Supplementary explanation When the headphones are plugged in, the output from the speaker is cut off and instead the sound will be heard in stereo from the headphones. When the earphones are not plugged in, the sound will be heard from the speaker in monaural. In the latter case, the distribution of the sound from all channels (three tones & noise) will be enabled. If the output from the left and right channels was disabled not by attenuator control but by distribution, the sound will not be heard from the headphones but will be heard from the speaker. To turn off both the left and right channels of the speaker, use the PSG by PSG side control.

C. Communications

① Connecting the communications cable Cross-connect the game gear communications cable as shown below.

Communications connector

1	PC0	To opposite side PC2
2	PC1	To opposite side PC3
3	PC2	To opposite side PC0
4	PC3	To opposite side PC1
5	+5V	
6	PC4	To opposite side PC5
7	PC6	To opposite side PC6
8	GND	To opposite side GND
9	PC5	To opposite side PC4
10	NC	

② Parallel communications

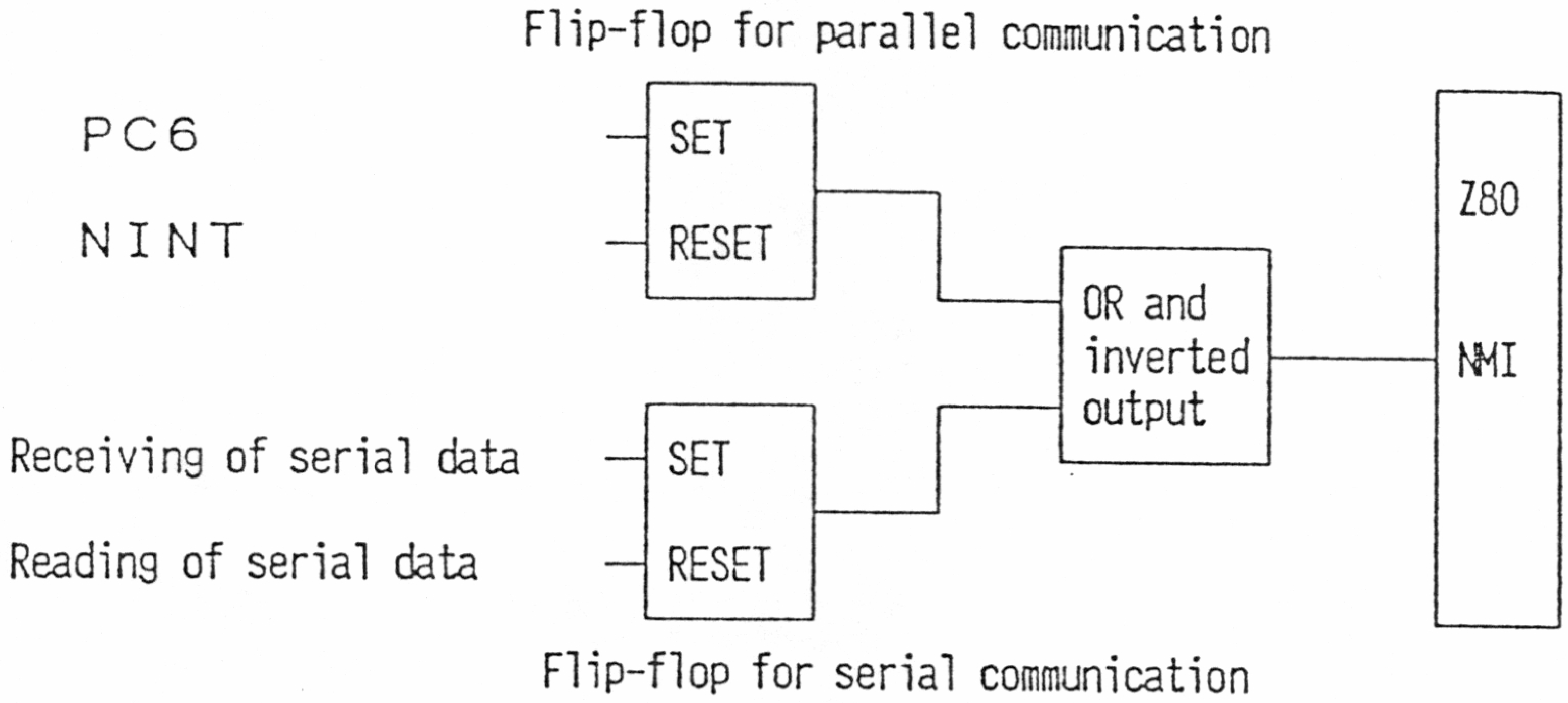
PC0 to PC6 can be set to an arbitrary input or output by means of the control register of the I/O port. Be sure to set the connecting terminals so that the terminal on one side is the output, and that on the opposite side is the input. (Never make the terminals on both sides the output.) In the case of parallel communications, control the exchange of data either by polling using software (check the data), or by applying an interrupt (NMI) using PC6. When applying an NMI using PC6, however, it is necessary to take noise into account because an NMI will be generated by a momentary change in PC6.

③ Serial communications

Serial communications can be performed in one of two single directions, from PC4 (output from one's own side) -> PC5 (input to opposite side), or from PC5 (input to one's own side) <- PC4 (output from opposite side). Serial and parallel conversion and interrupt (NMI) generation (when data is received) accompanying the receiving or sending of data take place automatically when the hardware is connected to these terminals. To perform serial communications, set TON and RON of I/O port 05H to "1". By doing this, PC4 will automatically become the output, and PC5 the input. These settings will take priority over the PC4 and PC5 input/output settings. Bits other than those of PC4 and PC5 will become the settings of I/O port 02H. (Like (2), never make both terminals the output.) When performing serial communications only, be sure to set NINT of I/O port 02H to prevent PC6 from generating an NMI.

④ Coexistence of parallel and serial communications

When performing serial communications, PC4 and PC5 are used to send and receive serial data. Parallel communications can be performed using the bits other than these. An NMI can be generated by PC6 and also by receiving of serial data. In the former case, care must be taken because PC6 goes not have a data receiving flag such as the serial RXRD. The blocks of the circuit used to generate these NMI are shown below.



An NMI is generated when one of the two reset flip-flops is set. (This is because an NMI is generated not when the pulse level is LOW but when the pulse falls.) It should be appreciated that it is necessary to reset the set flip-flop so that the next NMI can be generated.

- END -