DIGITAL PROGRAMMABLE ALGORITHM SYNTHESIZER

DX7/9

SERVICE MANUAL

■ CONTENTS(目次)

006569

SINCE 1887 YAMAHA
NIPPON GAKKI CO., LTD. HAMAMATSU, JAPAN
'83.4 2.4K-3211 Printed in Japan

SPECIFICATIONS（仕様）

●DX7

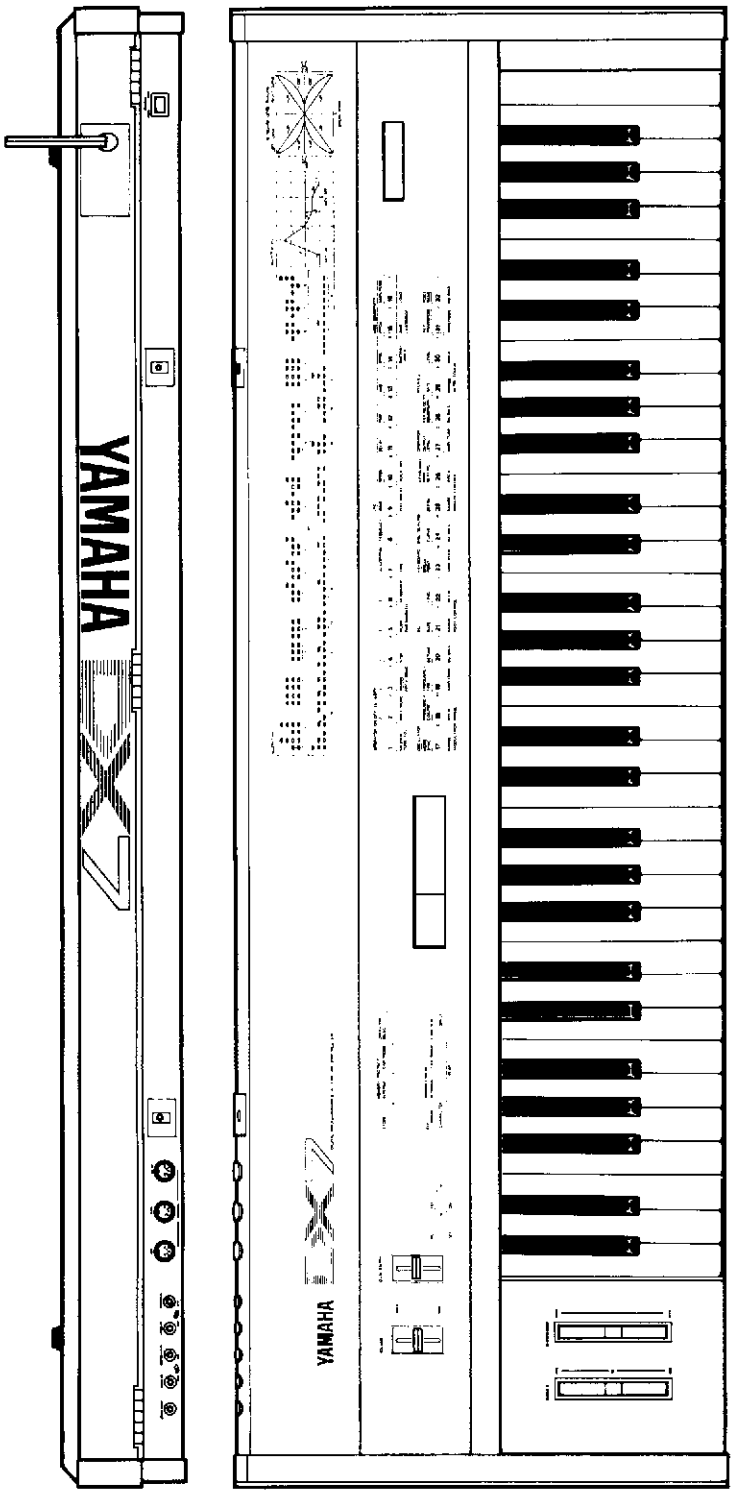
Keyboard **鍵盤**	61 keys: New Light Touch Keyboard
Sound Source **音源**	FM Tone Generator: 6 operator 32 algorithm
Number of sounds produced simultaneously	16
Sound Character Memory **音色メモリー**	32 Bank (32 Memory)
External ROM Memory **外部ROMメモリー**	32 Bank x 2 (64 Memory)
External RAM Memory **外部RAMメモリー**	32 Bank (32 Memory)
Controls **コントロール**	VOLUME, DATA ENTRY, YES/NO, ON/OFF, STORE, MEMORY PROTECT (INTERNAL, CARTRIDGE), OPERATOR SELECT, PLAY-MEMORY SELECT (INTERNAL/CARTRIDGE 1~32), EDIT/COMPARE (OPERATOR ON-OFF/EG COPY 1~6), ALGORITHM, FEEDBACK, LFO (WAVE, SPEED, DELAY, PMD, AMD, SYNC), MOD SENSITIVITY (PITCH, AMPLITUDE), OSCILLATOR (MODE/SYNC, FREQUENCY COARSE, FREQUENCY FINE, DETUNE), EG (RATE, LEVEL), KEYBOARD LEVEL SCALING (BREAK POINT, CURVE, DEPTH), KEYBOARD RATE SCALING, OPERATOR (OUTPUT LEVEL, KEY VELOCITY SENSITIVITY), PITCH EG (RATE, LEVEL), KEY TRANSPOSE, VOICE (NAME), FUNCTION (MASTER TUNE ADJ, POLY/MONO, PITCH BEND (RANGE, STEP), PORTAMENT (MODE, GLISSANDO, TIME), EDIT RECALL, VOICE INIT, BATTERY CHECK, CARTRIDGE (SAVE, LOAD), MODULATION WHEEL (RANGE, PITCH, AMPLITUDE, EG BIAS), FOOT CONTROL (RANGE, PITCH, AMPLITUDE, EG BIAS), BREATH CONTROL (RANGE, PITCH, AMPLITUDE, EG BIAS), AFTER TOUCH (RANGE, PITCH, AMPLITUDE, EG BIAS)), LCD DISPLAY, CARTRIDGE INTERFACE, PITCH WHEEL, MODULATION WHEEL
Connecting Terminal **接続端子**	OUTPUT, PHONES
Control Terminal **コントロール端子**	FOOT SWITCH (SUSTAIN, PORTAMENT), FOOT CONTROL (VOLUME, MODULATION), BREATH CONTROL, MIDI (IN, OUT, THRU)
Dimensions/Weight **寸法・重量**	101.8W x 10.2H x 32.9D cm (40" x 4" x 13"), 14.2kg (28.6 lbs.)
Power Consumption	40W (UL, CSA, GENERAL)
定格消費電力	30W (JAPAN)
Accessories **標準装備品**	Music Stand, ROM Cartridge x 2 (68 sound character x 2) （68音色×2）

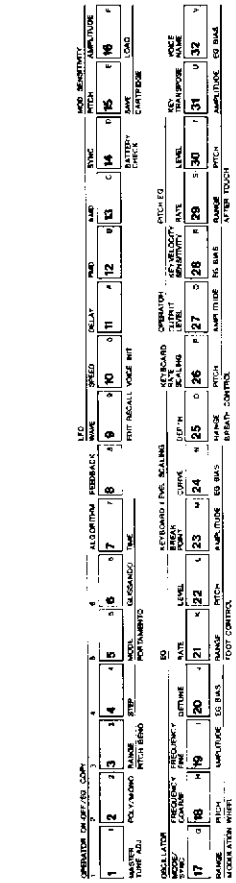
*Specifications and design are subject to change without notice for improvement.

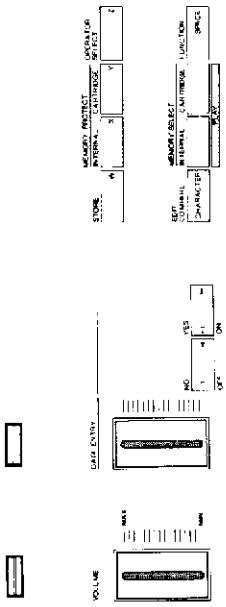
※規格および仕様は、改良のため予告なく変更する場合がありますので、ご了承ください。

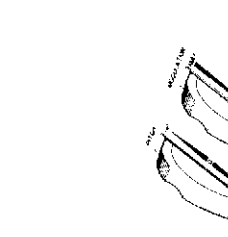
PANEL LAYOUT

● DX7

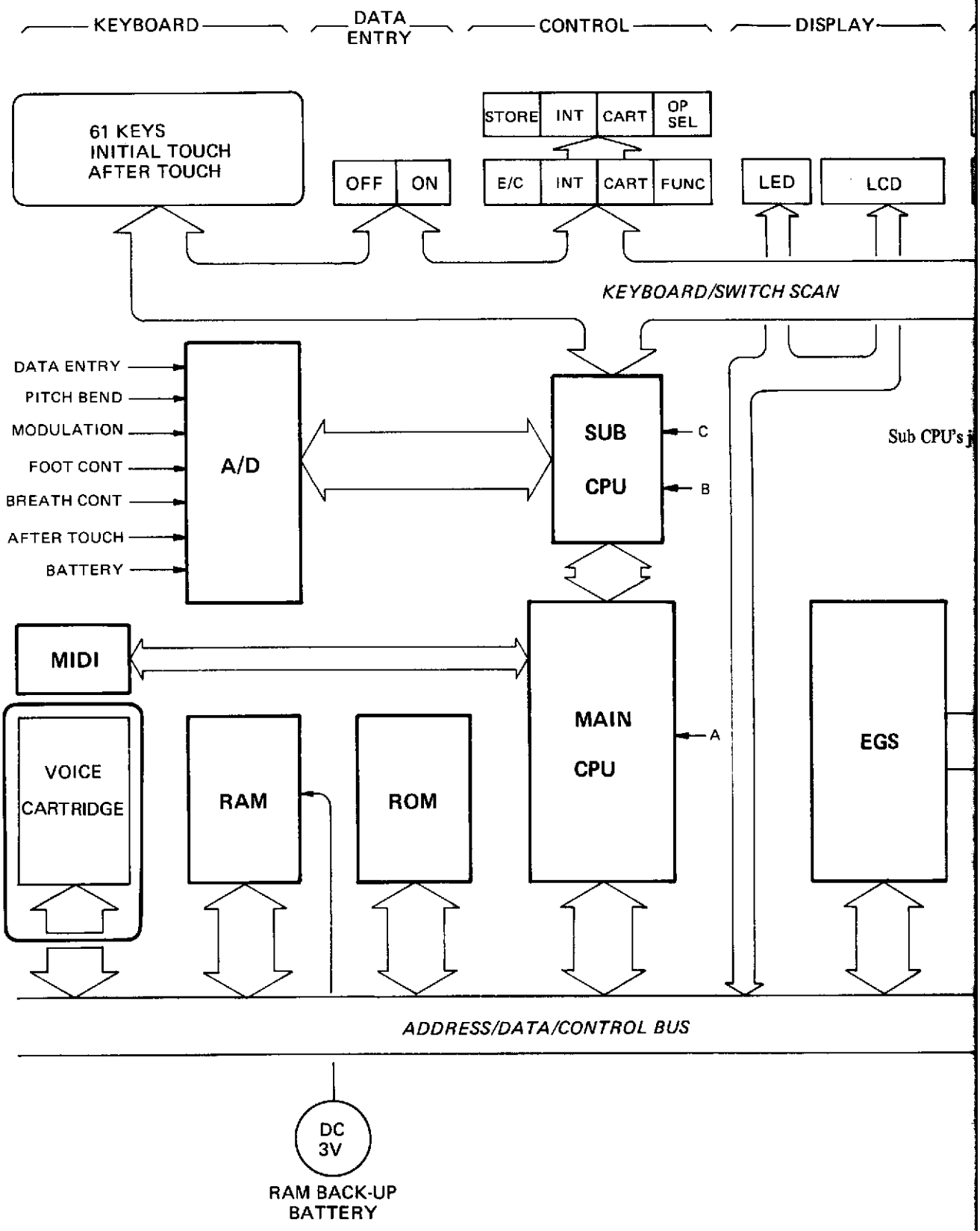


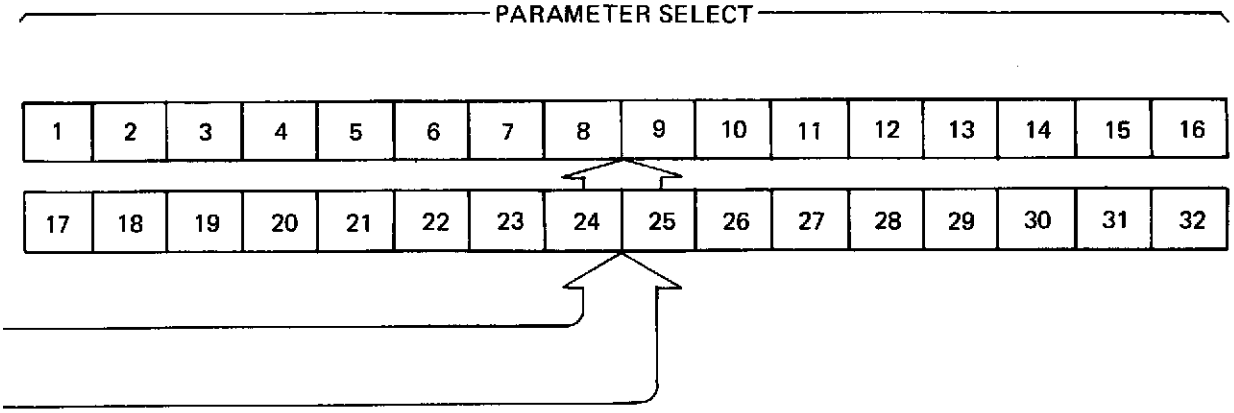






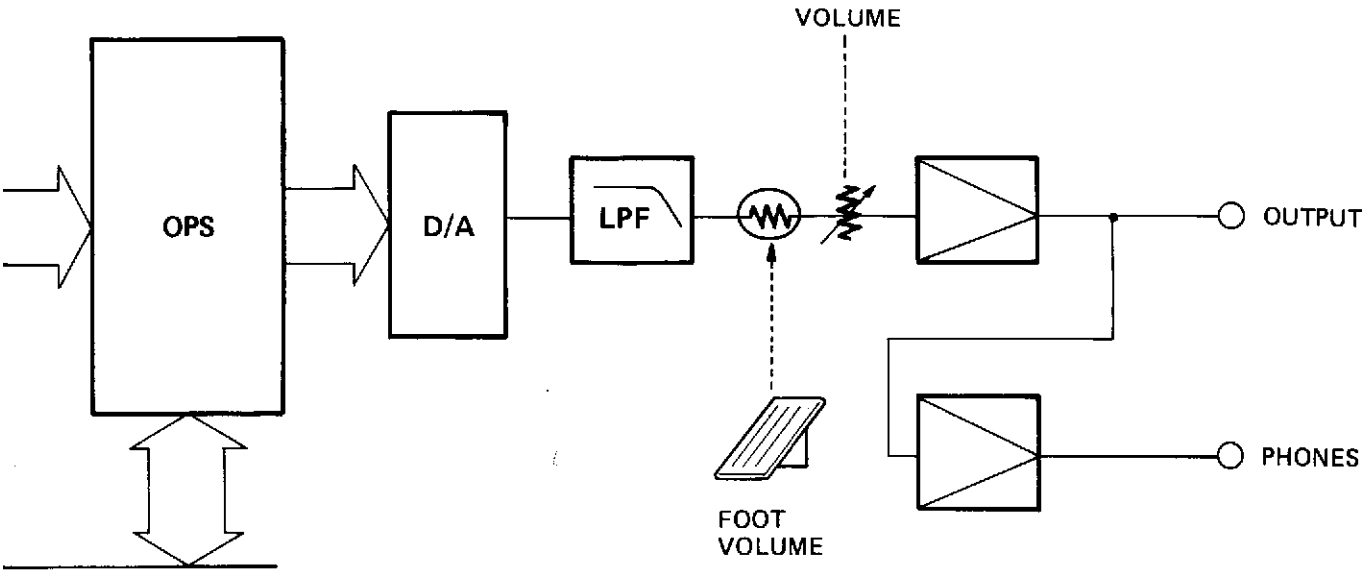
DX7 BLOCK DIAGRAM

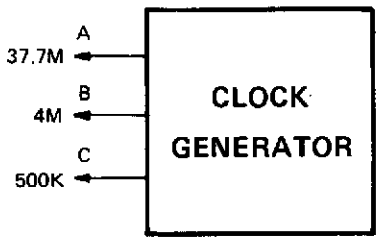


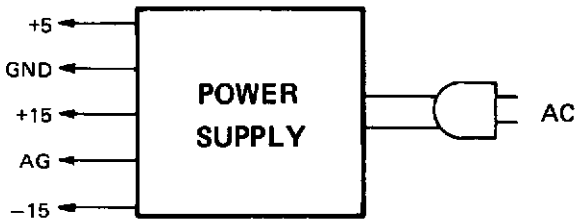


's job
1. Keyboard scanning of ON/OFF and touch data
2. Panel switch scanning
3. Analog voltage scanning
4. Data transmission to main CPU

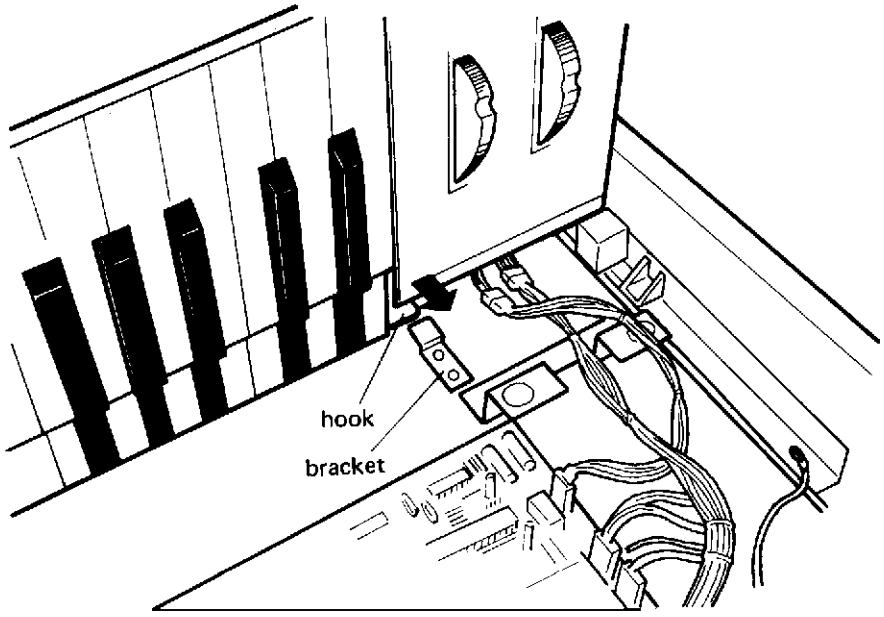
Main CPU's job
1. Receive data from sub CPU
2. EGS and OPS control
3. Load and store Voice ROM cartridge
4. LED and LCD control







Placing Upright Keyboard DX7 (DX7 鍵盤の立て方)



DX7 CIRCUIT DESCRIPTION

1. Keyboard and Panel Switch Scanning

The 4 bits BO ~ B3 from the sub-CPU (6805S) are input to the decoder (40H138).
The decoder output is sent to the keyboard transfer contacts and the panel switches.
The on or off state of the keyboard break contacts, make contacts and panel switches are sent to the sub-CPU AO ~ A7 lines via a line driver (40H240) when the sub-CPU B4 and B5 lines are low.

2. Key ON/OFF and Touch Data

The time it takes for the transfer contact to connect with the make contact after separating from the break contact is recorded by the sub-CPU timer. This value is the Touch data.
The key ON signal is generated when the transfer contact connects with the make contact, and the key OFF signal is generated when the transfer contact connects with the break contact.

3. ADC

Data entry
Pitch bend wheel
Modulation wheel
Foot controller
Breath controller
After-touch controller
Battery voltage

The 7 analog control voltages given above are fed to the ADC (M58990P-l). The analog input selected by the sub-CPU BO~ B2 bits is converted to a digital value when the sub-CPU B7 line goes low. The ADC outputs a high level to the sub-CPU C3 line when the conversion is complete. The ADC sends the 8-bit digital value to the sub-CPU when the sub-CPU B6 line goes low.

4. Data Transmission from Sub-CPU to Main CPU

4-1. When a key event occurs the sub-CPU CO line goes high, changing the state of the ready flag (R S F/F) causing the main CPU IRQ and P21 lines to go low.

4-2. The main CPU accepts one byte of data on lines AO~ A7 from the sub-CPU when the P21 line goes low.

4-3. Once this byte is accepted, pin 9 of IC24 goes low, changing the state of F/F and forcing the sub-CPU Cl line low.

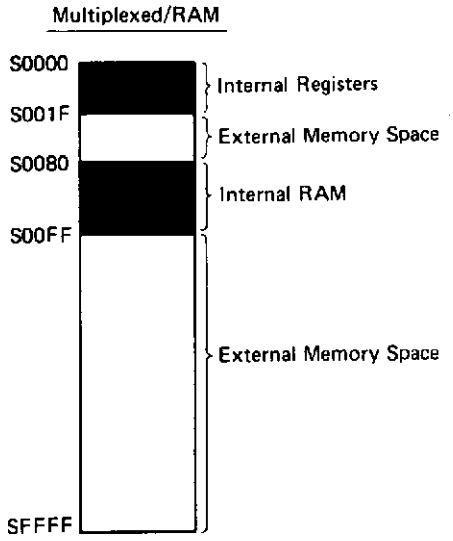
4-4. When the sub-CPU Cl line goes low, step 4-1 (above) is repeated and then in step 4-2 a second byte of data is accepted by the main CPU.

4-5. During the IRQ routine the main CPU P20 line holds C2 on the sub-CPU line low until the second byte has been transferred.

4-6. Data is not accepted from the panel switches and keys while the sub-CPU C2 line is low.

5. Main CPU Operation

The main CPU mode is set by externally initializing lines P20~P22. When “L, L, H” is applied to the P20~P22 lines and latched into the CPU on the rising edge of RES, the Extended Multiplex Mode is selected.
In this mode, P40 ~ P47 function as address lines. The lower address bits are multiplexed with the data on lines P30 ~ P37, and are separated by the address strobe signal SCI.
P20 ~ P24 and P10 ~ P17 function as I/O lines.



左図は、この動作モードの時のメモリーマップです。

The memory map for this mode is given in the diagram to the LEFT.

6. RAM

M5M511BP-15 X 8-bit CMOS RAM.

7. ROM

2764 8K X 8-bit NMOS EROM

1000 ~ 17FF	IC19	Voice Memory
1800 ~ 1FFF	IC20	
2000 ~ 27FF	IC21	Working Area

8. LED

The LED display is created via software. The LEDs are lit by data latched from the main CPU.

C000 ~ DFFF	IC14	System Program
E000 ~ EFFF	IC15	
F000 ~ FFFF		Test Program

9. LCD

Data from the main CPU is decoded and displayed at the LCD unit.

10. EGS (Envelope Generator)

8 bits of data are received from the main CPU, and envelope and frequency data are sent to OPS.

11. OPS (Operator)

The OPS uses a sine table to generate waveform data to be sent to the DAC from the received envelope and frequency data.
The OPS permits combining the 6 operators in 32 different combinations. The combinations are called “algorithms”. One of the 6 operators is able to feed the sine table output back to the input. The feedback level and algorithm data is received from the main CPU.

12. DAC

A BA9221 DAC is used. The DAC converts the digital waveform data from the OPS to an actual analog waveform. The amplitude scale factor of the analog waveform is controlled via SF0 ~ SF3. This signal is then fed to the sample & hold and low-pass filter circuits from which it is sent to the output terminal. A reference voltage is applied to pin 14 of the DAC. 8 reference voltages are generated by the muPD405 1, and the total level is externally controlled.

13. MIDI (Musical Instrument Digital Interface)

Permits data transfer with other devices. Data is received by P3 of the main CPU via a photo-coupler, and data is output from main CPU pin P24.

EGS

1 EGS Functions

Receives data from the CPU, generates envelope & frequency data, & transmits the generated data to OPS.

(see EGS block diagram)

Data received from the CPU is latched in the EGS & sent to the internal data buss.

2 EGS Rate/Level Buffer

Rate refers to the time required for the next level to be reached. For example, Rl is the time

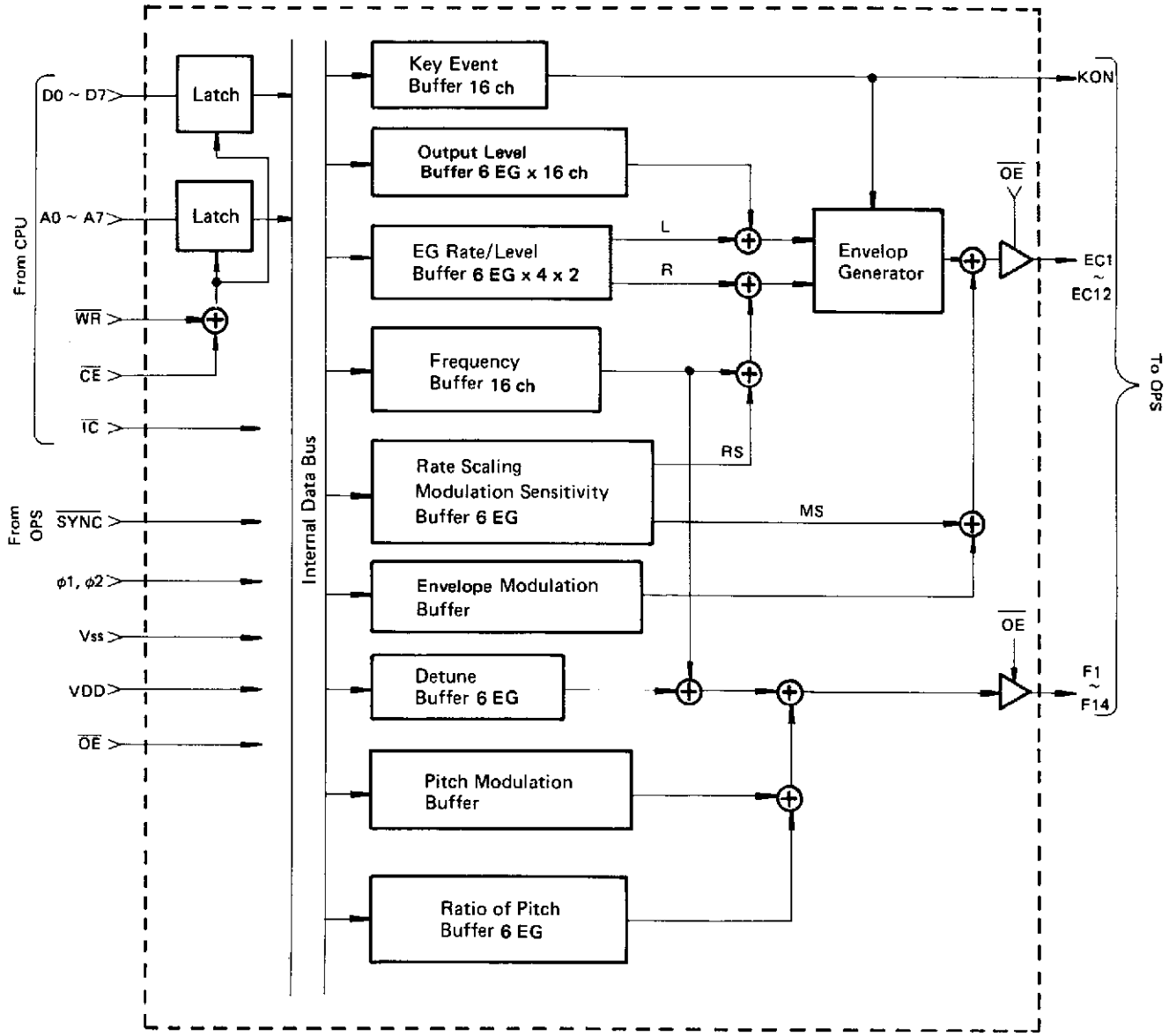
EGS

EGSの機能

CPUよりDataを受けEnvelope DataとFrequency Data を作り、OPSに送ります。

EGS functions:

Receives data from the CPU, generates envelope and frequency data, and transmits the generated data to OPS.

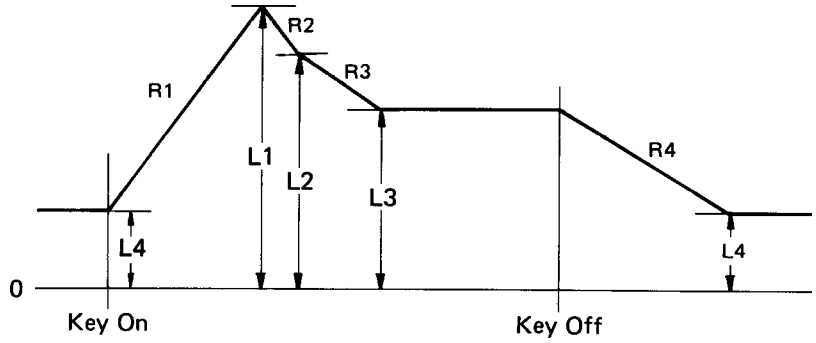


EGS Block Diagram

1．CPUより受けたDataはEGS内部のLatch にラッチされ Intenal Data Busに送られます。

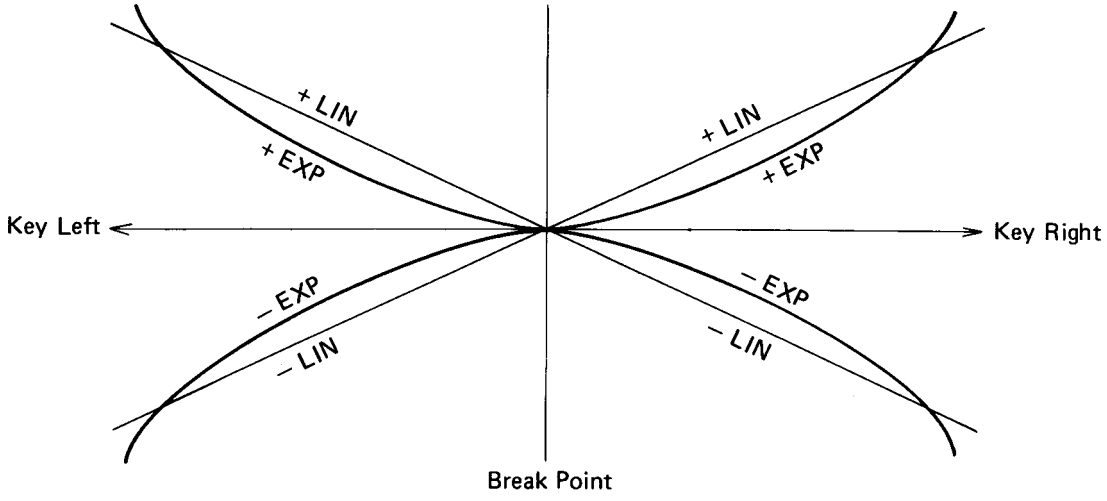
1. Data received from the CPU is latched in the EGS and sent to the internal data buss.

it takes until Ll is reached from L4. The larger the Rl value, the sharper the attack.
Rates and levels determine the basic shape of the envelope, but the actual envelope shape is affected by output level and key scaling.



3. Output Level Buffer

Output level buffer receives data concerning key scaling, after touch, and output level from the CPU. Actual values used for Ll to L4 are determined by the data stored in this buffer. Key scaling changes the output level as shown below.



LIN : Linear
EXP : Exponential

4. Frequency Buffer

Frequency data related to key code, pitch envelope, and transposition is received from the CPU and stored in the frequency buffer.

5. Rate Scaling Buffer

Data used to determine rate values according to key scaling is stored in the rate scaling buffer.

6. Values used for Rl to R4 are determined by the data stored in the frequency and rate scaling buffers.

7. Key Event Buffer

Key states (ON/OFF) are stored in the key event buffer.

8. Modulation Sensitivity Buffer

Amplitude modulation sensitivities are stored in the modulation sensitivity buffer.

9. Envelope Modulation Buffer

Modulation states of the LFO are stored in the envelope generator buffer.

10. Envelope produced by the envelope generator is modulated by the data stored in the modulation sensitivity and envelope modulation buffers to generate the final envelope.

11. Detune Buffer

Detune data according to key scaling is stored in the detune buffer.

12. Pitch Modulation Buffer

LFO pitch modulation data is stored in the pitch modulation buffer.

13. Pitch Ratio Buffer

Pitch ratios are stored in the pitch ratio buffer.

14. Data in the frequency buffer is modulated by the data stored in the detune, pitch modulation, and pitch ratio buffer to generate frequency data.

OPS

OPSの機能

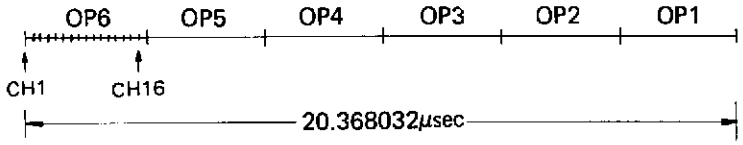
EGSよりFrequency DataとEnvelope Data, CPUよりAlgorithm DataとFeed Back Level Dataを受け、D/A ConverterにDigitalのAudio Signal Dataを送ります。

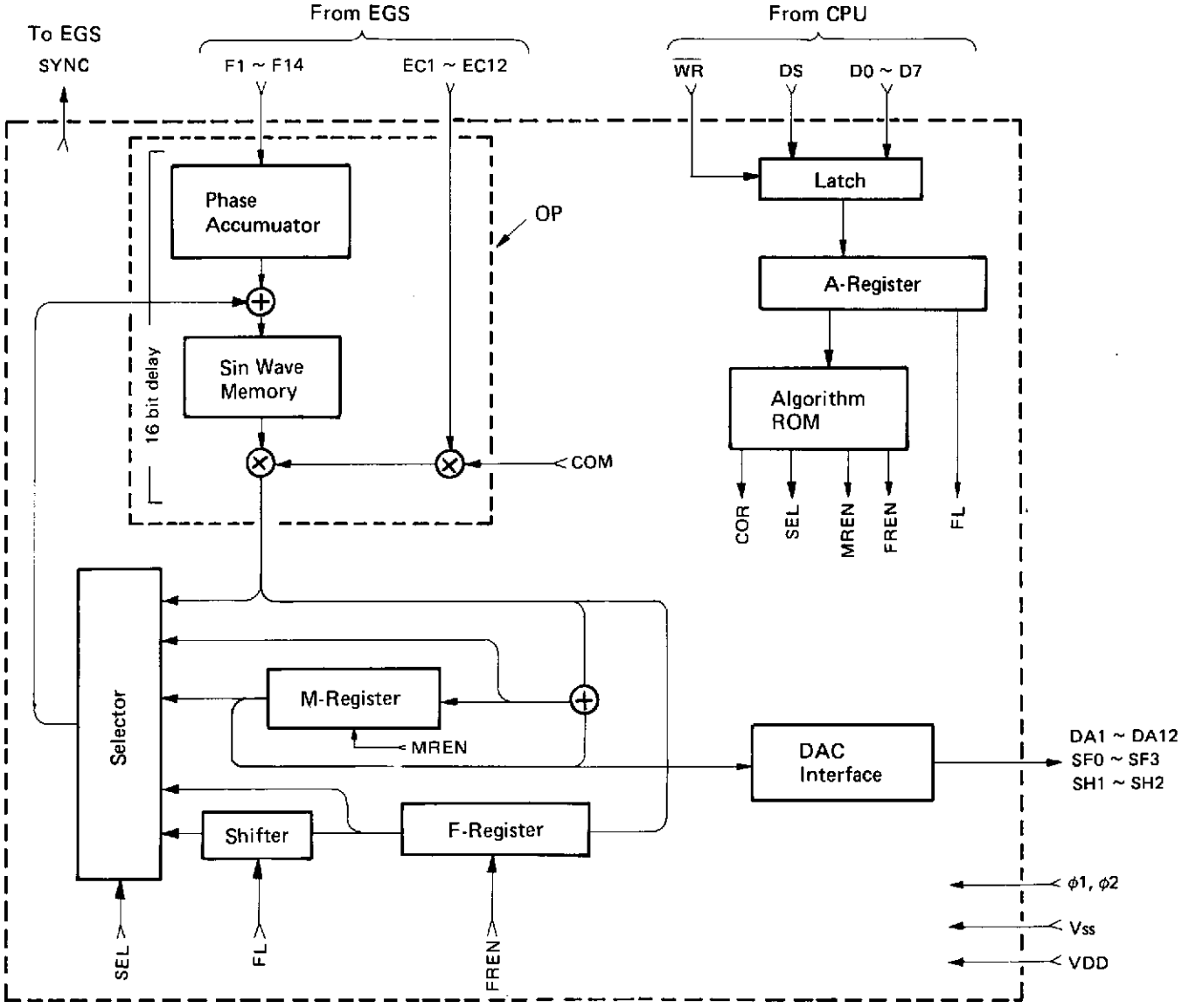
OPS functions:

Receives frequency and envelope data from EGS, and alogorithm and feedback level from the CPU, and transmits digital audio signal data to the D/A converter.

1．EGSから送られて来る、Frequency Data と Envelope Data

1. Frequency and envelope data transmitted from EGS.





OPS Block Diagram

2．OPS Block Diagramの用語説明

2. OPS block diagram description.

OP	Operator	F_1～F_{14}のFrequency Dataを、Phase AccumlatorでPhase Dataに変換し、Sin Wave Memoryを読み出します。さらにEC_1～EC_{12}のEnvelop Dataにより、Sin Wave Memoryの出力のLevelを制御します。入力に対して出力は16bit Delayします。 Converts frequency data of F1 to F14 into phase data using the phase accumulator and reads sine wave memory. Controls sine wave memory output level using envelope data of EC1 to EC12. Delays output for 16 bits after input.
COM	Compensation	Algorithmの一番下にあるOPをCarrierといいますが、COMは、このCarrierの数によりLevelを制御します。例えば、Carrierが2個の時は1/2、3個の時は1/3に各CarrierのLevelを制御します。これは、Algorithmを変えても、最終のLevelの変動が無いようにするためです。 The bottom OP in the algorithm is called a carrier. It controls level according to the number of carriers. For example, if there are two carriers, it controls each carrier level for 1/2. If there are three carriers, it controls each carrier level for 1/3. This is done to avoid changing the last level, even when the algorithm is changed.
SEL	Selection	Algorithmに従って、OPに送るModulatorの選択をします。 Selects modulators to be transmitted to the OP according the algorithm.
MREN	Memory Register Enable	M-Registerを制御します。 Controls M register
FREN	Feedback Register Enable	F-Registerを制御します。 Controls F register.
FL	Feedback Level	Feed Back Levelを制御します。 Controls feedback level.

3．OPS内部の動作は、Algorithmによって異なります。Algolithmの図とOPS Block Diagramを見ながら、いくつかの例で説明しましょう。F_1～F_{14}、EC_1～EC_{12}がOPS内部のOPで処理された結果をOPnの場合[n]と示します。
(n＝1～6)

Algorithm 1の場合

(1) 最初にOP6のDataはOPで処理され[6]となります。

(2) (1)の結果は、F-Registerにメモリーされると同時にSelectorを通ってOPを変調するための入力となります。

3. OPS internal operation depends on algorithms. Let's look at some examples using the algorithm figures and the OPS block diagram. The result, when F1 to F14 and EC1 to EC12 are processed at the OP in the OPS, is shown as [n] representing the OPn (n = 1 to 6).

When algorithm 1 is chosen:

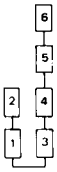
(1) First OP6 data is processed at the OP and the result is [6].

(2) The result from (1) is stored in the F register and at the same time, is input to the selector to modulate the OP.

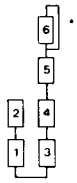
(3) OP5 data is processed at the OP by being modulated and the result is  .

(4) OP4 and OP3 data is processed at the OP in the same way as above, and the result  is stored in the M register.

(5) OP2 qnd OP1 data is processed at the OP and the result is added to the data stored in the M register. 

(6) The result in (5) is stored in M register and the result is output. 

(7) OP6 data is processed at the OP by being modulated as data in the F register, the result in (2), and the result is  .

(8) By repeating the above steps, the final output is  .

When algorithm 21 is chosen:

(1) First OP6 data is processed at the OP and the result is stored in the M register.

(2) OP5 data is processed by being modulated by the result in (1) and the result is  .

(3) The M register outputs the stored data, which is the result in (1), and stores the result in (2).

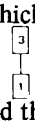
(4) OP4 data is processed by being modulated by the result in (1), and the result is 

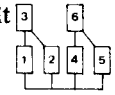
(5) Adds the results in (4) and the result in (2), which is stored in the M register, and stores the result  in the M register.

(6) OP3 data is processed at the OP and the result is stored in the F register.

(7) OP2 data is processed by being modulated by the result in (6) and the result is 

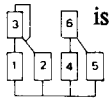
(8) Adds the result in (7) and the result in (5), which is stored in the M register, and stores the result  in the M register.

(9) OP1 data is processed at the OP being modulated by the result in (6), which is stored in the F register, and the result is  .

(10) Adds the result in (9) and the result is (8), which is stored in the M register, stores the result and outputs it. 

(11) Repeats steps from (1) to (5) and stores the result  in the M register.

(12) OP3 data is processed at the OP by being modulated by the result in (6), which is stored in the F register, and the result  is stored in the F register.

(13) Repeats steps from (7) to (10). The final result  is stored and output.

4. Examples of algorithms 1 to 21 explain the existence of only one OP in the OPS, but the OPS is multiplexed therefore it appears that there are a total of 6 operators.

OPS ALGORITHMS of DX7