HardWareMan
Dexus
Alone Coder
Shiru

TurboSound FM Programmer's Guide

(unofficial English translation by Szk@2021)

v1.11

Moscow-Ryazan, April 19, 2007

Content

1. Description

TurboSound FM (hereinafter TFM) is a sound device plugged into the socket instead of AY-3-8910 (YM2149). It has 6 regular TurboSound channels (hereinafter referred to as PSG channels, they are also SSG[1]) and 6 channels of more complex synthesized sound (FM channels). Six PSG channels are independent of six FM channels, and vice versa, so TFM has 12 channels at the same time[2].

TFM is built on two microcircuits (aka "chips") YM2203 (OPN series), compatible with AY-3-8910. Each of these microcircuits has, in addition to 16 standard registers AY-3-8910 (control of PSG channels and exchange via parallel ports), several dozen more registers that control FM synthesis and timers.

To debug programs for TFM, you can use the UnrealSpeccy v0.36 emulator.

[1] Approx. Shiru: PSG (Programmable Sound Generator) is a term used in the official documentation of General Instruments to denote the functional purpose of the AY-3-8910 (and other microcircuits from this series). SSG (Software-controlled Sound Generator) is a term used by Yamaha for the same purpose in relation to the YM2149F. For Texas Instruments SN76489, the term DCSG (Digital Complex Sound Generator) is used.

[2] If for each YM2203 microcircuit we consider the 3rd FM channel with the 7th algorithm as four independent channels that produce sinusoidal signals with the specified parameters, then 18 channels are obtained.

2. Converting Music with Sega Genesis (Mega Drive)

The Sega Genesis set-top box has a bundle of two sound chips (SN76489 and YM2612), which has approximately the same capabilities as the TFM. The main fundamental differences from TFM are as follows:

- PSG channels (SN76489) not 6, but 4 (counting the noise as a separate channel - it has a separate volume and can be used as a dirty tone), and PSG has no envelope;
- FM channels have LFO (Low Frequency Oscillator) for amplitude and frequency vibrato;
- FM channels have primitive panning (left / right / center), but not all units have a stereo output.

Almost complete music conversion from Sega Genesis to TFM is possible. The main problem is that digital drums are often used on the Sega Genesis (instead of one of the FM channels).

To convert music from Sega Genesis there is a utility gym2tfm v0.1 (by Dexus & Shiru). At the entrance, it needs an uncompressed .gym file, which is uploaded by the Megasis and Gens emulators (you can find ready-made .gym files on the Internet). This utility does not process some .gym files, including compressed ones (their distinctive feature is the size of only a few kilobytes). The utility automatically removes a pause of any length at the beginning of a dump.

The conversion result can be:

- Play with the player tfdplay.H (the simplest player by which you can understand the principles of TFM programming);
- Convert to a denser format with the utility RE_TFD v0.2 by Alone Coder (its player is called retfd02 + .H and allows you to play music many pages long of RAM).

The players have a constant pseudo60hz. When converting music from a set-top box, it must be equal to 1 in order to play music on the ZX at the same speed as it was played in the game (according to the NTSC standard).

Players are a product of collective creativity - originally written by Dexus and Shiru, then modified by Lord Vader and (mostly) Alone Coder. You can freely change the code if you want to insert the converted music into your program.

Using someone else's music in your program, indicate where this music was taken from, and (if you know) its author.

2.1. Dump format for music from Genesis emulators (*.gym)

If there is a "GYMX" header, then the initial 256 bytes with text should be skipped.

- #00 - wait for the next frame (1/60 second);
- #01 R n - write n to the register R of the 0th half of YM2612 (channels 1,2,3);
- #02 R n - write n to the R register of the 1st half of YM2612 (channels 4,5,6);

The YM2612 has a key on / off register that is common to both halves. Therefore, if a key on or key off event is read with the 3rd bit set, it must be redirected to the second TFM chip.

Ports ≥#b4 and <#30 (except #28) do not need to be translated to TFM.

YM2612 frequencies need to be adjusted when converting to TFM, because The YM2612 on the Sega Genesis is clocked at 7.67 MHz (actually sounds like 3.84 MHz).

- #03 val - write val value to PSG. Processed as follows:

```
if(val&128)
{
        chan=(val>>5)&3;
        div=(PSG.chanDiv[chan]&0xfff0)|(val&15);
        PSG.latchedChan=chan;
        PSG.latchedType=val&16;
}
else
{
        chan=PSG.latchedChan;
        div=(PSG.chanDiv[chan]&15)|((val&63)<<4);
}

if(PSG.latchedType)
{
        PSG.chanVol[chan]=(PSG.chanVol[chan]&16)|(val&15);
}
else
```

```
{
        PSG.chanDiv[chan]=div;
        if(chan==3)
{
                if(((div>>2)&1)) PSG.noiseTBits=9; else PSG.noiseTBits=1;
                PSG.noiseLFSR=0x8000;
        }
}
```

2.2. Dump format gym2tfm (*.tfm) and TFM Music Maker (*.tfd)

The file is a stream of paired bytes that define the register number and value for it, as well as a set of service markers 1 or 2 bytes long.

*.tfd header from TFM Music Maker since v1.0:

Offset	Size	Purpose
+0	4	Signature "TFMD"
+4	N	Track name, up to 64 bytes (including code 0)
+4+N	M	Author name, up to 64 bytes (including code 0)
+4+M	X	Comment, up to 384 bytes (including code 0)

All text fields are stored in ASCII encoding and end with code 0. If the field is empty, only code 0 is stored. The comment can be multi-line, line breaks are performed at #0D #0A.

Markers:

- #FF - frame start (at the beginning of playback and frame, the first chip is active);
- #FE n - wait n+3 frames - only in TFM Music Maker dumps;
- #FD - selection of the second chip;
- #FC - select the first chip (not used);
- #FB - the end of the track, or transition to a loop (placed at the end of the last frame of music);
- #FA - cycle marker (comes immediately before #FF or #FE; marker #FB should be used to jump to this place) - only in TFM Music Maker dumps.

Data:

- #00..F9 n - register number and value for this register.

2.3. Output format from converter RE_TFD v0.2 (*.tfm)

- Removed unnecessary #A6, #A5, #A4 (in which the value is equal to the old value of the corresponding register);
- #0E/F, #10/1, #12/3 - decrement/increment the high byte of frequency of PSG channel A/B/C;
- #14 - beginning of the cycle;
- #2A/B, #2C/D, #2E/F - key off/offon channel A/B/C;
- #A4 xx yy - #A4=xx, #A0=yy;
- #A5 xx yy - #A5=xx, #A1=yy;
- #A6 xx yy - #A6=xx, #A2=yy;
- #Cx, #Cx+8, #Dx - low byte of the frequency of channel A/B/C += x-4;
- #Dx+8, #Ex, #Ex+8 - low byte of the frequency of PSG channel A/B/C += x-4;
- #Fx, #Fx+4, #Fx+8 - volume of PSG channel A/B/C += x-2;
- #FC - end of the cycle (placed at the end of the last frame of the music);
- #FD - selection of the second chip;
- #FE n - wait for n+3 frames;
- #FF - frame start (at the beginning of playback and frame, the first chip is active);
- Other codes R n - writing n value to register R.

3. TFM Music Creation

Use TFM Music Maker by Shiru to write your own compositions. The editor currently only supports FM channels and there is no separate setting of frequencies for all operators of 3 YM2203 channels, which does not prevent TFM Music Maker from being a powerful tool for creating music.

The editor unloads music either in the internal .tfe format (which is not intended for playback on the ZX), or as a register dump with the .tfd extension.

This register dump can be:

- Play with tfdplay.H player (constant pseudo60hz must be equal to 0);
- Convert to a denser format with the utility RE_TFD v0.2 by Alone Coder (its player is called retfd02+.H and allows you to play music many pages long of RAM).
- Compile with the TFM Compiler v1.2 by Alone Coder utility (its player is called tfmcom12.H, it is the fastest of the listed).

The tfmcom12.H player calculates the minimum, maximum and average execution time. These are the 1st, 2nd and 3rd displayed numbers, respectively. Clock counting is set to Pentagon in slow mode. Typical peak time is 13000 ticks, typical average time is 1500 ticks.

To reduce the peak time, do not initialize all instruments at once, it is better to make a separate pattern before starting the song, where all initializations take place in turn. Thus, if your module does not constantly switch instruments, you can get a peak time of less than 7000 bars, which does not exceed the peak playing time of six PSG channels by players from Pro Tracker 3.x.

3.1. Compiled module format TFM Compiler v1.1 (*.tfc)

- 6 characters "TFMcom" and 3 characters of the version number (eg "1.0")
- playback speed in Hz (50 or 60)
- offset of channel A (16-bit, least significant byte forward)
- ...
- offset of channel F
- reserve 12 bytes for offsets for PSG channels (filled with zeros)
- ASCIIZ track name (max. 64 bytes, but the user program may not output more than 32 characters)
- ASCIIZ author name (similar)
- ASCIIZ comment (max. 384 bytes), can contain CR+LF codes (user program can break lines longer than 32 characters and not output more than 12 lines, including those added at break)
- channel A
- ...
- channel F

Channel data is encoded frame by frame, with a control byte placed at the beginning of the frame:

- `%11111111,-disp8 = use old frame data`
- `%111ttttt = skip 32..2 frames`
- `%110ddddd = slide d+16`
- `%11010000,frames,-disp16 = repeat block (skips are used as 1 frame)`
- `%10111111,-disp16 = use old frame data`
- `%10NNNNNf = keyoff,[freq,]0..30 regs, keyon`
- `%01111111 = end`
- `%01111110 = begin`
- `%01NNNNNf = keyoff,[freq,]0..31 regs`
- `%00NNNNNf =        [freq,]0..30 regs`

Where:

- f - sign of the presence of freq. If f=1, then the value of the register #A4+ is read from the stream, then the value of the register #A0+;
- -disp8 - negative 8-bit offset;

- -disp16 - negative 16-bit offset, stored in the most significant byte forward;
- NNNNN - the number of registers to be filled in (not counting those filled in the previous paragraph). Register format: register number (1 byte), register value (1 byte).

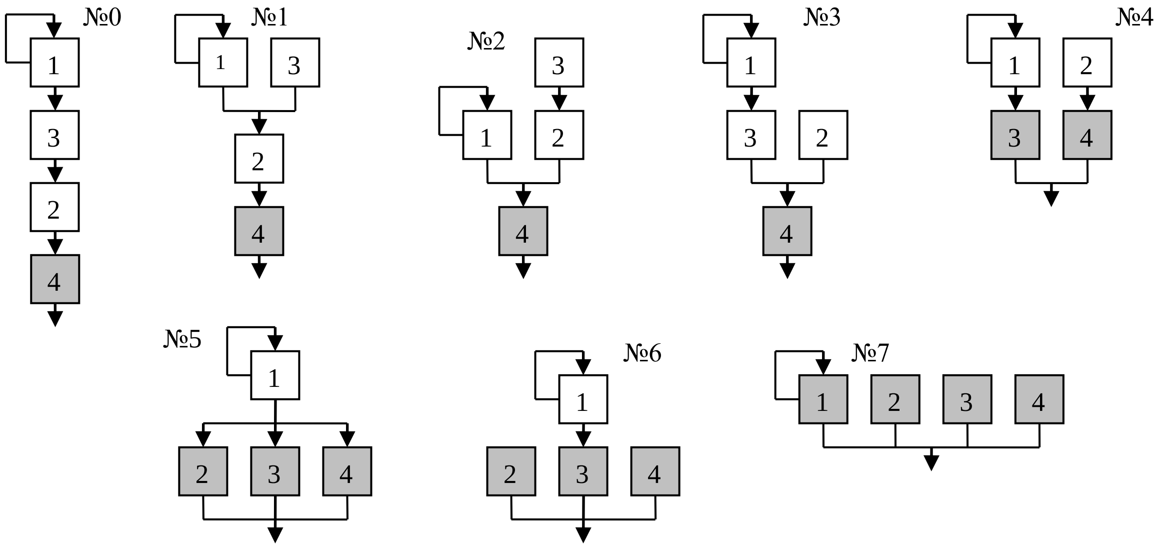
4. Description of the YM2203 chip

YM2203 contains:

- SSG, completely similar to YM2149 (3 channels + noise generator + envelope generator), with analog output;
- 3 channels of FM synthesis, the signal of which is summed and sent to an external DAC in a 13-bit serial code;
- 2 timers for use by the program. In TFM, an interrupt is not generated on these timers.

4.1. Synthesis principle

Each FM channel is assigned a fundamental frequency (tone) and data for converting this frequency into a complex signal. For this, the channel has 4 sinusoidal signal generators (called "**operators**"), which are connected to one of eight circuits ("**algorithms**"), where some generators modulate others. Below are the operator connection diagrams.



Correspondence of algorithms and received sounds:
0. Guitar with "distortion", cymbals, bass;
1. Harp, meander;
2. Bass, electric guitar, wind instruments, piano, woodwind;
3. Stringed instruments, acoustic guitar, bells;
4. Flute, bells, chorus, bass drum, snare drum, tom;
5. Wind instruments, organ;
6. Xylophone, volume, organ, vibraphone, snare drum, bass drum;
7. Large organ.

4.2. Channel operators

Each operator has its own amplitude envelope, as well as its own frequency, which depends on the fundamental frequency of the channel. The operator's frequency, with an accuracy of the so-called "detuning", is a multiple of the fundamental frequency, or equal to 1/2 of the fundamental frequency[3].

The operator's phase is modulated by a signal at its input. The signal at the output of a particular operator is described by the formula:

$$F = A \sin (\omega Ct + I \sin \omega Mt),$$

Where:

[3] Channel 3 can use completely independent frequencies for each operator, even if they are not multiples of the fundamental (not harmonics). Suitable for synthesizing drums and various chords (however, major and minor triads can be played on other channels as well).

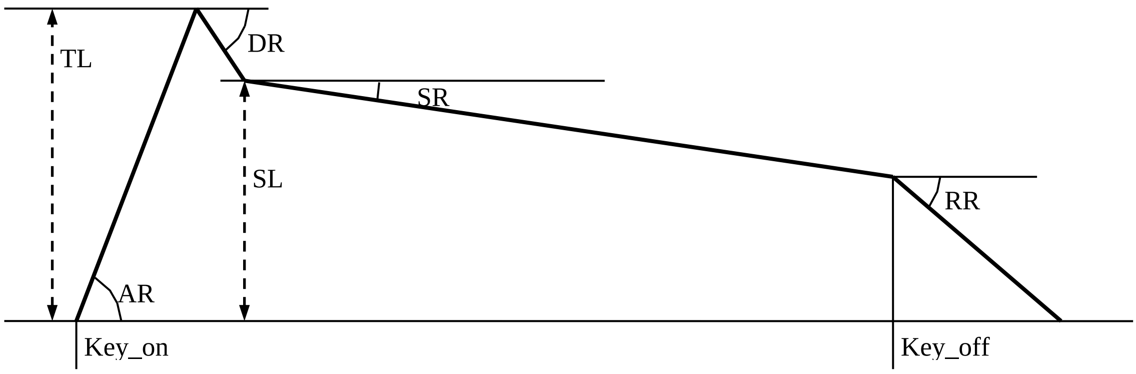
- A - amplitude (operator signal level);
- I - modulation factor (signal amplitude at the operator's input);
- ωC - angular frequency (carrier) of the operator itself;
- ωM - angular frequency (modulating) at the operator input. Operator 1 of each channel has feedback. The signal at the output of the first operator is described by the formula:

$$F = A \sin(\omega Ct + \beta F),$$

where β is the feedback coefficient.

Some operators modulate another operator with their output, others (called "**slots**" - they are shaded in gray on the algorithm diagrams) go to the final output, where they are summed up. Slots can be used as independent channels, but, as a rule, they are all turned on at the same time, and therefore, separate sounding does not occur.

4.3. Amplitude envelope operator



The signal begins to develop after pressing a key ("**key on**" event). *With "**key on**" the signal level is not reset, the development continues from the current level.*

The signal has an "attack" (Attack), a strong main decay (Decay), then a slow secondary decay (Sustain, aka Decay 2). After releasing the key (event "**key off**"), the signal is abruptly damped (Release) - like a piano: after releasing the key, the damper falls on the strings and muffles them.

The amplitude envelope of the signal is controlled by the amplitude TL, the level of the beginning of the retention SL (aka T1L), angles AR, DR (aka D1R), SR (aka D2R) and RR, as well as several additional registers. Values in the diagram:

TL	(Total Level) Total amplitude, highest signal level.
AR	(Attack Rate) Attack rate, sets the rate of the initial growth of the signal level. If the attack is slow enough, a key off event can occur before the signal level reaches the TL value.
DR	(Decay rate) Decay rate after growth.
SL	(Sustain level) The level (relative to TL) at which the decay starts when the key is held down.
SR	(Sustain rate) The decay rate when the key is held down. Will continue until the "**key off**" event.
RR	(Release rate) Final decay rate after "**key off**" event.

Additional registers affecting the envelope:

RS	Rate Scaling. A factor that indicates how much shorter all envelope elements become at higher frequencies. For example, on a grand piano, high notes decay faster than low notes.
SSG-EG	Register indicating the sequence of envelope stages (loop, flip, etc.). Acts by analogy with the SSG envelope.

4.4. YM2203 Registers

Main system registers:

- Control of the microcircuit frequency prescaler;
- Timer management;
- Separate on-off ("**key on**" and "**key off**") operators of FM channels. There is an independent on / off control of each of the 12 operators;
- Channel 3 mode. Usually FM channel has one fundamental frequency and multipliers for operators, but in extended mode 4 independent frequencies can be selected - one for each operator.

Channel control registers:

- The value of the channel frequency (in normal mode) or the main frequency for operator 4 (in the extended mode of the 3rd channel);
- The feedback factor for the operator is 1;
- Number of operators' communication algorithm.

The rest of the registers are managed by separate operators. Operator relationships are determined by the selected algorithm, but the envelope is always set independently for each operator. In the case of the 3rd FM channel, the fundamental frequency can also be set independently for each operator.

NOTE: before writing new data, it is necessary to read a byte from any register of the YM2203. Bit 7 indicates that the chip is busy. It is necessary to wait until it becomes equal to 0. (See section 5.1.)

D7							D0
Bus	X	X	X	X	X	OvA	OvB

Bus: 1=the chip is busy, 0=the chip is free;
OvA/OvB: 1=timer has finished counting and overflowed.

NOTE: in the case of writing double registers, for example, a 10-bit timer register or a 14-bit frequency register, you should write the upper half first.

	D7	D6	D5	D4	D3	D2	D1	D0
24H	Low Byte of Timer A							
25H							Art. Timer A bits	
26H	Timer B							
27H	Channel 3 Mode		Reset B	Reset A	Enable B	Enable A	Boot B	Boot A
28H	State of operators (key on/off)					Channel		
2D-2FH	Chip prescaler control							
	Channel management							
30H+		DT1 (Detune)			MUL (Multiple)			
40H+		TL (Total level)						
50H+	RS (Rate scaling)			AR (Attack rate)				
60H+				DR (Decay rate)				
70H+				SR (Sustain rate)				
80H+	SL (Sustain level)				RR (Release rate)			
90H+					SSG-EG			
A0H+	Least significant byte of frequency							
A4H+			Block (octave)			Most significant bits of frequency		
A8H+	Low byte of channel 3 operator frequency							
ACH+			Channel 3 operator block (octave)			Most significant bits of channel 3 op. frequency		
B0H+			Self-Feedback (Feedback)			Connection (Algorithm)		

The "+" next to the register number means a range. For registers 30H+ to 90H+, the range has 16 values. Of these, only 12 are used (3 channels with 4 operators each). The formation of the register number for the 30H+ range can be seen from the table:

R7	R6	R5	R4	R3	R2	R1	R0
0	0	1	1	Operator		Channel	

30H	Channel 1, Operator 1
31H	Channel 2, Operator 1
32H	Channel 3, Operator 1
33H	
34H	Channel 1, Operator 2
35H	Channel 2, Operator 2
36H	Channel 3, Operator 2
37H	
38H	Channel 1, Operator 3
39H	Channel 2, Operator 3
3AH	Channel 3, Operator 3
3BH	
3CH	Channel 1, Operator 4
3DH	Channel 2, Operator 4
3EH	Channel 3, Operator 4
3FH	

Values of "Operator" and "Channel" fields:

N	D3	D2	Operator	D1	D0	Channel
0	0	0	1	0	0	1
1	0	1	2	0	1	2
2	1	0	3	1	0	3
3	1	1	4	1	1	unacceptable

On the right, in the form of a table, an illustrative example of the formation of the final register address is given. Gray boxes are not used. Formation of the address for the ranges from 40H+ to 90H+ is similar.

In the ranges from A0H+ to B0H+, only 3 register numbers out of every 4. For A0H+ these are numbers:

R7	R6	R5	R4	R3	R2	R1	R0
1	0	1	0	0	0	Channel	

A0H	Channel 1, l
A1H	Channel 2, l
A2H	Channel 3, l
A3H	
A4H	Channel 1, h
A5H	Channel 2, h
A6H	Channel 3, h
A7H	

The exceptions are the ranges A8H+ and ACH+:

A8H	Channel 3, operator 2, l
A9H	Channel 3, operator 1, l
AAH	Channel 3, operator 3, l
ABH	
ACH	Channel 3, operator 2, h
ADH	Channel 3, operator 1, h
AEH	Channel 3, operator 3, h
AFH	

4.5. Description of registers

Registers 24H and 25H – Timer A

Register	D7	D6	D5	D4	D3	D2	D1	D0
24H	Least significant byte							
25H	Not used						Most sign. bits	

Registers 24H and 25H are combined into a 10-bit timer A. The write order should be as follows: first write the number to register 24H, then to register 25H (or is it rather the other way around?). The response period is calculated using the formula:

18 * (1024 - Timer A) microseconds

Timer A - all 1 → 18 μs = 0.018 ms
Timer A - all 0 → 18400 μs = 18.4 ms

Register 26H – Timer B

D7	D6	D5	D4	D3	D2	D1	D0
Timer B							

8-bit timer B. Formula:

288 * (256 - Timer B) microseconds

Timer B = all 1 → 0.288 ms
Timer B = all 0 → 73.44 ms

Register 27H

D7	D6	D5	D4	D3	D2	D1	D0
CSM MODE[4]	Channel 3 mode	Reset B	Reset A	Turn on A	Turn on B	Boot A	Boot B

Register 27H controls Timers A and B and FM Channel 3 mode.

Channel 3 mode	D6	
Normal	0	Channel 3 is the same as the rest
Special	1	Channel 3 has 4 different frequencies

In normal mode, channel operators use frequencies that are multiples of operator 1.
In special mode, each operator has its own frequency. Operator frequency 4 of channel 3 is in registers A2H and A6H. The frequencies of operators 2, 1 and 3 of channel 3 are in registers A8H and ACH, A9H and ADH, AAH and AEH, respectively.

Loading	1 starts the timer, 0 stops it.
Turning on	1 allows the flag to be set on overflow. The timer counts down to 0 without setting the flag.
Reset	Entry 1 clears the flag, entry 0 does nothing.

[4] If here 1, then on timer A overflow, key on of all operators of channel 3 is performed.

Register 28H – Note on/off

<table>
<tr><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td colspan="4">Operator</td><td rowspan="2">Not used</td><td colspan="3" rowspan="2">Channel number</td></tr>
<tr><td>4</td><td>2</td><td>3</td><td>1</td></tr>
</table>

The register is used to generate the "**key on**" and "**key off**" events. "**Key on**" is an analogue of pressing a synthesizer key. "**Key off**" is analogous to releasing a key. The sequence of extracting a note: 1 - set the parameters, 2 - “**key on**”, 3 - wait, 4 - “**key off**”. When “**key off**” occurs, the operator stops the slow decay and starts the fast decay, the rate of which is set in “RR” (Release Rate).

With a single write to register 28H, the state of all channel operators is set simultaneously. Usually they are set to either all to 1 (on) or all to 0 (off). Thanks to the channel 3 feature, you can use separate notes for each operator, turning them on and off separately.

The microcircuit responds directly to the change of the key on / off bits, so you cannot make two key ons in a row for the same operator - there must be a key off between them.

The channel number is encoded as follows:

D2	D1	D0	Channel
0	0	0	1
0	0	1	2
0	1	0	3

Registers 2D-2FH – Chip Prescaler Control

See section 5.3.

Register Range 30H+ – Detune and Multiplier

<table>
<tr><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>X</td><td colspan="3">DT1</td><td colspan="4">MUL</td></tr>
</table>

DT1 (Detune) and MUL (Multiple) set the operator frequency in relation to the fundamental frequency in the channel.

The MUL value lies in the range 0..15 and is a multiplier of the fundamental frequency. An exception is the value 0 - it corresponds to 1/2 of the fundamental frequency. So MUL=0..15 gives *1/2, *1, *2, ... *15.

DT1 gives a slight variation (detuning) of the fundamental *MUL. The most significant bit of DT1 (bit D6 in 30H+) is the sign and the other two are the value.

D6	D5	D4	Factor
0	0	0	without changes
0	0	1	×(1+E)
0	1	0	×(1+2E)
0	1	1	×(1+3E)
1	0	0	without changes
1	0	1	×(1-E)
1	1	0	×(1-2E)
1	1	1	×(1-3E)

Where E is a small number.

Register Range 40H+ – Basic amplitude

<table>
<tr><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>X</td><td colspan="7">TL</td></tr>
</table>

TL (Total Level) sets the maximum amplitude of the operator's envelope, with 0 - the maximum amplitude, 127 - the smallest. The change occurs in approximately 0.75 dB steps.

To control the volume of a note, change the TL on slots (output operators) only. Changing the TL of other operators will change the timbre.

Register Range 50H+ – Speed scale and attack speed

D7	D6	D5	D4	D3	D2	D1	D0
RS		X	AR				

Registers 50H+ contain RS (Rate Scaling) and AR (Attack Rate). AR is the initial attack speed, which continues up to the maximum TL level (see above). *With AR=31 the attack is the fastest, with AR=1 - the slowest, with AR=0 there is no movement (similar to other speeds). With an increase in AR (and other speeds) by 2 units, the speed doubles.*

RS is the coefficient of acceleration of the amplitude envelope depending on the frequency of the sound (the higher the frequency, the more it accelerates), and affects all speeds (AR, DR, SR and RR) in the same way.

The five most significant bits of the frequency (3 octave bits and 2 note bits) are called KC (Key code) and are used to calculate the total RS using the formulas:
RS=0 → Final speed = 2 * Speed + (KC/8)
RS=1 → Final speed = 2 * Speed + (KC/4)
RS=2 → Final speed = 2 * Speed + (KC/2)
RS=3 → Final speed = 2 * Speed + (KC/1)
key scale rate=kcode>>(3-KSR)
KC/N is always rounded down.

Since all speeds vary within 0..31, the range of RS influence on the speed varies from small (0-3) to large (0-31).

Register Range 60H+ – First decay rate

D7	D6	D5	D4	D3	D2	D1	D0
X			DR				

DR is the speed of the first decay (see envelope figure). Like all other speeds, it depends on the RS.

Register Range 70H+ – Hold decay rate

D7	D6	D5	D4	D3	D2	D1	D0
X			SR				

SR is the speed of the second (slow) decay, continuing for as long as the key is pressed.

Register Range 80H+ – Hold Amplitude and Release Speed

D7	D6	D5	D4	D3	D2	D1	D0
SL				RR			

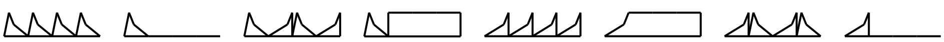
SL - secondary amplitude, after which the first attenuation turns into the second. The larger the number, the smaller the amplitude (0 - maximum).

RR - release rate (more precisely, fast decay after the key is released). All speeds are 5-bit numbers, but only 4 bits are available here. The 4 most significant bits of the release rate are used here to match, and the least significant bit is always 1. In other words, multiply by 2 and add 1.

Register Range 90H+ – Envelope Direction/Loop Option

D7	D6	D5	D4	D3	D2	D1	D0
X				SSG-EG			

The register controls the shape, direction and looping of the envelope. If bit D3 is 0, then the register is considered inactive. The values are interpreted as follows (*all times are reduced by a factor of 4, and the flip is carried out along the volume axis, and not along the time axis*):



Type 8 Type 9 Type A Type B Type C Type D Type E Type F

Register range A0H+, A4H+, A8H+, ACH+ – Frequency and octave

	D7	D6	D5	D4	D3	D2	D1	D0
A0H+	Least significant byte of frequency							
A4H+	X		Block (octave)			Most significant bits of frequency		
A8H+	Low byte of channel 3 operator frequency							
ACH+	X		Channel 3 operator block (octave)			Channel 3 operator frequency most significant bits		

Channel 1 frequency - in registers A0H and A4H.
Channel 2 frequency - in registers A1H and A5H.
Channel 3 frequency in normal mode (see above) - in registers A2H and A6H.

If channel 3 is in special mode:
Channel 3 operator frequency 4 - in registers A2H and A6H
Channel 3 operator frequency 2 - in registers A8H and ACH
Frequency 1 operator channel 3 - in registers A9H and ADH
Frequency 3 operator channel 3 - in registers AAH and AEH

Frequency is a 14-bit number. It is necessary to set first the most significant, then the least significant byte (for example, first the A4H register, then the A0H register). The most significant 3 bits are called "**block**" and represent the octave. The next 11 bits set the position in the octave, and a sequence of 12 values is possible:

Frequency value		Frequency (indicated for the 4th block at a frequency of 4MHz microcircuit)	
Decimal	Hexadecimal		
617	269	261.7 Hz	Do
653	28D	277.2 Hz	Do sharp/Re flat
692	2B4	293.7 Hz	Re
733	2DD	311.1 Hz	Re sharp/Ми flat
777	309	329.6 Hz	Mi
823	337	349.2 Hz	Fa
872	368	370.0 Hz	Fa sharp/Sol flat
924	39C	392.0 Hz	Sol
979	3D3	415.3 Hz	Sol sharp/La flat
1037	40D	440.0 Hz	La
1099	44B	466.2 Hz	La sharp/Si flat
1164	48C	493.9 Hz	Si

These numbers (they can be recalculated for a different scale) are used in each octave.
Apparently, the frequency formula is as follows:

$$F = N \times \frac{Clk \times 2^{Block}}{Scale \times 2^{21}}$$

Where:

- *F* – received frequency;
- *N* – frequency value.
- *Block* – block number (octave), from 0 to 7;
- *Clk* – microcircuit frequency;
- *Scale* – frequency division coefficient of the microcircuit. The default (after reset) is 6*12 (72), the values 2*12 (24) and 3*12 (36) are also possible - after switching the prescaler (registers 2DH/2EH/2FH).

If you smoothly change the frequency while playing a note, you can get vibrato and portamento effects.

Register range B0H+ – Feedback and algorithm

D7	D6	D5	D4	D3	D2	D1	D0
X		Self-Feedback			Connection (Algorithm)		

Self-Feedback is the degree of return of the 1st operator's own signal back to it.
Algorithm is the way operators interact. See above for the algorithm diagrams.

4.6. Test program

Test program for initializing the "Grand Piano" note:

Register	Value	Note
27H	0	Normal Channel 3 Mode
28H 28H 28H	0 1 2	Turn off all channels
30H 34H 38H 3CH	71H 0DH 33H 01H	DT1 and MUL
40H 44H 48H 4CH	23H 2DH 26H 00H	TL
50H 54H 58H 5CH	5FH 99H 5FH 94H	RS and AR
60H 64H 68H 6CH	5 5 5 7	DR
70H 74H 78H 7CH	2 2 2 2	SR
80H 84H 88H 8CH	11H 11H 11H A6H	SL/RR
90H 94H 98H 9CH	0 0 0 0	Zero SSG-EG
B0H	32H	Feedback and algorithm
28H	00H	Key off
A4H	22H	Install art. frequency byte
A0H	69H	Install ml. frequency byte
28H	F0H	Key on
<Wait>		
28H	00H	Key off

Note: we always write the high byte first, then the low one.

5. The subtleties of TFM programming

5.1. Working with registers

TFM (even PSG registers) cannot write data too quickly. Before and after specifying the register number, expect the YM2203 to be ready. The ready flag is read from the 7th bit of any YM2203 register (i.e. from the #fffd port) if the ready read mode is enabled. This mode is enabled if the last selected TFM pseudo-register was number %1111100c, where c is the chip number. All registers of the form %11111frc are pseudo-TFM registers, where f turns on FM synthesis (0=on), r turns on the readiness polling mode (0=on), c is the microcircuit number. The choice of the pseudo-register is processed by the FPGA, it does not reach the YM2203 - the current register does not change.
If FM synthesis is off (f=1), then TFM can be written at any speed.

Working with TFM is as follows:

- Select register %11111001, work with one microcircuit;
- Select register %11111000, work with another microcircuit;
- After finishing work with FM (end of music), select register %11111111.

Working with a specific chip, provided that the pseudo-register %1111100c is selected:
1. Read the status bit (7th in port #fffd) and wait for readiness (=0);
2. We put the register number;
3. Read the status bit (7th in port #fffd) and wait for readiness (=0);
4. We write to the register, the number of which is set in paragraph 2.

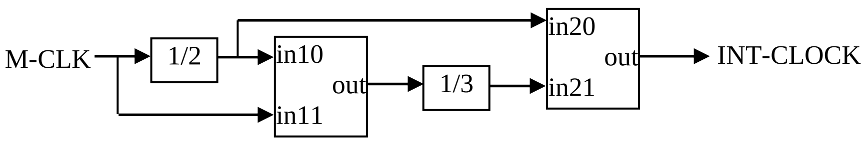
5.2. FM synthesis features

Changes to the instrument parameters (except for TL, MUL and, apparently, Detune) are processed only after the next key on. Therefore, the sequence "keyoff, keyon, change" does not work (more precisely, the change will occur on the next note). This allows you to fill in the instrument parameters per frame (frame) before the start of the note, as in TFM Compiler.
During key on and at the beginning of a new envelope period, the sine wave generator is initialized, which gives a more predictable sound than on the PSG.
You cannot write only the upper or only the lower frequency register - you must fill in both (higher, then lower) each time. Otherwise, the frequency is not what you expected.

5.3. TFM Initialization Feature

During initialization (after zeroing the registers), write any number to register #2F, then any number to register #2D. Otherwise, the music is overpriced. These are the prescaler registers - they control the frequency prescaler. MAME describes their work as follows:

```
prescaler circuit (best guess to verified chip behaviour)
```




```
reg.2d : sel2 = in21 (select sel2)
reg.2e : sel1 = in11 (select sel1)
reg.2f : sel1 = in10 , sel2 = in20 (clear selector)
reset  : sel1 = in11 , sel2 = in21 (clear both)
```

* * *

Send your comments and advice to: dmitry.alonecoder@gmail.com