APPLE-1 SID INTERFACE

by Claudio Parmigiani

PURPOSE

Allow the Apple-1 computer to generate sounds and simple melodies via BASIC or Assembly/machine code.

IMPLEMENTATION

A SID (Sound Interface Device, MOS 6581) has been used.

It is addressed through a TTL logic, which will be explained later. Some circuitry has been added to ensure compatibility with the MOS 8580. An amplifier has also been added in order to provide an adequate output level.

CIRCUIT

1) POWER
   - 5V is taken directly from the Expansion Bus
   - 12V (for 6581) or 9V (for 8580) is derived from the "12V Unregulated" on the Expansion Bus. Raw voltage is about ~20V (not stabilized). A jumper is used to select the right voltage.
2) FILTERS
   - Typical 6581's CAP1/CAP2 filter capacitors value is 470pF. These capacitors are present on the board. It is possible to select other capacitors by switching another couple of jumpers. Alternative capacitors must be connected to the appropriate sockets.
3) AUDIO AMPLIFIER
   - A simple 1-transistor (2N2222) audio amplifier is used. Output is connected to a stereo 3.5mm jack female connector.
4) MISCELLANEOUS
   - POTX/POTY SID's pin are connected to auxiliary connector, for future use.

ADDRESSING

Connections to `/RES`, `R/W` and φ2 signals are *by the book*, just like Data Bus (D0 to D7) and Address Bus (A0 to A4) signals.

Chip Select (`/CS`) signal is derived from a TTL 74LS10 logic port, which provide a logic LOW signal when:
   - Address bus is `$C8xx`
   - φ2 is at logic HIGH level.

`$C8xx` address is resolved as follows:

HEX:	C				8				0				0			
ADDR LINE:	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BIN:	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	↓				↓				↓				↓			
ACTION:	pick "R" LOW signal				pick A11 HIGH signal				does not care				does not care			

"R" signal is usually hard-wired on the CHIP SELECT section of the Mainboard of the Apple-1. It is usually connected to the four kBytes identified by $Cxxx addresses (R-C wire), in order to allow the *Cassette Interface* to be addressed properly at $C1xx.

Therefore, it is not necessary to analyse all the four most significant bits (1100) to gather the "**C**" part of the address: it will be enough to pick up the LOW state of "R" line.

To pick up the "8" part of the address: A11 HIGH state will be taken into account.

IC 74LS10 performs the necessary logic operations mentioned above, plus the φ2 clock.

According to the datasheet, the 6581's access time is typically 340 ns.

At 1 MHz φ2 clock the IC should provide a /CS signal of about 500 ns, which fits perfectly with 6581's specifications.

ADDRESSING CONSIDERATIONS

The *Cassette Interface* is hard-coded to be addressed between $C000 and $C1FF (512 memory locations, but only 256 bytes used for PROM addresses, other addresses are either "duplicates" or dedicated to output).

The *Apple-1 SID Interface* is hard-coded to be addressed between $C800 and $CFFF (2048 memory locations). This means that the SID's 29 registers will be "duplicated" many times at different addresses.

This is not a big deal: originally all the four kBytes of the $Cxxx address block would have to be occupied by the cassette interface only.

The split of the 4096 addresses in two distinct parts does not cause bus collisions, is harmless, and it does not needs CHIP SELECT re-wiring on original Apple-1s.

Due to the address mapping explained above the first register of the SID will start at decimal address 51200. Apple-1's "Integer BASIC" handles numbers with the *16-bit Signed Integer* format.

51200 is greater than 32767 (the largest acceptable number) therefore, it will "roll over" and become a negative number.

To *POKE* a specific register just use the negative value. For example, POKE -14312,15 will raise the volume to the maximum.

The same applies to *PEEKs* as well: PRINT PEEK(-14311) will show the content of the POTX register.

Of course, there are no such things like "negative memory locations"; it is just a matter of notation.

REGISTERS MAPPING

For convenience, here is a table that identifies all the 29 registers in different notations:

SID REGISTER	HEX ADDR	DEC ADDR	INTEGER BASIC ADDR
0	C800	51200	-14336
1	C801	51201	-14335
2	C802	51202	-14334
3	C803	51203	-14333
4	C804	51204	-14332
5	C805	51205	-14331

6	C806	51206	-14330
7	C807	51207	-14329
8	C808	51208	-14328
9	C809	51209	-14327
10	C80A	51210	-14326
11	C80B	51211	-14325
12	C80C	51212	-14324
13	C80D	51213	-14323
14	C80E	51214	-14322
15	C80F	51215	-14321
16	C810	51216	-14320
17	C811	51217	-14319
18	C812	51218	-14318
19	C813	51219	-14317
20	C814	51220	-14316
21	C815	51221	-14315
22	C816	51222	-14314
23	C817	51223	-14313
24	C818	51224	-14312
25	C819	51225	-14311
26	C81A	51226	-14310
27	C81B	51227	-14309
28	C81C	51228	-14308

INTEGER BASIC TEST PROGRAM

The following program has been adapted to Integer BASIC from BASIC V2 in order to match the new SID initial address and the one-character variable's name.

```
10 S=-14336
15 L=S:H=S+1:W=S+4:A=S+5:I=S+6:J=S+24
20 POKE J,15
25 POKE A,16+9
30 POKE I,4*16+4
35 POKE H,29
40 POKE L,69
45 POKE W,17
50 FOR T=1 TO 1000:NEXT T
55 POKE W,0:POKE A,0: POKE H,0
60 END
```

The execution of the program above will produce a single tone for a few seconds.

SIMILAR PROJECTS

Back in 2014 Sampo Peltonen worked on a similar project [http://sampopeltonen.github.io/Vicious/] using a Replica-1 instead of a Replica (or original). Replica-1 and Replicas rely on different hardware. However, Sampo's prototype could have worked on a Replica (or even original).

The goal of his project was to play SID tunes. Sampo included also a CIA and an EEPROM to store the player's code. CIA's purpose was to generate IRQs in order to allow the software to play the tunes properly. Nevertheless, the compatibility with the tunes was limited due to lack of IRQs.

I contacted Sampo to have informations about the status of his project, which seems to be discontinued. I am quite confident in stating that we both agree that my project is totally different in approach, purpose and implementation.

BIBLIOGRAPHY

6581 & 74LS10 Datasheet
http://archive.6502.org/datasheets/mos_6581_sid.pdf
http://ecee.colorado.edu/~mcclurel/sn74ls10rev5.pdf

Apple-1 Operations manual & BASIC
https://www.applefritter.com/files/a1man.pdf
https://www.applefritter.com/files/basicman.pdf

Apple-1 Cassette Interface:
https://www.sbprojects.net/projects/apple1/aci.php

For AF Amplifier & Filter capacitor:
https://www.commodore.ca/manuals/funet/cbm/schematics/computers/c64/252312-right.gif

Sound test program:
ftp://ftp.zimmers.net/pub/cbm/c64/manuals/C64UsersManual(IT).pdf