SCI I-602 Combo chip Information R. Grieb 2013

Note: The information about the internal operation of the chip provided here was deduced by making measurements on a chip wired to a PIC MCU on a breadboard. There may be some additional features of the chip that are still undiscovered. Mostly I tried to determine the fcn of registers and bits that the code was writing to or reading.

Chip Pinout:

1	LATCH0b	O	(MTrak DAC LOb)	40	VDD	-	(+5V)
2	A1	I	(Address 1)	39	LATCH1b	O	(MTrak DAC HIb)
3	A2	I	(Address 2)	38	LATCH2b	O	(MTrak SHADb)
4	IOREQb	I	(Z80 IORQb)	37	LATCH3b	O	(MTrak SHENb)
5	A3	I	(Address 3)	36	A0	I	(Address 0)
6	TUNECLK	I	(Comparator output)	35	WRb	I	(Z80 WRb)
7	D0	I/O	(Data 0)	34	IOWRBb	O	(Decoded IO Wr stb)
8	D1	I/O	(Data 1)	33	IORDBb	O	(Decoded IO Rd stb)
9	D2	I/O	(Data 2)	32	VBB	-	(NC)
10	D3	I/O	(Data 3)	31	ROM0b	O	(Decoded Memory Enable)
11	D4	I/O	(Data 4)	30	RAM0b	O	(Decoded Memory Enable)
12	D5	I/O	(Data 5)	29	RAM1b	O	(Decoded Memory Enable)
13	D6	I/O	(Data 6)	28	A15	I	(Address 15)
14	D7	I/O	(Data 7)	27	A14	I	(Address 14)
15	RFSHb	I	(Z80 Refreshb)	26	MREQb	I	(Z80 MREQb)
16	RDb	I	(Z80 RDb)	25	RESETb	I	(Reset Input)
17	E	O	(68B50 Enable)	24	RESOUTb	O	(Reset Output to Z80)
18	4MHZ	I	(Clk input)	23	INTb	O	(7 Msec interrupt)
19	500KHz	O	(Clk output)	22	NMIb	O	(NMIb to Z80)
20	VSS	-	(Gnd)	21	UINTb	I	(68B50 IREQb)

It appears that this chip was made by VTI, and probably designed by SCI, or perhaps jointly. It was used in the MAX and also the Multitrak. I am writing this to provide more information than is available in the service manuals. Hopefully this will help anyone who is working on one of these synths. Some of this information is also in the service manuals.

Memory decoding:

A15	A14	
0	0	ROM0b active
0	1	RAM0b active
1	0	RAM1b active
1	1	Possibly part of the decode for the E signal. This area is used by the 68B50

I/O decoding:

A3	A2	A1	A0	
0	0	0	0	LATCH0b output active
0	0	0	1	LATCH1b output active
0	0	1	0	LATCH2b output active
0	0	1	1	LATCH3b output active
0	1	0	0	IORDb output active for reads, IOWRb output active for writes

0	1	0	1	IORDb output active for reads, IOWRb output active for writes
0	1	1	0	IORDb output active for reads, IOWRb output active for writes
0	1	1	1	IORDb output active for reads, IOWRb output active for writes
1	0	0	0	Register 8 inside combo chip, controls tune cycle count
1	0	0	1	Doesn't seem to be used
1	0	1	0	Low byte of tune period count result read here
1	0	1	1	High byte of tune period count result read here
1	1	0	0	Controls tune clk counting, status read in D0
1	1	0	1	Doesn't seem to be used
1	1	1	0	Watchdog timer low byte?
1	1	1	1	Watchdog timer high byte?

Reset:
Reset is fed from an external circuit, and passed to the Z80. There also appears to be a watchdog timer, which will generate a reset pulse to the Z80 if not written to every so often. Seems like a 17-bit counter is used, as it took 512 mSec to reset when I was running the clk at 2 uSec period. At 4 MHz, it would time out after 64 mSec. The code writes to addresses 0E and 0F in the maskable interrupt handler (every 7 mSec), so the watchdog never times out. The watchdog timer is not mentioned in the service manuals.

500 KHz clk:
This is a fixed 500 KHz clock, generated in the combo chip, and fed out. In the Multitrak it's used by the interval timer, and also by the 68B50 serial chip.

7 Msec Interrupt:
The combo chip asserts this line every 28-29,000 4 MHz clks. Writing anything to I/O address EH clears the 7 mSec interrupt after it has been asserted.

NMI:
The INTRQb from the 68B50 is fed into the combo chip to create the NMIb signal fed to the Z80. Bit 4 of address CH, if set, OR's with the NMIb output forcing it high. NMIb does not appear to be registered or synced to the 4 MHz clk at all. It's just gated. The Multitrak code does disable NMI at startup for a short time, but after that it's always enabled, even during tuning.

Tuning:
The combo chip contains two counters which are used for tuning the voices. The cycle counter counts cycles of the tuning comparator output (AKA "Tune Clk"). This counter is used to gate a second counter, which counts 2 MHz clk pulses, but only while the cycle counter is counting. We can call the second counter the tune period counter. To get good accuracy at higher frequencies, we need to count over more than one cycle of the oscillator being tuned. The cycle counter seems to support counting over 1-31 cycles of the comparator input, although I don't know if the full range is used.

When we write to I/O address 8, several things happen:

Data bits D5-D1 are used to determine the number of cycles we count over:
- 0: 1 cycle
- 1: 2 cycles
- 3: 4 cycles
- 4: 5 cycles
- 8: 9 cycles

16: 17 cycles

these are the ones I tested. I'm sure the other values also work. D6 and D7 do not appear to be used. D0 is used on rd, but maybe not when writing.

Note that D0 is not used for the cycle count.

Writing to address 8 also clears the "count done" flag that can be read on D0, address C. The next read of this address should have a 0 in D0. The sw depends on this.

Writing to address 8 also clears the 16-bit tune period counter.

So we write to address 8 to start the tune frequency measurement, also telling the chip how many cycles of the tuned voice we want to measure the period of.

After we write to address 8, the next low->high edge of the comparator ("tune clk") input starts the tune period counter counting. It counts 2 MHz clocks, using an internal clk divided from the 4 MHz input. If the cycle counter has been set to 1 cycle, on the next low->high edge of the comparator input, the period counter will stop counting, and the "count done" flag will be set. The sw waits for this flag to be set. After the flag is set, the sw reads the value of the 16-bit counter at addresses A (low byte) and B (high).
These values are used to check or adjust the oscillator tuning. If the cycle counter has been set to 2 cycles, we wait for the second L->H edge of the comparator to stop counting. In other words we count over two cycles of the voice waveform instead of one.

It seems that the sequence described above is the same regardless of whether the comparator is high or low when we start counting. In either case, we count between positive edges of the comparator. Actually, the counter starts counting about 5 4 MHz clock cycles after the comparator edge, and stops 5 cycles after the ending edge. There may be some registering and possibly glitch filtering of the comparator input inside the chip. Since it's delayed the same amount when starting or stopping, the count value is not affected.

It appears that bit 3 of the register at address C enables the tune period counter to count if it's set, and inhibits it if cleared. I would guess that it doesn't reset the ctr, just stops it from counting.

The "count done" flag cannot be written to by writing to address C. It can be read at that address, but writes to C do not affect it. Also, once the "count done" flag is set, it will stay set even if the comparator is still pulsing.

The combo chip appears to drive all 8 bits of the data bus when reading from I/O addresses 8-F, even though meaningful data is only available on some bits and at some addresses.

Note that the tune code in the Z80 polls the combo chip, waiting for the "count done" flag to go active. If there is no activity on the comparator due to a problem with one of the voice chips, or the DAC, voltage S/H muxes, or the comparator itself, the Z80 will hang when it tries to tune that voice. There is no time out, and the watchdog won't reset the CPU, as the Z80 is still responding to interrupts and resetting the watchdog every 7 mSec. While the combo chip could have a problem, anything that prevents the tune comparator from toggling will also cause this symptom.

It seems that the "tune clk" (comparator output) input to the combo chip is not a schmitt-

trigger input. If I set the correct DC bias level, I can get normal counting with just a 200 mV pk-pk square wave on this signal.