The eZ80 has 24 address lines, giving a maximum 16M address space. The CPCNG will use 4M blocks in the CPCNG mode. Each 4M region of the 16M address space is mapable to any of 256 4M blocks in the large SDRAM. This gives a maxmimum possible memory size of 1G, though CPCNG users will themselves decide how much memory they want to buy, so the OS must be capable of detecting memory size and operating with whatever is available. Addressing 1 GByte requires 30 address lines. The CPCNG is even able to function when NO main memory is present on the board, in this case it still runs in the CPC mode, with 128 RAM provided by the video RAM.

The mapping takes place as follows:

In CPCNG mode, there are four 8-bit registers CPCNGA0 - 3, one for each of the 4M blocks in the 16M eZ80 address space. These registers reside in the FPGA and are written using I/O ports &0018-&001B. The lower 22 address lines of the eZ80 address bus, a0 - a21 connect directly to the memory. During memory access the upper 2 lines a22 and a23 select one of the 4 8-bit banking registers. The output 8-bits of the selected register drive address lines a22 - a29 of the large memory.

CPCNG mode summary:

a0 - a21	from a0 - a21 of CPU
a22 - a29	from 1 of 4 banking registers CPCNGA0 - 3, selected by a22 and a23 of CP

In the CPC mode, the eZ80 is in native Z80 mode and resets a16 - a23 of its address bus to '0'. In this mode, a0 - a13 still drives the memory directly. a22 and a23 are '0' and therefore in CPC mode only the lowest 4M block of the 16M address space is used. All 16K memory blocks that are used for the CPC mode must therefore exist in the same contiguous 4M block of memory, but that can be any 4M block chosen from the entire memory. One 4M block contains 256 16K CPC blocks.

In CPC mode, four 8-bit FPGA registers CPCRDA0 - 3 on I/O ports &0010-&0013 select four 16K blocks for reading from memory. a14 and a15 of the eZ80 address bus select one of these 4 registers which drives address lines a14 - a21 of the main memory. Therefore each of the 4 16K sections of the native Z80 mode 64K address space can be mapped for reading from any of 256 16K blocks of the selected 4M block from main memory.

Similarly, another four 8-bit registers CPCWRA0 - 3 on I/O ports &0014-&0017 select 4 16K blocks for writing to memory.

The reason for having separate registers for the reading and writing memory maps is that in a

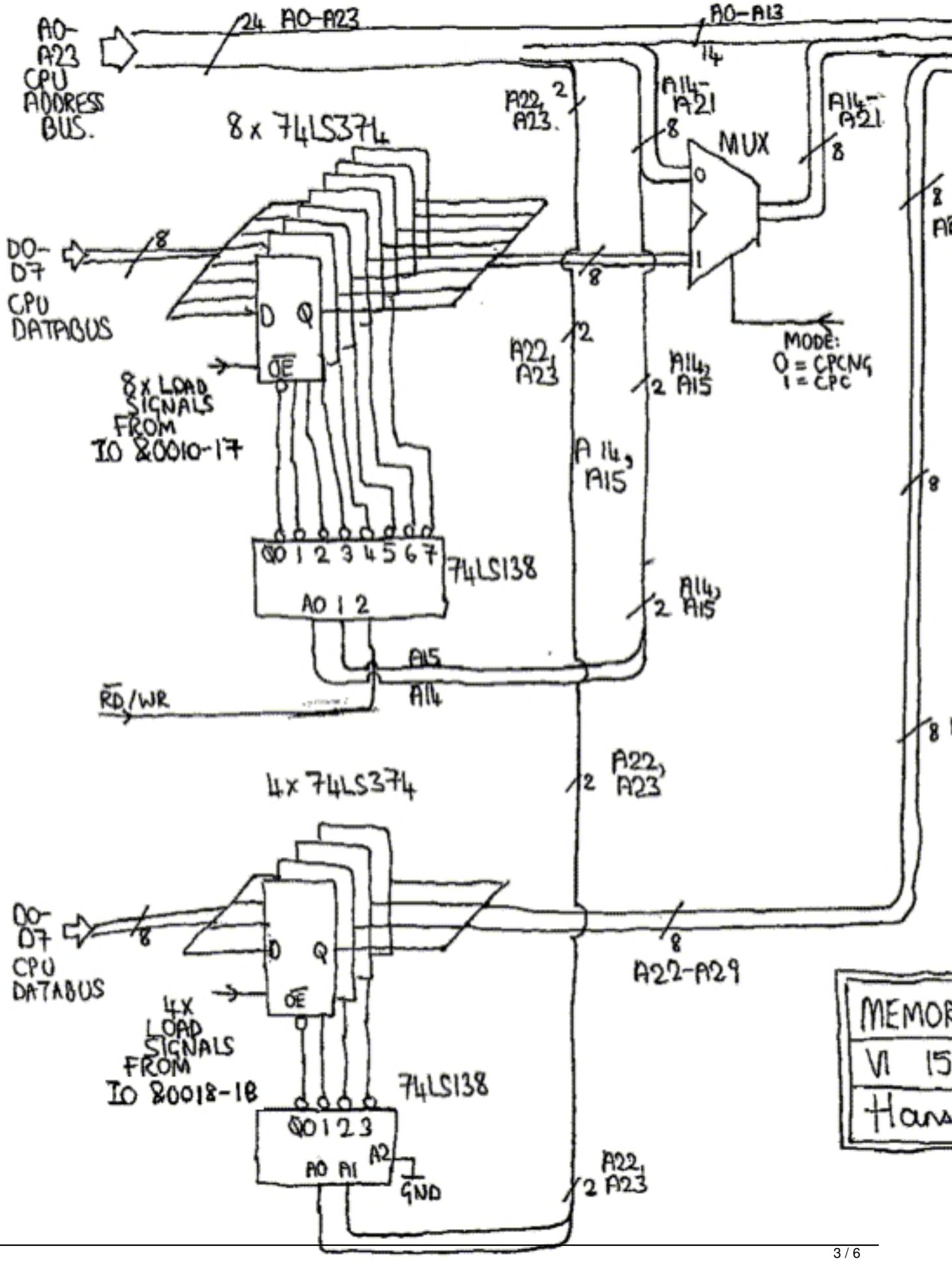
real CPC, if the ROM is mapped to a particular 16K block, reading takes place from ROM but writing to addresses in the block writes to the underlying RAM. In the CPCNG in CPC mode, I allow different parts of memory to be used for reading and writing for each of the 16K blocks. If a 16K block was to be used as ordinary RAM, the read and write registers would contain the same value. If a 16K block of the CPC was to be mapped as ROM, any 16K block of the main memory can be used for that ROM, and a different 16K block for the RAM at that location.

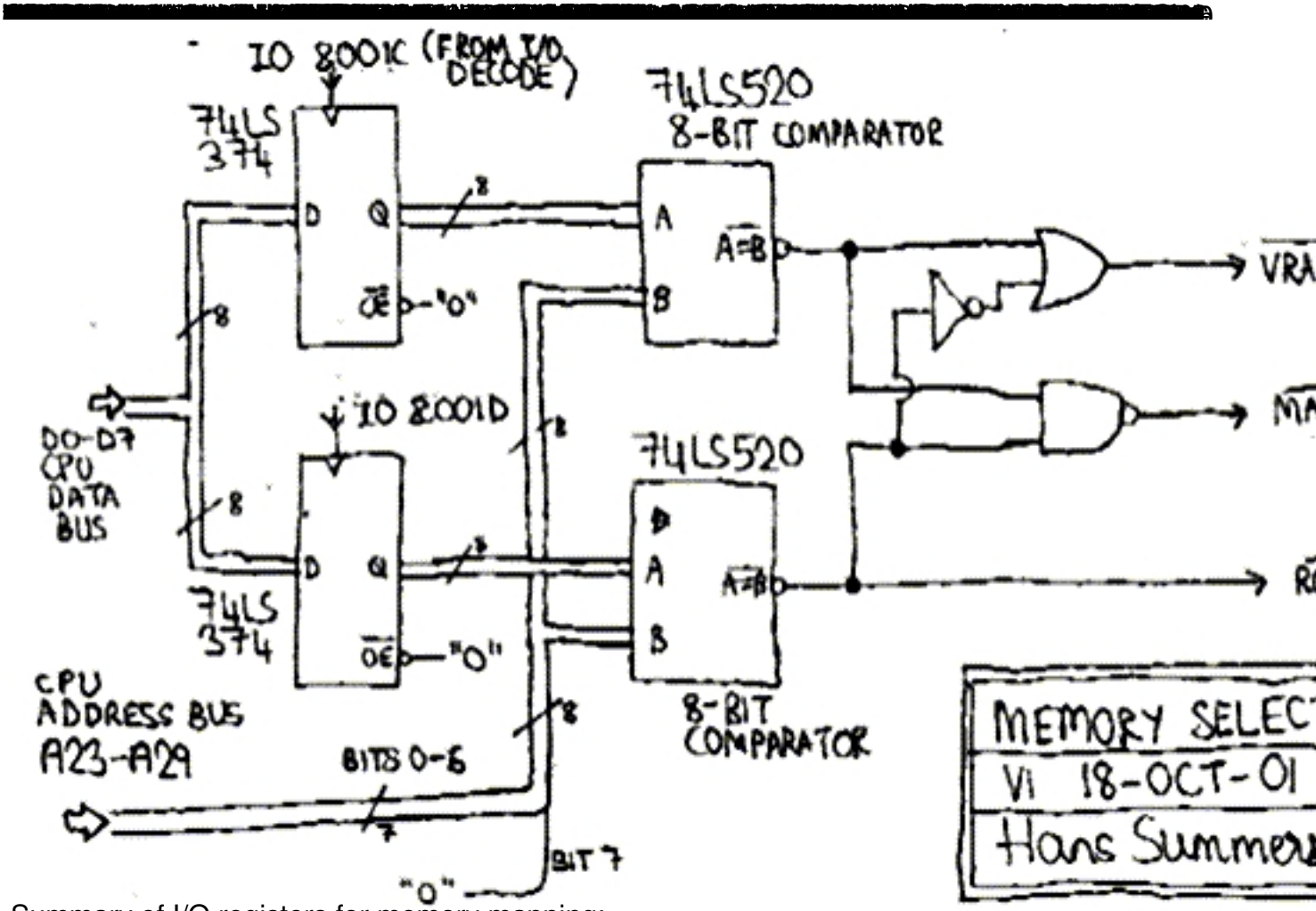
All of the CPC memory mappings are available using this method (100% compatibility), and it is possible to accomodate without too much difficulty in hardware, peacefully coexisting with the larger available memory and CPCNG mode mapping. Crucially, when an I/O occurs in CPC software, this will be handled by the I/O intercept trick: a /NMI (non maskable interrupt) is generated, calling routine at &0066. This will decode the CPC I/O request and in the case of memory mapping I/O, translate the old-style codes as documented at http://andercheran.aiind.upv.es/~amstrad/ , to the necessary values for the 8 CPC-mode memory map registers CPCRDA0 - 3 and CPCWRA0 - 3.

CPC mode summary:

a0 - a13	from a0 - a13 of CPU
a13 - a21	READ: from 1 of 4 banking registers CPCRDA0-3, WRITE: from 1 of 4 banl
a22 - a29	from NG-mode banking register CPCNGA0

Below is my draft circuit diagram for the mapping unit using equivalent TTL 74LS-series part numbers. The multiplexer switches A14-A21 of the memory address bus from A14-A21 of the CPU address bus in CPCNG mode to the outputs of the CPC-mode mapping registers.


A0-A23 CPU ADDRESS BUS.
24 A0-A23
A0-A13
14
A14-A21
A22, A23.
2
8 x 74LS374
MUX
8
0
1
D0-D7 CPU DATABUS
8
D Q
OE
8 x LOAD SIGNALS FROM IO &0010-17
A22, A23
2
A14, A15
MODE: 0 = CPCNG 1 = CPC
A14, A15
Q0 1 2 3 4 5 6 7
74LS138
A0 1 2
A15
A14
RD/WR
4 x 74LS374
A22, A23
D0-D7 CPU DATABUS
4X LOAD SIGNALS FROM IO &0018-1B
74LS138
Q0 1 2 3
A0 A1 A2
GND
A22-A29
MEMOR
VI 15
Hans



Summary of I/O registers for memory mapping:

Port	Register	Mode	Function
&0010	CPCRDA0	CPC	1 of 256 16K blocks for F
&0011	CPCRDA1	CPC	1 of 256 16K blocks for F
&0012	CPCRDA2	CPC	1 of 256 16K blocks for F

&0013	CPCRDA3	CPC	1 of 256 16K blocks for F
&0014	CPCWRA0	CPC	1 of 256 16K blocks for W
&0015	CPCWRA1	CPC	1 of 256 16K blocks for W
&0016	CPCWRA2	CPC	1 of 256 16K blocks for W
&0017	CPCWRA3	CPC	1 of 256 16K blocks for W
&0018	CPCNGA0	Both	1 of 256 4M blocks for &0
&0019	CPCNGA1	CPCNG	1 of 256 4M blocks for &4
&001A	CPCNGA2	CPCNG	1 of 256 4M blocks for &8
&001B	CPCNGA3	CPCNG	1 of 256 4M blocks for &C

&001C	VRAMA	Both	Location of 128K Video R
&001D	ROMA	Both	Location of 128K ROM m