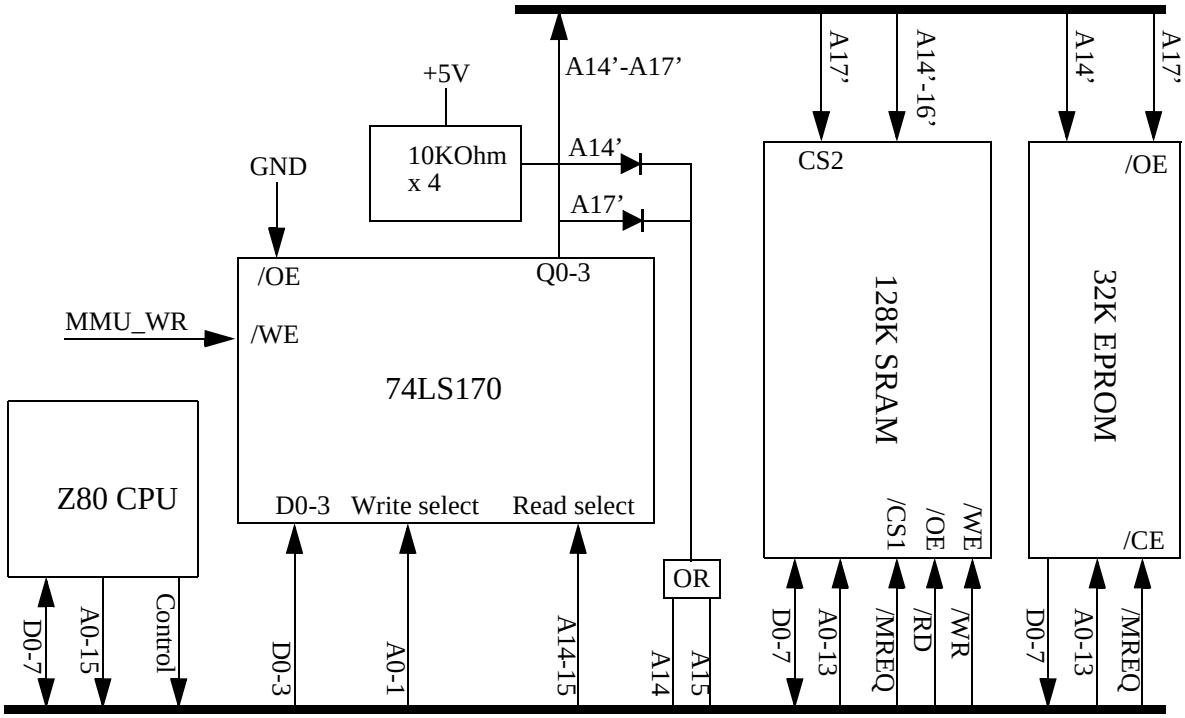
Memory Management Unit



Functional description

This circuit will extend the Z80 address space to 18 bits (256 Kbytes). A bank switching scheme is used where the Z80 address space is divided into four 16KB pages and each page can be mapped to any physical memory address in the 256KB address space. The mapping is done with the 74LS170 register file IC. It contains four 4 bit registers with separate data inputs/outputs, separate read/write address inputs and read/write strobes. The MMU (74LS170) is configured by the CPU by writing to any of the four I/O-addresses assigned. The MMU_WR (active low) signal could be decoded like this to place the MMU at I/O-addresses A0-A3 (hex):

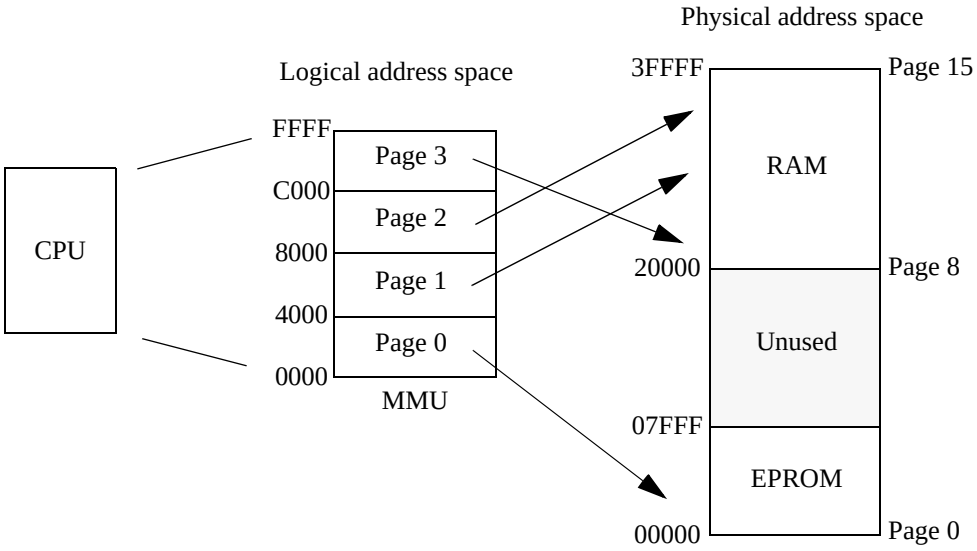
[1]MMU_WR = IORQ + WR + MMU_SELECT
MMU_SELECT = ~A7 + A6 + ~A5 + A4

Note that the MMU has open collector output which explains the four 10K pull-up resistors on the A14'-A17' signals. The OR-gate and the two diodes are used to force EPROM at CPU address 0-3FFF. This is needed to ensure that the CPU starts executing code from the EPROM after a reset.

1. Logic symbols used: + means OR, * means AND, ~ means NOT

Address mapping

The purpose of the MMU circuit is to expand the amount of memory that can be used in a Z80 system. By updating the MMU registers during the execution of a program, more than 64KB of code and data can be used. The picture below illustrates how the address mapping works.



Each MMU register represents one logical page and contains a mapping to the corresponding physical page. Example: the picture above shows one possible mapping where logical page 0 is mapped to physical page 0, logical page 1 is mapped to physical page 10 etc. In table form it looks like this:

Logical address	Logical page	MMU content	Physical page	Physical address
C000-FFFF	3	8	8	20000-23FFF
8000-BFFF	2	14	14	38000-3BFFF
4000-7FFF	1	12	12	30000-33FFF
0000-3FFF	0	0	0	00000-03FFF

In OS-X the MMU is used to give each concurrent process a separate address space. Each process has it’s own 32KB RAM in logical page 1-2, shared RAM is mapped in logical page 3, and the operating system entry points are always in logical page 0. The MMU is managed by the operating system and the processes can communicate through the shared memory using message passing (also managed by OS-X).