**Extern Kernal 8 – REX9628 Rev. 0**

Functional **Description**

J1 is the edge connector for the C64 expansion port. ST1 and ST2 are the two EPROMs, that hold the kernals.

IC1 is responsible for the cartridge being able to run a kernal. is the output of the NAND gate IC1. It can switch on or off with SW2-10 to the configuration signal of the expansion port. In case it is switched off, the cartridge is inactive. IC1 produces an extra chip select signal (taking a detour via and ). The logic equation for the latter signal is:

*!ROMH = (\_HIRAM & A15 & !A14 & A13 & !\_AEC & R\_\_W & !\_EXROM & !\_GAME*

*# A15 & A14 & A13 & !\_AEC & \_EXROM & !\_GAME*

*# \_AEC & \_EXROM & !\_GAME & VA13 & VA12 );*

Equation 1

Obviously is required for to get LOW (=active). A HIGH level on (or an open SW2-10) will inhibit the EPROM access.

There is a difference between this and the internal kernel chip select: The internal kernel ROM can be switched off by software to get read access to the RAM in the same address space. This is done with the signal. The latter signal is not contained in the 2nd line of equation 1, which is responsible for to be mapped to 0xE000 – 0xFFFF.

The logic equation for the internal chip select is:

*!KERNAL = (\_HIRAM & A15 & A14 & A13 & !\_AEC & R\_\_W & \_GAME*

*# \_HIRAM & A15 & A14 & A13 & !\_AEC & R\_\_W & !\_EXROM & !\_GAME );*

Equation 2

A possible solution which fixes the problem is connecting the signal (CPU 6510, pin 28) to the NAND-Gate via the solder pad “TP1”. It is wired to an input pin of IC1.

The logic equation for the signal is:

!GAME\* = ( PHI2 & A15 & A14 & A13 & R\_\_W & BA & CS )

Equation 3

PHI is the system clock Φ2, BA is the Bus Available signal, generated by the VIC-II chip.

SW2-9 can hold the signal low, when switched on. This was a game stop function is provided.

The first 8 switched of SW1 select the kernal. IC4 decodes these 8 switched into an octal number output on A0..A2. The bit A2 ()selects which EPROM is active. and will be converted into the address bits A13 and A14 of the EPROMs. This information is stored in a latch mechanism consisting of the bus driver IC2 and the Inverter IC3.

IC2 will let the three bits pass as long as the signal ROMH is low. IC5A inverts the bus chip-select for this purpose. Hence the information is passed through as long as , which also serves as a chip select for the EPROMs is inactive (HIGH). The paired inverters are functioning as a storage element. As long as the bus drives are HIZ (ROMH = HIGH), it holds the logic level, that was asserted by the bus driver IC2 before.

The inverter IC5B generates the enable signal for EPROM ST1 from which is the chip enable signal of ST2.

The Address bit EA13 is connected to both EPROMs, while EA14 is switched separately by SW1 for each EPROM in order to configure the EPROM size/type for ST1 and ST2. In case the address bit is not required (for 27c128 and 27c64), it can be set to HIGH with SW1.

SW3 is a reset switch. When pressed, a system reset is performed.

The circuitry is reverse engineered from an original REX Datentechnik 9628 cartridge. It is provided “as is”. It has some weak points: There are several open inputs, which act as HIGH in the LS logic family, but might be troublesome with other logic families. A good practice is, to connect them to a fix logic level. For IC1, that would be connecting pin 11 and pin12 and holding them high with a (10k) Pull-up resistor. SL1..SL8 on IC4 should also have a pull-up resistor (network).

The not used gates (IC2C, IC5D and IC5C) should also be connected to HIGH (+5V) on each input.

Further on, there is only one blocking capacitor for the whole cartridge. This might work, a better practice would be a blocking capacitor (100n) for each IC.