Tanex Plus – Banked/Paged EPROM Circuit Description

Section 1 of this document shows the original discrete logic solution along with a description of operation.

Section 2 lays out the programmable logic solution using 2 x GAL16V8 devices and provides a VERY brief GAL primer enough to be able to follow the Boolean expressions.

1a) EPROM Bank & Page Logic

Working from the top of diagram fig 1, the I/O address range signal (active high) is qualified with the R/W and 02 clock signal in U3A to produce an active low logic state on its output when a memory write operation is performed in the I/O address range BCxx to BFxx. This signal is used to enable the outputs on a 2 line to 4 line decoder IC (U7A) to become active low in response to the inputs A0 & A1. Each of the outputs of U7A corresponds to a segment of the I/O address space range & Jumper block P4 enables user selection of the required upper address range byte.

The lower address byte is decoded by 8 bit magnitude comparator U4, the address line inputs on P0 through P7 are compared to the value set by DIL switch block SW1 (Q0 through Q7) and if there is a match then output P=Q goes active low. NOR gate U10A qualifies U4 pin 19 against the jumpered output (P4) of U7A and if both outputs are low then output pin 1 goes high causing U12 (D type latch) to latch the lowest 4 bits of the CPU data bus and present this value on its output pins.
These output pins are used to manage EPROM bank and page switching in EPROMs U16 & U17.

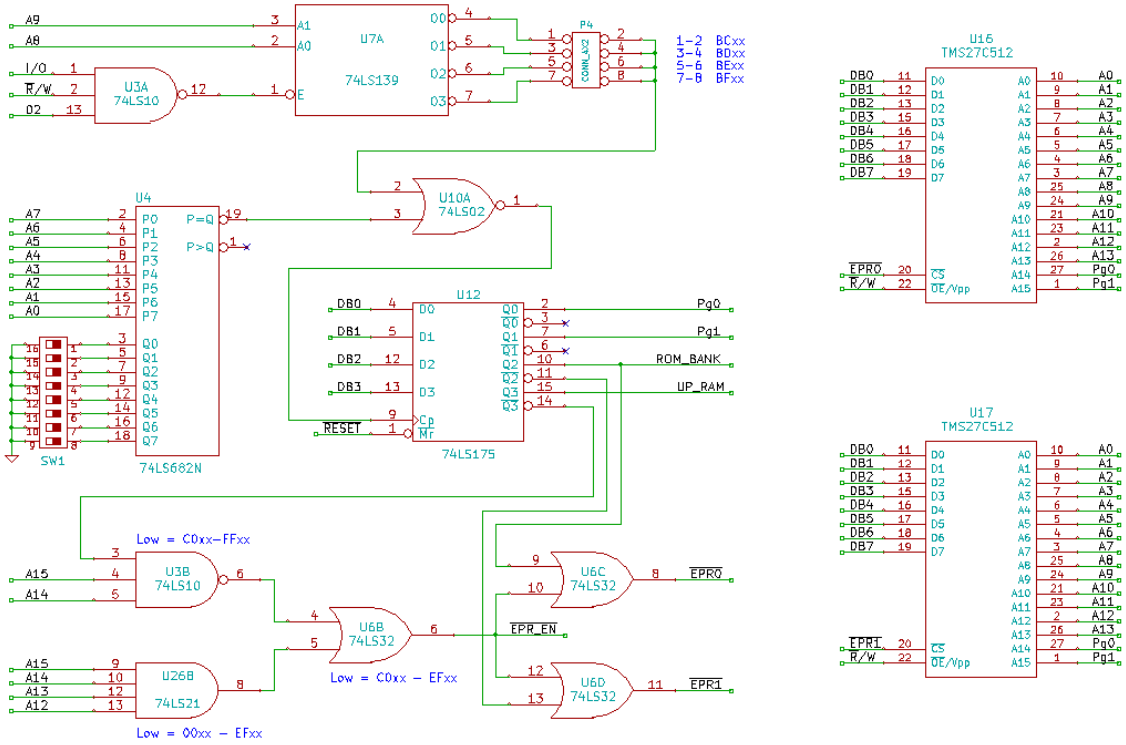


Fig 1

1b) EPROM Address Logic

We need to ensure that EPROMs U16 & U17 are only active when the address range C0xx – EFxx is being addressed, this is achieved using 3 gates U3B, U26B, & U6B.

U26B logically ANDs the upper 4 address lines A12 through A15 so that the output pin goes high for addresses on the range F0xx – FFxx and low for all other values, U3B is looking for A14 & A15 to be logic high at the same time that U12 latch pin 14 (inverted output) is active high (bit 4 if high indicates RAM, low indicates EPROM). In this condition U3B output goes low corresponding to address range C0xx – FFxx, by combining both outputs U3B & U26B in OR gate U6B, the output of U6B will go low for any memory address accessed in the range C0xx – EF00.

The active low EPR_EM signal is then used to qualify and steer the active low 'ROM_Bank' signal to generate EPR0 & EPR1 which drive the EPROM chip select pins.

1c) SRAM Address Decode Logic

The purpose of this circuit is to ensure RAM is enabled for all memory locations except those assigned to a reserved function, i.e XBUG, TANBUG, MT65 1K RAM.

U8 performs the function of carving out reserved areas of the memory map by monitoring the various status lines, if these status lines are all inactive (High) pin 8 of U8 will go low enabling the SRAM U18 and OE & WE buffers (U14C, U14D), if on the other hand one of the status signals are active then U8 pin 8 will be high and SRAM U18 will be disabled and the read/write signals (OE & WE) will be driven to a logic high (inactive).

The RAM read/write signals (OE & WE) are generated by U9A & U9B using the 02 clock signal to ensure that OE and WE are only active during the 02 active CPU cycle when data is valid on the CPU data bus.

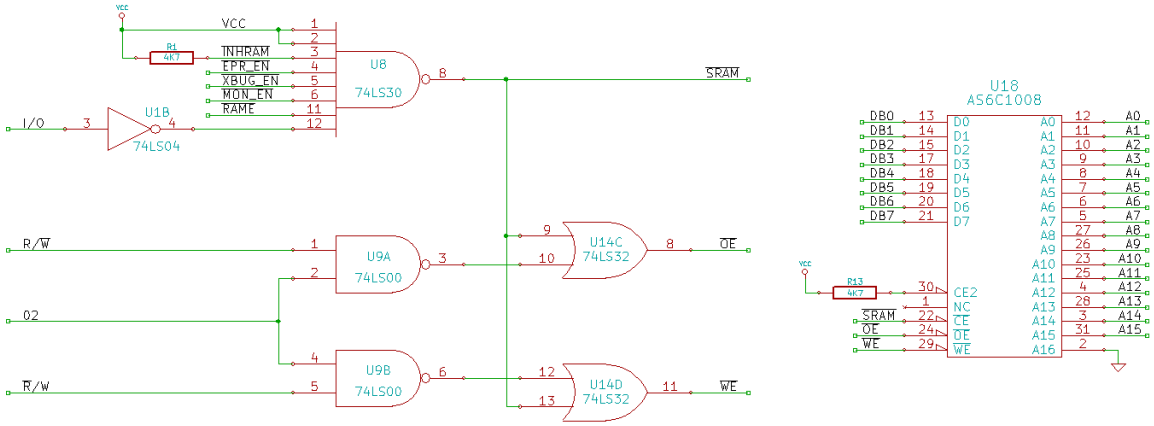
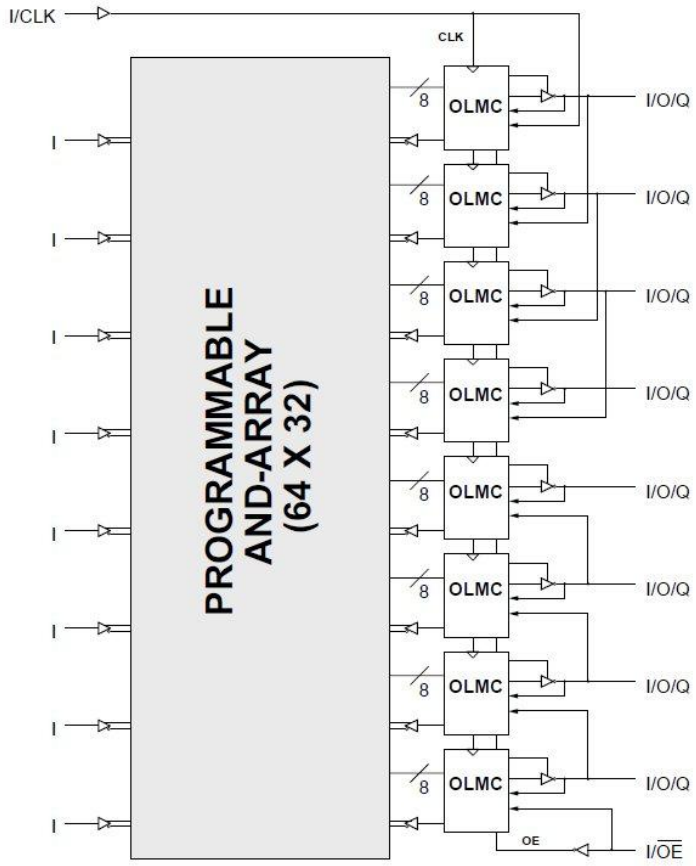


Fig 2

2a) Programmable Logic – The Basics

The purpose of this section is to give a VERY basic cut down overview of the GAL16V8 internal architecture, and to describe the basics of the programming language so that the Tanex-Plus GAL (U3 and U7) boolean equations can be understood.

In brief, the GAL16V8 PLD is a 20 pin device of which 8 pins are dedicated input pins, 8 pins are programmable I/O pins, and 2 pins are special purpose which will not be described. The electrical pathway between the input pins and the programmable I/O pins are described by a logical boolean expression which notates the logic function required. The boolean expression can be built up in such a way as to enable quite complex logical functions to be carried out providing the benefit of reducing the chip count of a hard ware design.



The diagram above shows the main features of the architecture, each dedicated input pin can be programmed for either 'True' or 'Inverted' input, these inputs then feed into a programmable mesh which performs a logic 'AND' operation on it's inputs, the output of the 'AND' operation is then passed to a block called an 'Output Level Macrocell' which can be programmed for 3 different modes of operation, for now just think of it as a multiple input 'OR' gate.

There are a number of logical operators that we will use to build up our boolean expression and these are listed below:-

'*' denotes an AND operator
'+' denotes an OR operator
'/' denotes a NOT operator

The boolean expression format for a given I/O pin is:-

Output = Input1 AND Input2
OR Input3

Note:- It is convention to put each OR instance on a new line

Taking the first line of our example we can see that it perfoms the function of a simple two input AND gate, if Input1 AND Input2 are logical high then Output will be high. We can simulate a NAND gate by inverting the Output argument with a NOT operator

/Output = Input1 AND Input2

Similarly we can simulate a NOR gate by inverting both inputs

Output = /Input1 AND /Input2

So by combining inputs and operators we can express a complex logic function, also because of the design of the 'Output Level Macrocell' the I/O pins are folded back into the programmable 'AND-Array' enabling an output from one expression to be used as an input to another expression, and we will do that in our application on the Tanex Plus.

2b) GAL U7 SRAM Address Map Logic

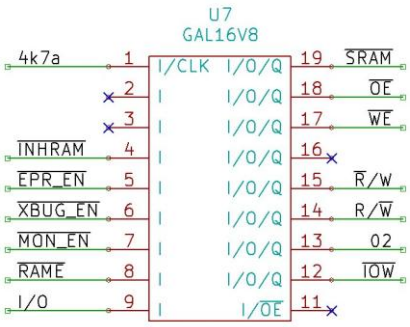


Fig 3

The primary purpose of GAL U7 is to provide the neccesary output control signals (SRAM, OE & WE) to insert SRAM memory into the address map of the MT65 system. The fourth output control signal (IOW) is used by GAL U3, but is generated on this chip U7 as the necessary inputs are already present and there was insufficient spare input pins on U3 to place the function on that chip.

/SRAM = INHRAM * EPR_EN * XBUG_EN * MON_EN * RAME * /IO

Our first boolean equation is made up of six inputs and one output and describes the active low output pin /SRAM, this pin connects to the SRAM chip enable pin and allows RAM to be present in the memory map whenever all the equation inputs are logic level high (inactive).

Note, that to make the output expression active low we apply a 'NOT' operator to our output label. We have also inverted the equation input 'I/O' with a 'NOT' operator because in the MT65 this line is active high for I/O operations (BCxx – BFFF) and we need this input to be high when I/O is inactive.

/OE = /SRAM * RHWL * O2

Our second boolean equation is made up of three inputs and one output, notice that one of the inputs is our SRAM signal from above, we need to invert this signal so that it becomes an active high input. The other two inputs are the clock signal 02 which when high signifies that the databus is active for program data, and the other signal is Read High Write Low which when high indicates a Read operation. These three signals are ANDed together to produce our SRAM read signal OE which is inverted to make it active low.

Note:- The OE signal and WE signal (below) are only produced when our SRAM chip is active.

/WE = /SRAM * RLWH * O2

The active low SRAM write signal is generated in a similar fashion to the above read signal except the Read Low Write High signal is used to signify a memory write operation.

Note:- We could have just inverted the RHWL signal but as RLWH was available I used this instead.

/IOW = IO * RLWH * O2

The active low IOW (I/O write) signal indicates when a memory write operation occurs in the memory I/O area BC00 – BFFF , this signal is used by GAL U3.

2c) GAL U3 EPROM Bank & Page Logic

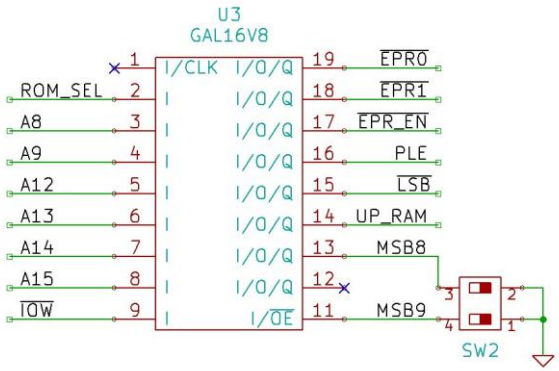


Fig 4

The function of GAL U3 is to provide the necessary clock signal (PLE) to latch the EPROM bank and page data into the 4 way D type latch U9 (74LS175), and to provide the active low EPROM selection and enable lines (EPR0, EPR1, & EPR_EN)

PLE = /IOW * /LSB * /A9 * /A8 * /MSB9 * /MSB8
+ /IOW * /LSB * /A9 * A8 * /MSB9 * MSB8
+ /IOW * /LSB * A9 * /A8 * MSB9 * /MSB8
+ /IOW * /LSB * A9 * A8 * MSB9 * MSB8

The first Boolean equation looks a little more complicated than it actually is. Basically four separate 6 input AND expressions are ORed together and subtle use of the NOT operator allows for the four combinations of switch SW2 to be decoded as four upper address locations for the 4 bit data latch U9.

/EPR_EN = /UP_RAM * A15 * A14 * /A13 * /A12
+ /UP_RAM * A15 * A14 * /A13 * A12
+ /UP_RAM * A15 * A14 * A13 * /A12

EPR0 = ROM_SEL + EPR_EN

EPR1 = /ROM_SEL + EPR_EN