5801 CPU Controller

by Daniël Mantione

August 2024

Introduction

The 5801 CPU controller is a design for a GAL16V8 or GreenPAK 46620 that allows you to implement a 6510, 8500, 8501 and 8502 compatible CPU with a 6502 CPU and low cost standard components. NMOS 6502 CPUs are still plenty available in the 21th century and therefore a design with a 5801 CPU controller allows you to design a CPU solution for Commodore 8-bit computer mainboards with components that are still available.

The 5801 CPU controller can be used to create adapters that convert a 6502 to a 6510, 8500, 8501 and 8502, but is also suitable to create modern replacement mainboards that do not require difficult to find parts.

Why 5801?

The 5801 CPU controller is implemented by a GAL16V8. A GAL is a gate array. In MOS Technology product numbers, chips with a model number 5000 range were gate arrays. For example the CIA replacement in the C128DCR 1571 is a 5701. The chip in the Commodore 1351 mouse has model number 5717 and Amiga Gary has number 5719. Considering that a GAL is about arrays, a number in the 5000 range fits well with MOS Technology chip numbering, and 5801 is 8501 with two digits reversed.

Creating a 5801 controller

An 5801 CPU controller can be created from either GAL or GreenPAK chips. Creating an 5801 CPU controller is as simple as taking a chip and programming the required chip configuration file file.

In case of a GAL, either a GAL16V8 or ATF16V8 can be used. A popular programmer that can be used is the TL866 from XGecu. You can program the provided .jed file with the TL866. The speed of the GAL does not appear to be very important, multiple speeds have been successfully used.

In case of a GreenPAK, an SLG46620 can be used. In this document the SLG46620G is assumed, but the SLG46620V can be used as well. An SLG4DVKDIP or SLG4DVKADV with socket adapter can be used to program the .gp4 file into the SLG46620. The software required is the Renesas “Go Configure Software Hub”.

Pin description

The 5801 controller has the following pinout:

GAL 16V8

```
                  --------------
                 |5801_contrllr |
         i_reset x---|1          20|---x Vcc
     i_addror1_7 x---|2          19|---x o_ddr2data
          i_phi2 x---|3          18|---x o_rw
            i_rw x---|4          17|---x o_port2data
          i_phi0 x---|5          16|---x o_ddrlatch
           i_aec x---|6          15|---x o_portlatch
    i_addror8_15 x---|7          14|---x o_dataoe
          i_gate x---|8          13|---x t_aec_delay
            i_a0 x---|9          12|---x o_addroe
             GND x---|10         11|---x i_no_ddr
                 |______________|
```

GreenPAK SLG46620G (TSSOP)

```
                  --------------
                 |5801_contrllr |
             Vcc x---|1          20|---x Unused
         i_reset x---|2          19|---x o_ddrlatch
           i_aec x---|3          18|---x o_port2data
          i_phi2 x---|4          17|---x o_rw
            i_rw x---|5          16|---x o_portlatch
          i_phi0 x---|6          15|---x o_dataoe
     i_addror1_7 x---|7          14|---x o_ddr2data
    i_addror8_15 x---|8          13|---x o_addroe
            i_a0 x---|9          12|---x i_gate
        i_no_ddr x---|10         11|---x GND
                 |______________|
```

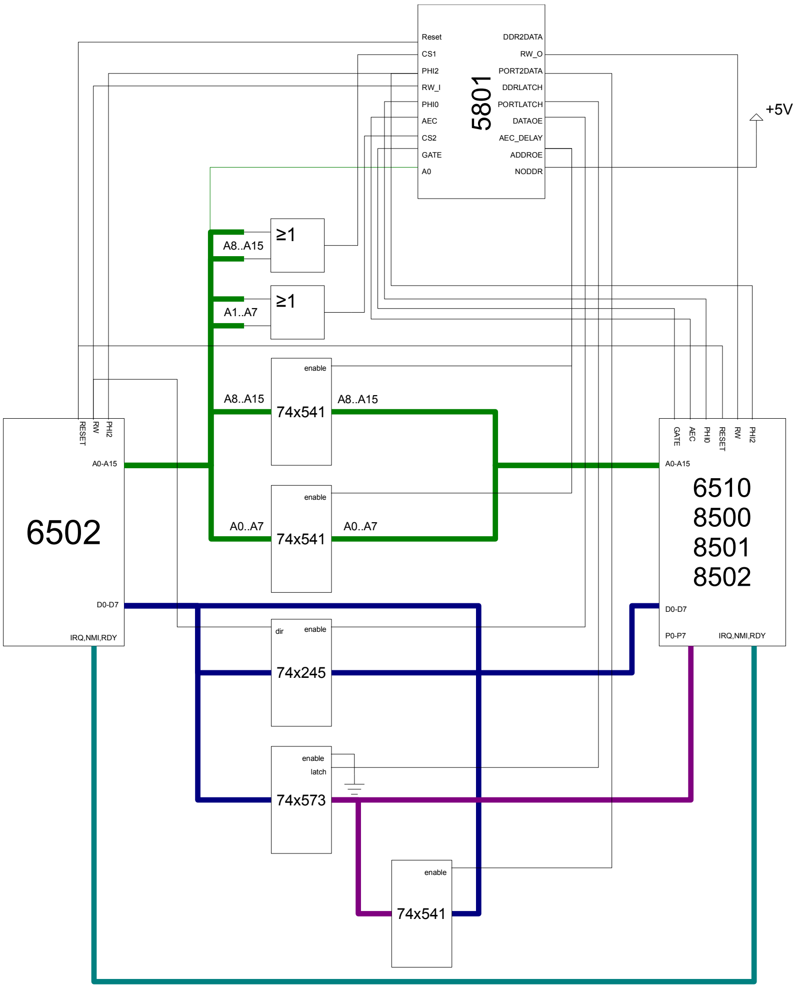
Pin GAL	Pin Green-PAK	Signal	I/O/T	Description
1	2	i_reset	I	This should be connected to the RESET line of the computer. The controller uses the reset signal to implement the right CPU behaviour during a reset,
2	7	i_addror1_7	I	First "chip select" for the CPU port. This line should be connected to an OR function of address lines A1 to A7 of the 6502.
3	4	i_phi2	I	Main clock. This line should be connected to the PHI2 line of the 6502.
4	5	i_rw	I	Read/write. This line should be connected to the RW line of the 6502.
5	6	i_phi0	I	Clock input. This line should be connected to PHI0 of the imitated CPU.
6	3	i_aec	I	Address enable control. This line should be connected to PHI0 of the imitated CPU.
7	8	i_addror8_15	I	Second "chip select" for the CPU port. This line should be connected to an OR function of address lines A8 to A15 of the 6502.
8	12	i_gate	I	Gate IN. If the imitated CPU has a GATE IN pin (8501), then this pin should be connected to GATE IN of the imitated CPU. Otherwise the pin can be left unconnected, as the GAL has a built-in pull-up register.
9	9	i_a0	I	Address line A0. Should be connected to A0 of the 6502.
10	11	GND		Ground
11	10	i_no_ddr	I	Data direction register enable. If this pin is low, control signals for the Data Direction Register will be generated. If this pin is high, a hard wired data direction register is assumed and no control signals for the DDR are generated. The pin can be left unconnected, as the GAL has a built-in pull-up register.
12	13	o_addroe	O	Address bus output enable. This is a control signal that should be connected to 74x541 chips to enable/disable the address bus driving.
13	X	t_aec_delay	T	Internally used signal to generate a delayed AEC. Do not connect.
14	15	o_dataoe	O	Data bus output enable. This is a control signal that should be connected to a 74x245 chip to enable/disable the data bus driving.
15	16	o_portlatch	O	CPU port latch. This pin should be connect to the latch enable of a 74x573 or clock pin of a 74x574. The 5801 generates a short spike, therefore it should matter whether you use an 8-bit latch or flip-flop.
16	19	o_ddrlatch	O	Data direction port latch. This pin should be connect to the latch enable of a 74x573 or clock pin of a 74x574. The 5801 generates a short spike, therefore it should matter

				whether you use an 8-bit latch or flip-flop.
17	18	o_port2data	I	CPU port to data bus. This pin should be connected to the output enable of a 74x541 to output the CPU port onto the data bus if required.
18	17	o_rw	O	R/W out. Should be connected to R/W of the imitated CPU.
19	14	o_ddr2data	O	Data direction port to data bus. This pin should be connected to the output enable of a 74x541 to output the contents of the data direction register onto the data bus if required.
20	1	Vcc		5V power supply

Schematics for a CPU

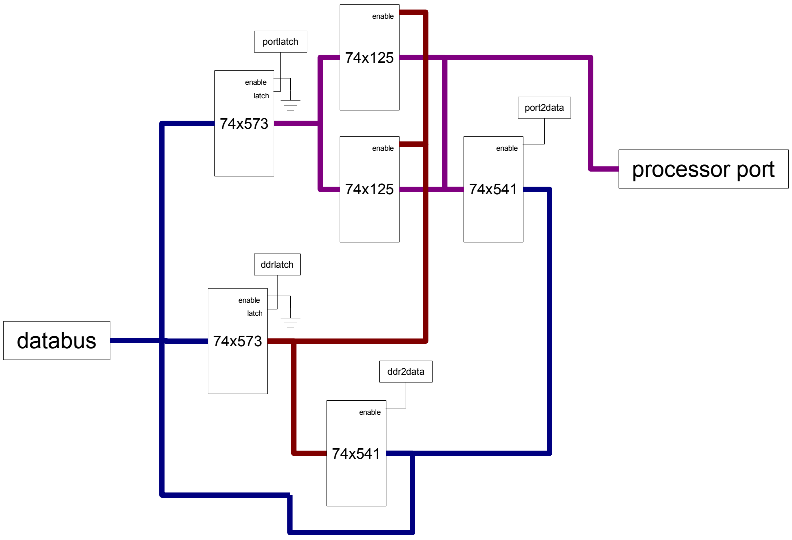
On many 8-bit Commodore computers, it doesn't make sense to change the data direction register on the CPU port from software. The DDR is initialized once when the computer reset and never changed again. Therefore it is often feasible to hardwire the Data Direction Register.

If the DDR is hardwired, a schematic of a CPU replacement can look as follows:



In other words, you need to implement two or gates (can be done with diode logic or ICs), three 74x541, a 74x573 and a t4x245. The direction of a pin on the CPU port is decided by the output pins of the 74x573: If a pin of this chip is not connected to the CPU port, the pin is an input, if the pin is connected, the pin is an output.

If a fully functional data direction register is desired, the NODDR pin needs to be connected to ground rather than 5V and the 74x573 and 74x541 that is connected to the CPU port in the above schematic (the purple line), can be replaced with the following schematic:



Selection of external components

The 5801 CPU controller allows some freedom in the selection of external components. One choice you will have to make is the logic family. The best results were achieved with HC/HCT logic chips. Faster chips such as FCT and ACT did result in working CPUs as well, but some minor compatibility problems were observed with some RAM chips.

We will discuss each of them:

74x245 - This chip is best implemented with a 74HCT245, as the MOS CPUs use TTL compatible logic levels. Nevertheless, a while on paper a 74HC245 is a less good idea, it did work surprisingly well in practise.

74x541 - These chips does not need to be TTL compatible, because the NMOS 6502 is compatible with CMOS logic levels. Therefore use either 74HC541 or 74HCT541. It should be possible to use 74x244 chips as well instead of 541.

74x573 - This chip does not need to be TTL compatible, because the NMOS 6502 is compatible with CMOS logic levels. Therefore use either 74HC573 or 74HCT573. You can also use a 74x574, because the 5801 will generate a short spike at the right time, which is both compatible with a latch and a flip-flop. In addition, it should be possible to use 74x373 or 74x374 as well.

74x125 - These chips do not need to be TTL compatible, because they would be driven by the CMOS compatible 74x573.

The OR gates can be implemented as diode logic, good results have been achieved with standard switching diodes such as the 1SS355 and 5.6kΩ pull-down resisters. You can of course also use chips to implement the OR gates.

GAL16V8B / ATF16V8B / SLG46620 datasheet

It is recommended to read the datasheet of the GAL or SLG46620 that you are using to implement the 5801 CPU controller to have the full specifications.