

POLITENICO DI MILANO

DIPARTIMENTO ELETTRONICA, INFORMAZIONE E
BIOINGEGNERIA

HEAPLAB PROJECT REPORT

DRAM Tester PCB Design

Author:
Davide FANZAGA

Supervisor:
Dr. Federico
REGHENZANI

December 30, 2020

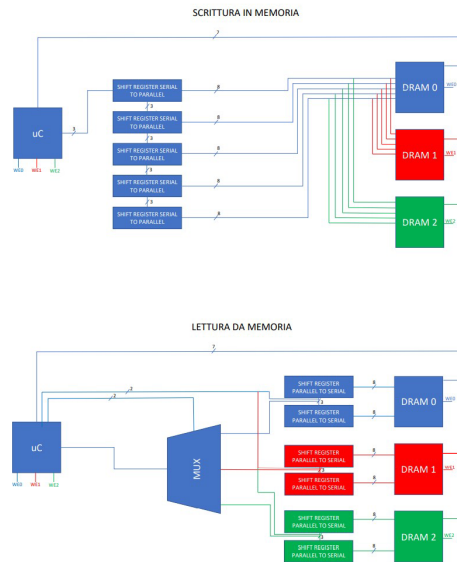


Abstract

This project was about designing a PCB layout. The circuit we wanted to design was meant to detect faults in the data stored in DRAMs. This problem is most common in space where cosmic rays can flip a bit stored in a memory cell. We started from the general idea for then moving to the selection of the components, the creation of the schematic, and finally the realization of the board.

1 General idea

The general idea of the project is to write in three DRAMs the same data at the same address and with two microcontrollers continuously checking if a datum in a DRAM is different from the datum at the same address of the other two. If this happens it means that there has been a fault. These faults are stored in the EPROM of the microcontrollers and can be extracted through an I2C interface.

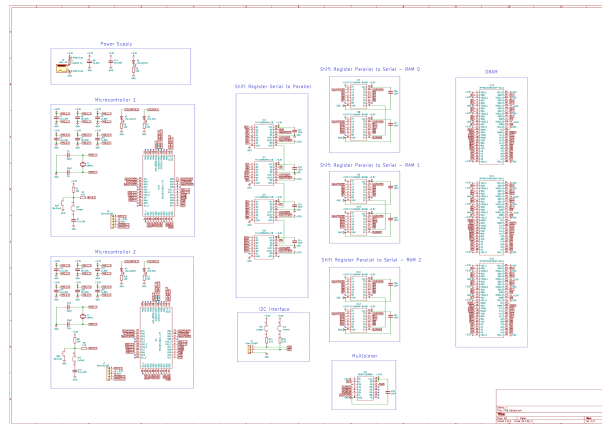


2 Choice of the components

For microcontrollers, we decided on the PIC18F47Q43 since it has more than enough GPIOs for our purpose. At first, we choose a DIP package that can be easily plugged into a socket and reprogrammed outside of the board, but then, due to the large number of links to do, I decided to change it into an SMD package and add the programmer connectors on the board. For DRAMs, we decided on the MT48LC16M16A2P that has a 256Mb memory (16 bits for the datum and 24 bits for the address). Since a direct connection of the microcontroller with the DRAM would have been unfeasible due to the large number of links, we decided to use some shift registers (from serial to parallel in the writing phase and from parallel to serial in the reading phase) in between. Shift registers introduce a significant delay but in our case it's not required a tight timing constraint. Finally, we decided to use a multiplexer (IDTQS3VH253) to decide which of the three DRAM read.

3 Schematic

For the Schematic design, I decided to use a free and open-source program: KiCad. KiCad already includes a lot of different symbols and footprints but unfortunately not all of the ones we decided to use. To overcome this problem KiCad offers the opportunity to create your own component and footprint but also to chance to include external libraries. I decided to go for the second choice and download what I needed from Mouser.it. The site offers a lot of different symbols, footprint and 3D models.



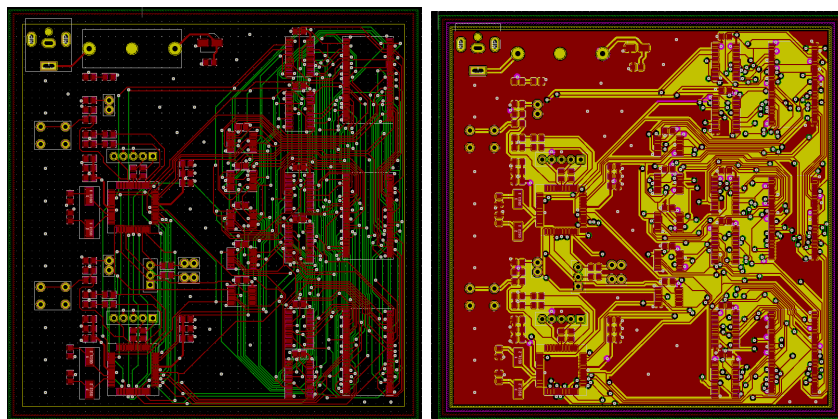
4 PCB layout

4.1 Footprint choice

At first, the idea was to go with as many through-hole components as possible, but since we decided to stay in a board with a dimension of 10 cm X 10 cm to have a lower cost, it turned out to be almost impossible. Once I changed the footprint of almost all components into SMD footprint we decided anyway to choose a fairly big size to allow a future hand soldering.

4.2 Routing

I choose to go for a 4 layer board with the top and bottom ones as signal layers and the inner two as power layers. Once I arranged all the ICs where I thought it was the best spot I placed all the critical components such as the crystal and the decoupling capacitors as close as possible to their corresponding pin. I used a thinner trace to connect signal pins and a bigger one to connect power supply pins to VDD. There wasn't so much space on the top layer so I was basically forced to use a lot of vias to jump from the top layer to the bottom one. The vias allowed me to connect fairly easily all the GND and VDD pins of the components to the two inner power planes. Finally to increase the immunity to noise and disturbances, to improve decupling, to shorter return paths and to enlarge the current-carrying capability I placed a GND pour all over the first and bottom layer.



4.3 3D view

Kicad offers the possibility to visualize in 3D what the final result should be. Since this project ends here, I particularly take care of this section by choosing all the 3D models of the components. This is how the complete board should look like:

