1. FMC signals from STM32 side

According to Table 261 on page 1595 of the Reference Manual, we need the following STM32 signals:

- FMC_CLK
- FMC_A[25:0]
- FMC_D[31:0]
- FMC_NE[1]
- FMC_NOE
- FMC_NWE
- FMC_NL (aka FMC_NADV)
- FMC_NWAIT

Signals FMC_NBL[3:0] are not needed, since we are going to use 32-bit access. Signals FMC_NE[4:2] are not needed, only one chip select FMC_NE[1] will be used by FPGA. In this configuration we need 32 data signals, 26 address signals, 6 control signals (64 lines total). To find out what signals map to what processor pins, refer to Table 10 on page 51 of the Datasheet.

2. FMC signals from FPGA side

According to expansion connector pinout on page 15 of the Novena PVT1 schematics, we have the following signals available:

- F_LVDS[15:0]
- F_LVDSA, F_LVDSB
- F_LVDSC
- F_LVDS_CK[1:0]
- DDC_SCL, DDC_SDA
- F_DX[8:0]
- F_DX[18:11]

LVDS pairs can be used as two independent single-ended signals. This gives us a total of 61 signals: 32 data, 6 control and 23 address lines.

3. STM32—FPGA signal mapping

I suggest, that we observe the following rules:

- Clock signal FMC_CLK should go somewhere in F_LVDS_CK[1:0]. These signals are connected to dedicated clock inputs in FPGA, we definitely want FMC_CLK to go into one of them.

- Bunnie's schematics has a warning box, that signals F_DX[14], F_DX[17], F_DX[18] are badly matched. I suggest, that we connect upper address bits FMC_A[22:20] to these badly matched lines. In case these lines do not work as expected, we will still have 20 lower working address lines, which is more than we currently have with EIM interface.

4. General considerations

STM32F429I-DISCO board uses LD3985M33R voltage regulator. It powers processor along with memory and display. Instead of memory and display we can use FTDI chip with USB connector. I think, we can use that same regulator or maybe something equivalent, if the original regulator is difficult to buy.

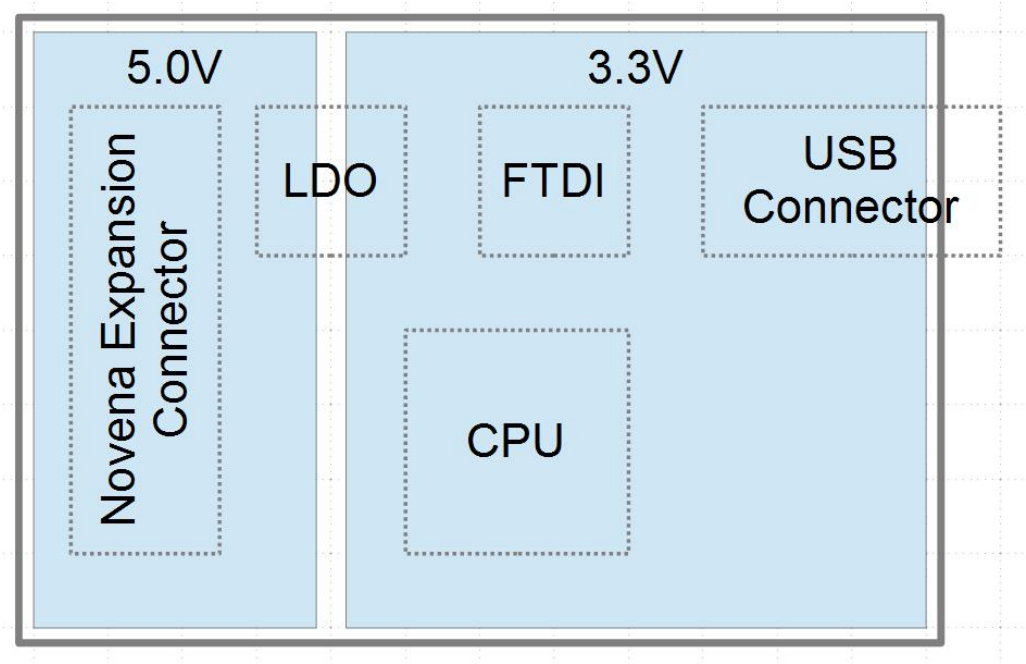
FT232R datasheet has good application circuit examples.

5. Suggested PCB design flow

1. Create new 4-layer board.

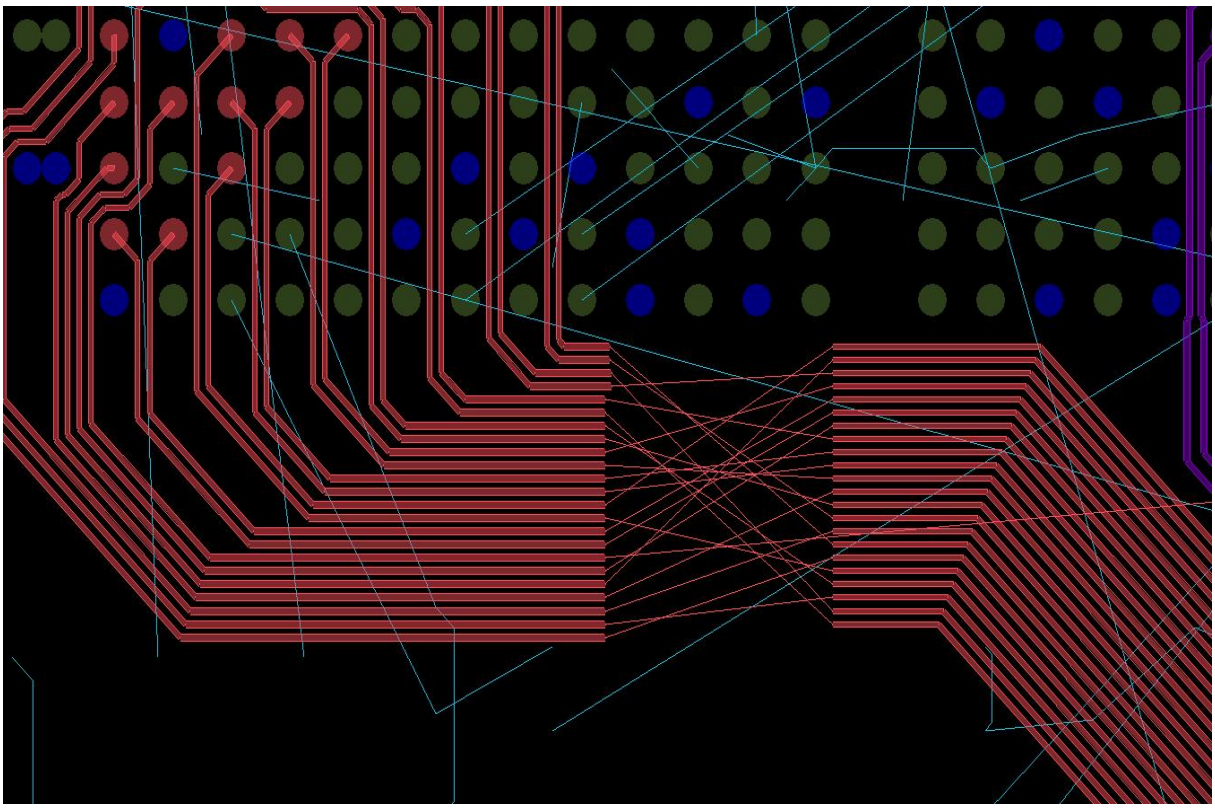
2. Fill one of the two internal layers with solid ground plane.

3. Fill the other internal layer with power polygons. I can suggest the following internal power plane layout. Please note, that this drawing is not to scale, you may end up with another arrangement after all.



4. Connect CPU pins to expansion connector. Try to observe rules from section 3) regarding clock signal and badly matched address lines. Connect all the remaining signals in some arbitrary order. It does not matter right now, what CPU pins go to what connector pins. Just make sure, that they are all connected and you can see their ratnets.

5. Use top and bottom layers to bring signals out of expansion connector and CPU. We need to connect around 60 signals, so try routing half of them in top layer and remaining half in bottom layer. Start routing from both ends (connector and CPU) towards the middle, but do not connect two ends, you should end up with something like the following pink bus.



6. Swap signals going into expansion connector as needed to avoid traces crossing each other. Try not to swap clock signal and three most significant address lines. All the other signals can be swapped, it does not matter to what FPGA pins they are connected.

7. Finish the layout and have fun!