Dad, can we build a computer?

DADSON-1

Prologue

“I want to build a computer for learning for kids. So all kids can learn and understand that making computer programs is not that easy. There should be one program installed that can fix 0’s and 1’s, and then there are some other programs and games there. Kids should use the main program to fix the applications, there is maybe only one 1 wrong in some place in the code. And then maybe there is one program that has no error, this is then as surprise.”

My son Andre at age of 11.

His dad at age 45 is home alone in the red house.

Andre is on summer vacation with his sister and mom.

Dad is home alone in a red house, watching “RED” on video.

Yes, it is not as is used to be, I have not build a computer in the last – 20 years?

Why? What excuses can I possible have to not build a computer with my son, for him and other children and all of us who want to discover things and do something exciting?

Because electronics and computers are not exciting anymore? This is not true.

Because nobody understands electronics and computers? This can be changed.

Because? No, there is no excuse.

I must do it.

Come with me, let’s go now!

We cannot build something that has success in the future without learning from the past – so let's take a quick look back. When I was young the most important you had to have to build a computer was – friends. And you had to travel as well. The best mix was to have friends around the world.

I still remember the first time I arrived in Minsk - two guys met me at underground with their wife's, both pregnant. They wanted to meet me, so they all did come – really nice welcome. Some years later I was able to source from them a set of multilayer PCB's for an IBM PC XT soviet made replica. Those PCB boards where really bad quality, no coating (plain copper), hard to solder, the drilled holes with very little metallization and so on. For the components I had to make many trips to Leningrad, Moscow, Riga and Minsk. The most famous black market for electronic components was in Leningrad, but similar ones existed in other cities as well. At one of such trips when I was trading NE555 clones made in Military IC fab in Riga on the IC market in Moscow I got lift offer to the train station. I ended up on the back seat of a new model of a black Volga car, with three man I had never seen before. Where does the trip go? To train station or somewhere else? We arrived at train station and I returned safely home with night train. But years later talking about this with an Major from soviet army (he was USSR champion in Fox hunting while I was as junior in Estonian national team), he said that the fact that those man did not take me in immediately actually means that they considered me as higher level threat (to be dealt with later).

Well, I managed to collect most of the IC needed to build that PC from the black market, except the memory IC. RAM was hard to get buy, it was manufactured in Minsk, and it was sometimes possible to get untested IC from the fab line. Those IC where in ceramic packages with truly gold plated pins and top metal housing – true military grade. Easier was to get un-separated RAM IC stripes extracted by the fab personnel just after the plastic molding process. All the "legs" of those IC had to be separated by own handwork using a self-made tool. After that we had to test the RAM IC's for bad bits, with self-made RAM tester.

For complete PC I had to assemble 4 boards:

- Central Processor: Intel 8088, original Intel, later Russian clones appeared also
- RAM Memory: 640K bytes
- Floppy disk controller
- VGA Video Card with MC6845 cloned IC from factory in Bulgaria

A few more trips to Minks and I had a full PC XT system with keyboard and two 5 inch Japan made floppy disk drives. Somewhere I got a color monitor also, I think.

Now I was ready to design a computer, all I needed was CAD software.

At that time the de-facto standard for PCB design was software sold under brand name "P-CAD". The only problem was that P-CAD had to be installed on hard disk. But I had none, my PC had two floppy drives only. That did not stop me, I searched for older version of P-CAD that had smaller memory footprint and was using overlays (a method when parts of application are loaded from hard disk). In order to run P-CAD I had lie to the operation system (MSDOS 3.x) that I have a hard disk:
SUBST C: B:

The above command basically tells the OS to use drive B (my second floppy disk drive) as first hard disk C. I installed P-CAD overlays to floppy in second drive (mounted as hard disk!) and started P-CAD from first floppy drive.

Using P-CAD on the PC I had built myself I design a single board computer compatible to Sinclair ZX Spectrum using components available on the black market, yes of course it was all done on the other side of the iron curtain. I finished that single board computer design and with some friends we manufactured a small series of them.

EBM – "Estonian Business Machines" was the proposed brand name we wanted to market the computers we did design and build. Without any marketing gods in our team we had little chance for financial success.

Nevertheless, we had fun by doing the things the hard way. I had to travel a lot, see places, and meet people. All because I was building computers!

What do I do today? As first thing I do not use P-CAD any more, I switched to Protel a long time ago. And a long time ago Protel did became Altium. And P-CAD became Altium as well.

So Altium is what I use today.

I do not need use ***subst C: B:*** to use P-CAD any more.

Today I use ***subst B: C:\data\altium*** for extra comfort with Altium Designer.

History is really repeating itself, only B and C have changed places.

Computer for kids

Ok, let's get serious, if we want to design something we need specifications, right? We need to know what the thing we are we are developing has to be able to-do.

So let's gather what we know for sure:

Learning

The kid's computer has to be able to learn new things right? This means it has to be possible to install new applications. We need some way to-do it. What options we have?

Flash Card/Cartridge

A lot of kids computers (and also portable game consoles) on the market use some custom cartridge that usually holds one game or education application on it.

Can we make a flash cart with a game on it? Andre has asked me several times. And each time I explain that those "game cartridges" are not sold as empty media, and that only validated publishing houses can release such items, and that it takes a lot of investment and so on. We have agreed that we will not go that way and use some media format that is a common standard, like SD card.

Price

We have to set a target price for customers, we have to calculate what price we can offer for distribution channel, and we have to calculate our self-cost.

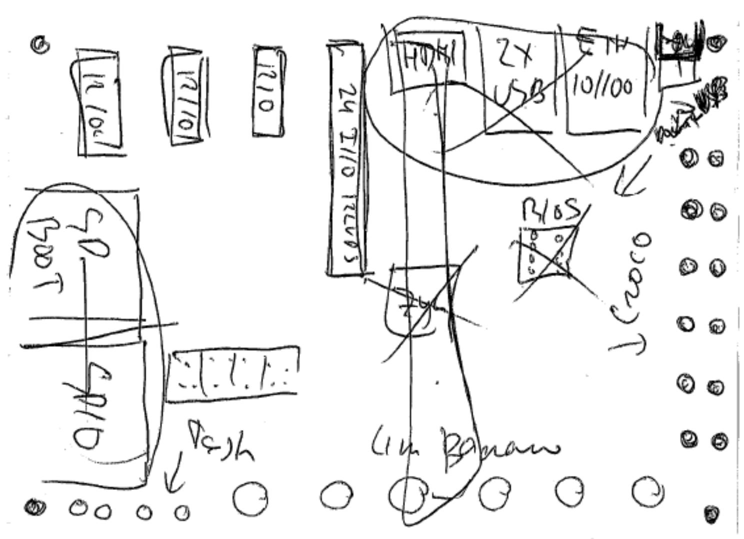
DIY factor

There are so many things today that are "just there" ready leaving nothing for the user to do. Our learn computer must offer at least something for the user also!

But is this possible at all considering the components used in modern electronic devices?

Size, take one

Big or small? How small is small enough? I am holding a blank sheet of A4 paper. Folding it once, A5 no this is too large. Folding it second time now I have A6 – a postcard size. This seems more like a good choice.



First sketch:

- Round holes for mounting in the corners.
- Back side right: HDMI, Dual USB, Ethernet 10/100, 5V Power plug, 4 slots for add-on cards
- Left side: Dual SD Card connectors
- Right side: dual metallized holes for “Crocodile clips”
- Front side: large holes for Banana connectors, some push buttons
- In the middle: central processor – Xilinx ZYNQ SoC, the right a DIP8 socket for BIOS

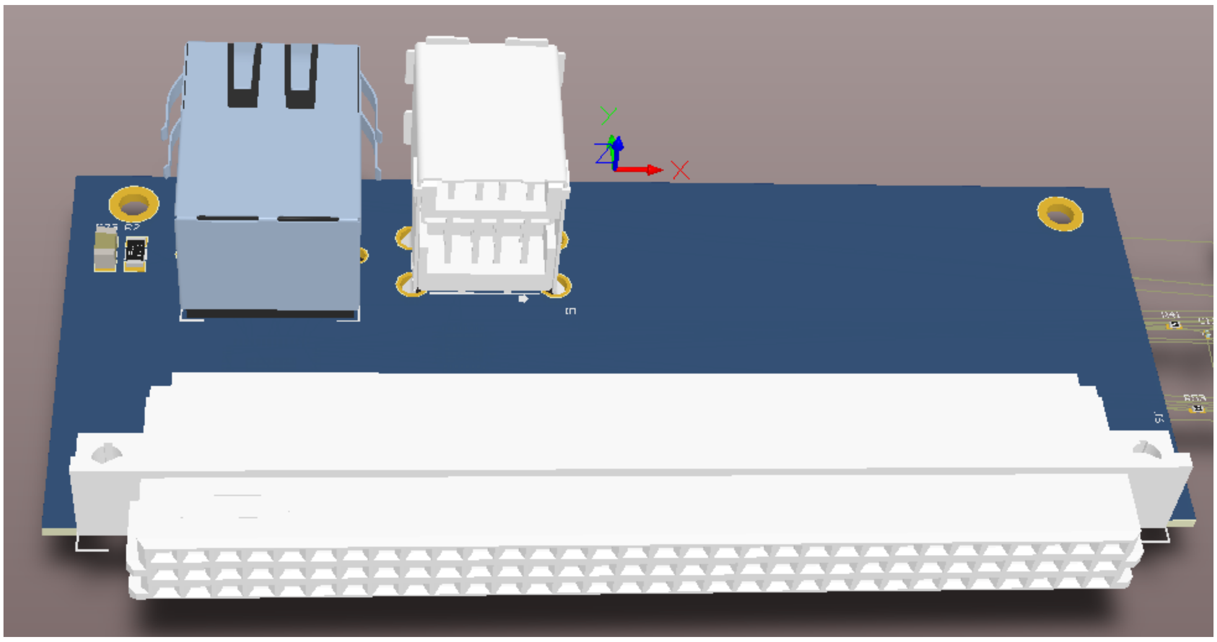
Is this it? Is it really? I am adding some place for 7 segment displays. What else?

This board could be produced below 99EUR for sure. We could sell it as kit, with all through-hole components separately for the dads and kids to solder.

The add-on board slots are also only holes in PCB so their manufacturing cost is 0.

This was first idea, to be abandoned and maybe to be reborn again.

Design Day 1:



DIN41612 slot connector, Gigabit Ethernet and dual stacked USB host connectors.

Initially I placed SD Card connector also on the top layer, but it takes so much place! The board would be much larger only for SD Card. So after some consideration SD Card goes away, this saves about 1.5 EUR of self-cost also. It makes more sense to have two USB connectors instead. We can leave out SD Card connector.

The idea is yes, it is – a computer on cartridge! Much similar like Intel Pentium II was at some time manufactured as a cartridge with the difference that Pentium Cartridge had only CPU and only single connector to the mainboard.

Well it's born – CoC: Computer on Cartridge! And got a new name: DADSON-1☺

Design Day 2: DADSON-1 specification

There are plenty of decisions to make, let’s decide!

Main Connector

I have played so many times with the idea to use PCB edge connector technologies. It has the advantage that CoC would not have a connector as physical components (Pentium II cartridge is PCB Edge type as example). There are problem though with PCB Edge, first the PCB coating should be “hard gold” not ENIG (Electroless Nickel Immersion Gold) like used in most PCB factories today as standard. Another problem are the connectors, first the old ones like used in ISA bus PC’s are rather large and ugly. The PCIe connectors from modern PC’s - are pretty nice. But using that type of connectors means we need special mechanical fixtures to keep the Cartridge steady as the PCB edge connector has low insertion force.

We could use just pin-header, but then there would be a problem with polarization and no this also not an option.

Or we could us DIN41612 Connector, also known as VME connector initially developed in Germany and named “VG Leiste”. This is how those connectors look like.

Let’s consider what we gain or lose when we select this type.

Price: cheap, so no pressure on the cost☺.

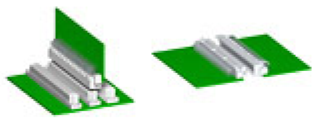
Connectors are polarized we cannot insert our CoC wrong way☺.

Total 96 pins in one connector that’s a lot already☺.

Easily available, many manufacturers☺.

Can be hand soldered, DIY Fun Factor☺.

Can be used as board to board, or board to backplane☺



Is it only ☺ for this connector that I can find? No, of course not – as this connector is fairly old, from the really old days it is not suitable for the super high speed modern communication protocols. Does it matter for us? No, not really. Besides we can achieve quite a good performance with it if we follow some rules while we define our CoC specification.

So done – decision is made: We use 96 pin (Type C) DIN41612 connector as CoC main connector.

CoC Pin-out

First thing to consider is if we can use some existing standard? To cut the long the story short, the answer is no. But that does not mean we should not look at available standards.

Quick look to Wikipedia VME bus P1 connector pin-out, what can we learn here? Power supply +5V spans 3 rows, this is something we should also do. In VME bus GND does not span 3 rows at no places, this is bad, and we change that in our specification. VME bus – SYSCLK is placed between GND signals that's good practice. System serial bus is placed in middle row B, this is bad. It surrounded by GND pins what is good we should do the same. So much for VME bus.

Let's make a better bus!

First we should place ALL important signals in the row A because that allows the use of cheapest 32 pin connectors (with limited function). This choice means we need to allocate all needed power and main signals in within 32 pins.

Row A

Power – we allocate one pin for +5V from row A. This gives us up to 2A current if only 32 pin connector is used. As we plan to run +5V in rows ABC, we would have up to 6A current with full connector, which gives us maximum of 30W power for the CoC. That is sufficient. It would be very hard to cool off more heat. It is very likely more that can be consumed, but having +5V span 3 rows makes the motherboard design simpler. With 32 pin connector the power would be limited to 10W, should be also more than needed in most cases.

Ground – there is never enough ground pins, but what is the minimum needed count? VME bus has total 8 GND pins. It would also nice to guard high speed signal pairs with GND on both sides, but then we would lose a lot of signal pins. So a compromise is needed. How about allocation in each row 3 GND pins and placing high speed signal pairs between them like this G-S-S-G-S-S-G?

Where should the ground pins be? Left, right or middle? As there is no way to choose right before left the obvious is middle☺. Actually there are other reasons to place the ground in the middle.

Where should he power pin be? Close to the ground or furthest away? VME bus has +5V at one edge far away from GND pins. Wisdom tells that we should not follow this and place power close to ground pins. This minimizes the high current loop area on PCB and length of power lines.

What special signals should we allocate in row A?

Would be good to have some sort of system management bus? Ok, done two pins allocated.

Reset Pin? Yes needed.

Interrupt/Alert pin? Better also to include it now.

One super-duper signal we define it later system pin? Absolutely.

What else can we decide now?

Would be nice to have a block of 16 signals in each row? Yes.

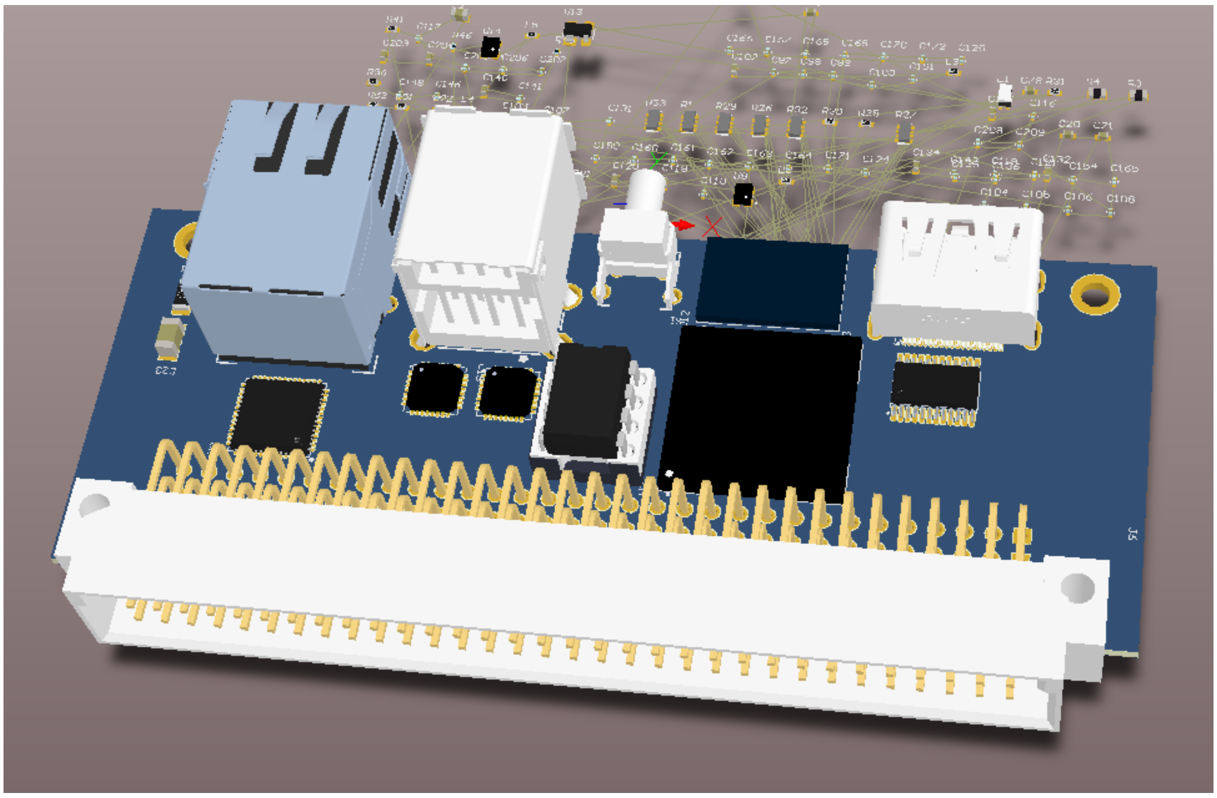
System serial console? Would also not hurt!

We can now almost finalize the pin-out. We could also just reserve some pins?

	A	B	C
1	IO	IO	IO
2	IO	IO	IO
3	IO	IO	IO
4	IO	IO	IO
5	IO	IO	IO
6	IO	IO	IO
7	IO	IO	IO
8	IO	IO	IO
9	IO	IO	IO
10	IO	IO	IO
11	IO	IO	IO
12	IO	IO	IO
13	IO	IO	IO
14	IO	IO	IO
15	IO	IO	IO
16	IO	IO	IO
17	**GND**	**GND**	**GND**
18	IO	IO	IO
19	IO	IO	IO
20	**GND**	**GND/sense**	**GND**
21	IO	IO	IO
22	IO	IO	IO
23	**GND**	**GND**	**GND**
24	Reset	Reserved	Reserved
25	**+5**	**+5**	**+5**
26	Super-Duper	Reserved	Reserved
27	Interrupt/Alert	IO	IO
28	UART RX or GPIO	IO	IO
29	UART TX or GPIO	IO	IO
30	**+3.3V**	IO	IO
31	SCL	IO	IO
32	SDA	IO	IO

Here is the pin-out based on the considerations above.

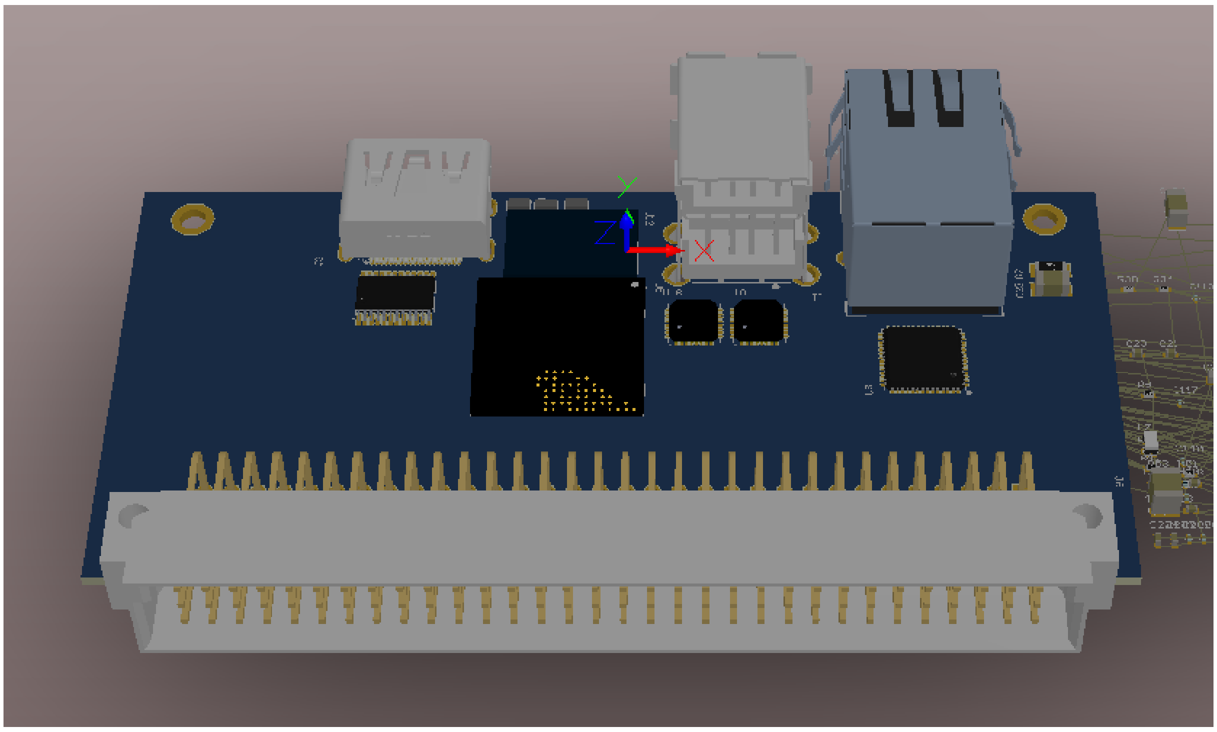
With this pin-out we have we have 72 universal I/O signals plus power and system management signals. Sounds good? I was initially hoping to have a bit more I/O signals in the connector, but when looking at the table it sounds good enough for me. We could consider using row B middle GND as connector sense pin, connecting to GND on the backplane and used sense pin on the CoC, yes good idea.



End of Day 2 of the active design phase. Board size is 100 x 40 mm. All main components are placed, the SoC, DDR3 memory, HDMI connector is added with some special protection for it, and in the middle there is socket for BIOS. Next to the Ethernet and USB are the “PHY” IC that are needed for the SoC. A push-button is also added.

No this cannot be done the board is too small for all the features planned. We need to add some IC for power supply, and some more small components. This cannot all fit when I plan to use standard PCB production technologies. So we have to revise the mechanical size specification.

Design Day 3



Connectors are rearranged for better routing. PCB size is changed from 100 x 40 to 100 x 50 mm.

How about debug and connection to host PC as USB device? We could use the SoC second USB as device port, or as OTG? But then, no it is better we do have keep two USB host ports. Maybe add FT2232H, channel A as Debug/JTAG and channel B as UART/Console? Well it does add almost 5 EUR to self-cost. Maybe FT232H, it's smaller and cheaper? But it only has one channel so we would lose the serial console feature. Could we use FT232H and small CPLD that multiplexes between JTAG and UART? No that not good either. But do we need high speed USB at all? Maybe we take the ultimate cheapest USB Microcontroller? Do we have space on PCB at all for this? I will have to decide on this later.

Another thing I messed up yesterday: in the pin-out table I allocated 3 x 16 = 48 pins as one block, but the SoC/FPGA devices have pins grouped in banks of 48! And we need the high speed pairs that are in-between our ground lines to be in the same bank. So we should allocate 12 + 2 + 2 in each row.

This means that we have more IO pins on the other end of the connector, those pins are allocated from another bank. Well, 48 is correct number for Xilinx, what about Altera? Quick look into datasheets tells me that Altera has also full bank size of 48 I/O. So our pin-out is compatible for both Altera and Xilinx devices. Good!

	A	B	C
1	IO	IO	IO
2	IO	IO	IO
3	IO	IO	IO
4	IO	IO	IO
5	IO	IO	IO
6	IO	IO	IO
7	IO	IO	IO
8	IO	IO	IO
9	IO	IO	IO
10	IO	IO	IO
11	IO	IO	IO
12	IO	IO	IO
13	**GND**	**GND**	**GND**
14	IO	IO	IO
15	IO	IO	IO
16	**GND**	**GND/sense**	**GND**
17	IO	IO	IO
18	IO	IO	IO
19	**GND**	**GND**	**GND**
20	Reset	Reserved	Reserved
21	+5	+5	+5
22	Super-Duper	Reserved	Reserved
23		IO	IO
24		IO	IO
25		IO	IO
26		IO	IO
27	Interrupt/Alert	IO	IO
28	UART RX or GPIO	IO	IO
29	UART TX or GPIO	IO	IO
30	+3.3V	IO	IO
31	SCL	IO	IO
32	SDA	IO	IO

Adjusted pin-out table

Meet the SoC

I have been so fast diving into the actual design that I have forgotten to explain the selection of the main component for our Computer.

Here you see a small module that is based on the same device I will be using for the computer on cartridge design right now. The largest rectangle you see is the main processor. But the processor is not only a processor it has in the same package also a "playground – sandbox". This is the reason why this chip has so much more play factor as anything else available today. Each time we turn power on, the "playground" is rearranged to become something we want to play with. Ok, this is very high level explanation. But this programmable playground called FPGA fabric is really something very exciting that makes our Computer different. The pins in the pin-out table marked IO, those pins can have different function as needed. At power up, they are nothing, only when our computer boots and starts executing programs it will configure those IO pins as needed. The best thing is that we do not need to be worry about what may be needed, as all we need to know, it can be done, by implementing a new hardware in the FPGA fabric.

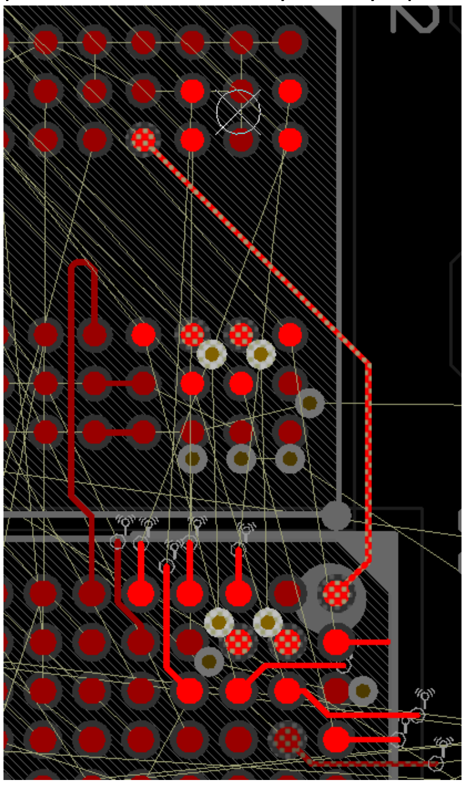
Why Xilinx – well Xilinx was the first to be on the market with this kind of device, and as of today the availability is much better than of competing devices from Altera. So there is no option as of today, a Zynq it is going to be!

So what can this Zynq do? First of all it has two ARM Cortex A9 Processors. Their maximum speed is about 600MHz what is lower than that of the ARM cores in modern mobile phones. But let's compare Zynq with my first PC (the one I did build myself):

Feature	The first PC I had	Zynq CoC we are designing
Processor	Intel 8088	Dual ARM Cortex A9
Processor Clock	8MHz	600Mhz (each processor core)
RAM Memory	0.64 MByte	512 Mbyte
Removable media	2 x 1.2 Mbyte floppy	USB storage, 4+ Terabyte?
Network	None	Gigabit Ethernet

I think we are good with the Zynq as main device for our computer.

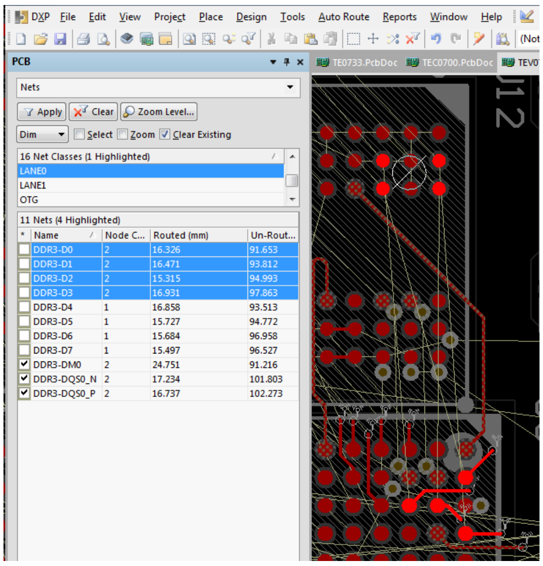
I cannot wait to see if I can get things routed with the placement so I try first the most complicated part, the DDR3 memory. The Zynq we use in 0.8mm pitch BGA package so is the DDR3 ram IC.



Routed are Data group signals that cannot be swapped DM0, DM1. Lane 0 differential clocks have to be routed in the inner or bottom layers, so I added fanout via's. Data bits within one lane can be swapped. I routed them out of the BGA boundary, and made initial swap so 4 data bits go at one side and the rest from other side of the non-swappable DM0 signal. Of course we must calculate the wire lengths in the Zynq package itself, this is done by running a special user script within Altium.

Next task – wire length matching. When I was struggling with DDR3 RAM on TE0720 micro module my son Andre did come to help me. So he can really say that he helped dad with that complex designs. It is really like a little puzzle game, you need to move the wires until they all have same length.

Let's see how we manage this time! If you look at the picture above you can instantly see that there will be mismatch between 4 data lines that go from the right compared to those more in the middle. Right, that is why we first match the 4 longest wires, and then proceed to the shorter ones. In the hope we can make the short wires longer to match the overall length. We do not care about data strobes at this time, those go in another layer and we can match the length without any issues.



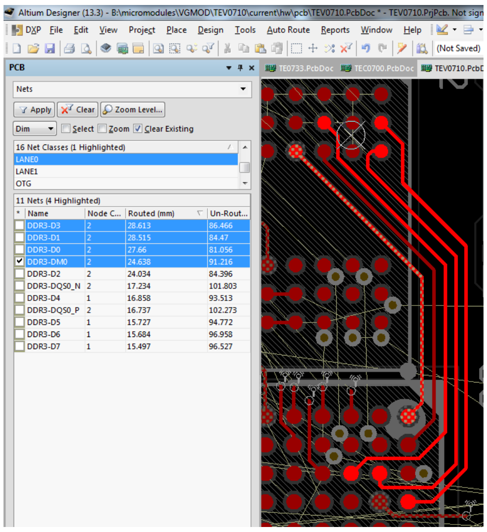
16 Net Classes (1 Highlighted)

- LANE0
- LANE1
- OTG

11 Nets (4 Highlighted)

*	Name	Node C...	Routed (mm)	Un-Rout...
☐	DDR3-D0	2	16.326	91.653
☐	DDR3-D1	2	16.471	93.812
☐	DDR3-D2	2	15.315	94.993
☐	DDR3-D3	2	16.931	97.863
☐	DDR3-D4	1	16.858	93.513
☐	DDR3-D5	1	15.727	94.772
☐	DDR3-D6	1	15.684	96.958
☐	DDR3-D7	1	15.497	96.527
✓	DDR3-DM0	2	24.751	91.216
✓	DDR3-DQS0_N	2	17.234	101.803
✓	DDR3-DQS0_P	2	16.737	102.273

Highlighted are those signals that will be longest, now first we connect them the way they attach easiest, and check their lengths. If all is good, we can proceed to next signals. If not we can look if we can make adjustments by swapping some signals.


Altium Designer (13.3) - B:\micromodules\VGMOD\TEV0710\current\hw\pcb\TEV0710.PcbDoc * - TEV0710.PrjPcb. Not sign
DXP File Edit View Project Place Design Tools Auto Route Reports Window Help
(Not Saved)
PCB
TE0733.PcbDoc
TEC0700.PcbDoc
TEV0710.PcbD
Nets
Apply
Clear
Zoom Level...
Dim
Select
Zoom
Clear Existing

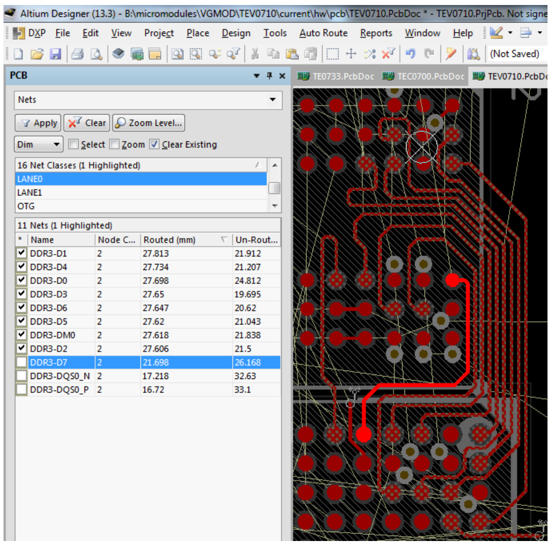
16 Net Classes (1 Highlighted)

- LANE0
- LANE1
- OTG

11 Nets (4 Highlighted)

*	Name	Node C...	Routed (mm)	Un-Rout...
☐	DDR3-D3	2	28.613	86.466
☐	DDR3-D1	2	28.515	84.47
☐	DDR3-D0	2	27.66	81.056
✓	DDR3-DM0	2	24.638	91.216
☐	DDR3-D2	2	24.034	84.396
☐	DDR3-DQS0_N	2	17.234	101.803
☐	DDR3-D4	1	16.858	93.513
☐	DDR3-DQS0_P	2	16.737	102.273
☐	DDR3-D5	1	15.727	94.772
☐	DDR3-D6	1	15.684	96.958
☐	DDR3-D7	1	15.497	96.527

I think this time the first placement is ok, I have already adjusted the two longest signals to same length. Now we have 7 more signals to go in this memory lane.


Altium Designer (13.3) - B:\micromodules\VGMOD\TEV0710\current\hw\pcb\TEV0710.PcbDoc * - TEV0710.PrjPcb. Not signe
DXP File Edit View Project Place Design Tools Auto Route Reports Window Help
(Not Saved)
PCB
TE0733.PcbDoc
TEC0700.PcbDoc
TEV0710.PcbD
Nets
Apply
Clear
Zoom Level...
Dim
Select
Zoom
Clear Existing

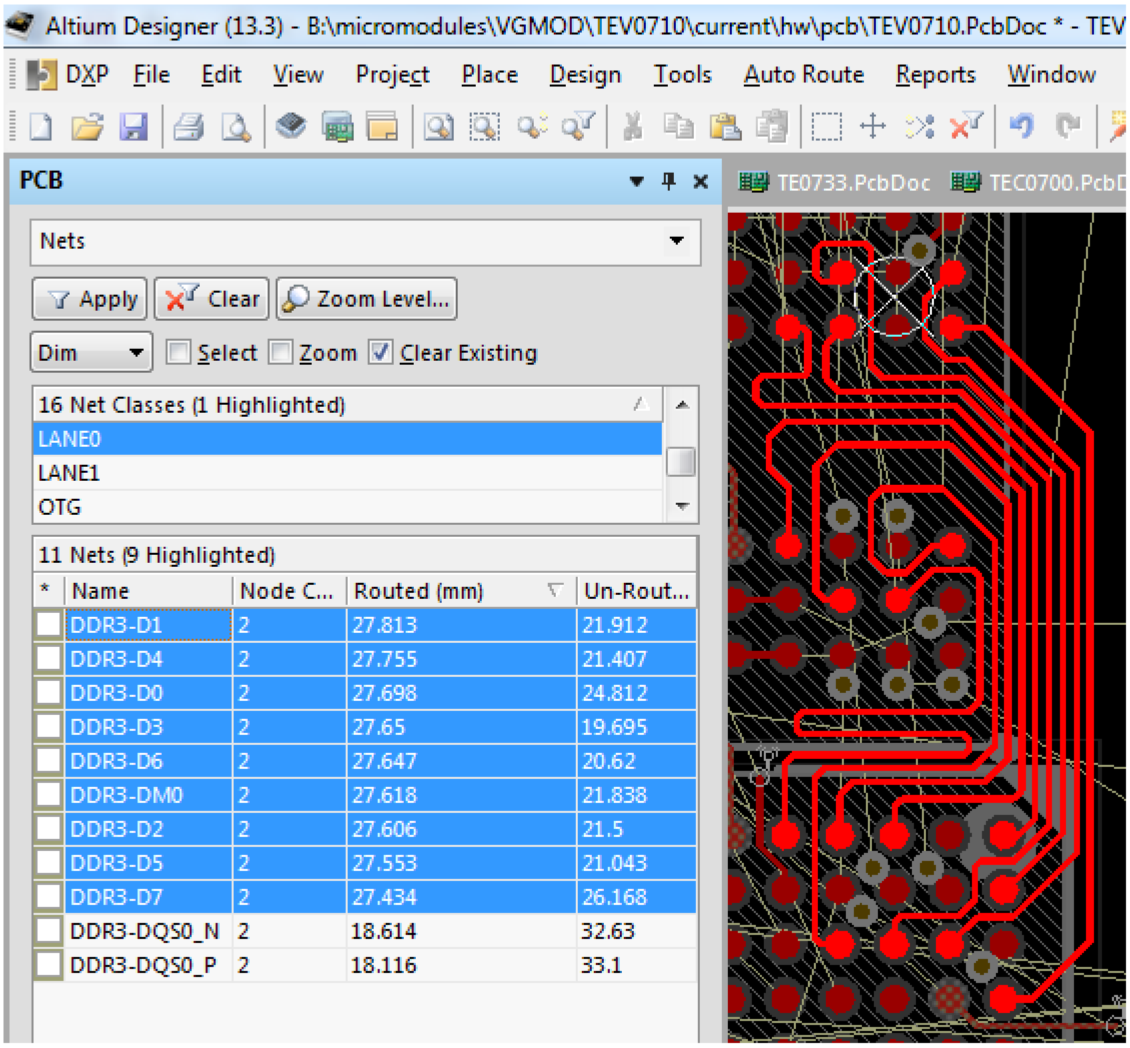
16 Net Classes (1 Highlighted)

- LANE0
- LANE1
- OTG

11 Nets (1 Highlighted)

*	Name	Node C...	Routed (mm)	Un-Rout...
✓	DDR3-D1	2	27.813	21.912
✓	DDR3-D4	2	27.734	21.207
✓	DDR3-D0	2	27.698	24.812
✓	DDR3-D3	2	27.65	19.695
✓	DDR3-D6	2	27.647	20.62
✓	DDR3-D5	2	27.62	21.043
✓	DDR3-DM0	2	27.618	21.838
✓	DDR3-D2	2	27.606	21.5
☐	DDR3-D7	2	21.698	26.168
☐	DDR3-DQS0_N	2	17.218	32.63
☐	DDR3-DQS0_P	2	16.72	33.1

We are almost done here, after some small swapping and some trickery we see that all signals in this group are matched except one (highlighted). We also see that we can use some zigzag to match the last connection length also. One memory lane (that is 8 bits a byte) are done completely in top layer. This is very good as we now have more chance to succeed with the rest of critical memory routing as well.


Altium Designer (13.3) - B:\micromodules\VGMOD\TEV0710\current\hw\pcb\TEV0710.PcbDoc * - TEV

Nets

Dim | Select [ ] | Zoom [ ] | Clear Existing [✓]

16 Net Classes (1 Highlighted)

LANE0
LANE1
OTG

11 Nets (9 Highlighted)

*	Name	Node C...	Routed (mm)	Un-Rout...
	DDR3-D1	2	27.813	21.912
	DDR3-D4	2	27.755	21.407
	DDR3-D0	2	27.698	24.812
	DDR3-D3	2	27.65	19.695
	DDR3-D6	2	27.647	20.62
	DDR3-DM0	2	27.618	21.838
	DDR3-D2	2	27.606	21.5
	DDR3-D5	2	27.553	21.043
	DDR3-D7	2	27.434	26.168
	DDR3-DQS0_N	2	18.614	32.63
	DDR3-DQS0_P	2	18.116	33.1

And voila: first timing group routed and matched! All on top layer (except data strobes that should be on different layer anyway). There was one more swap needed.

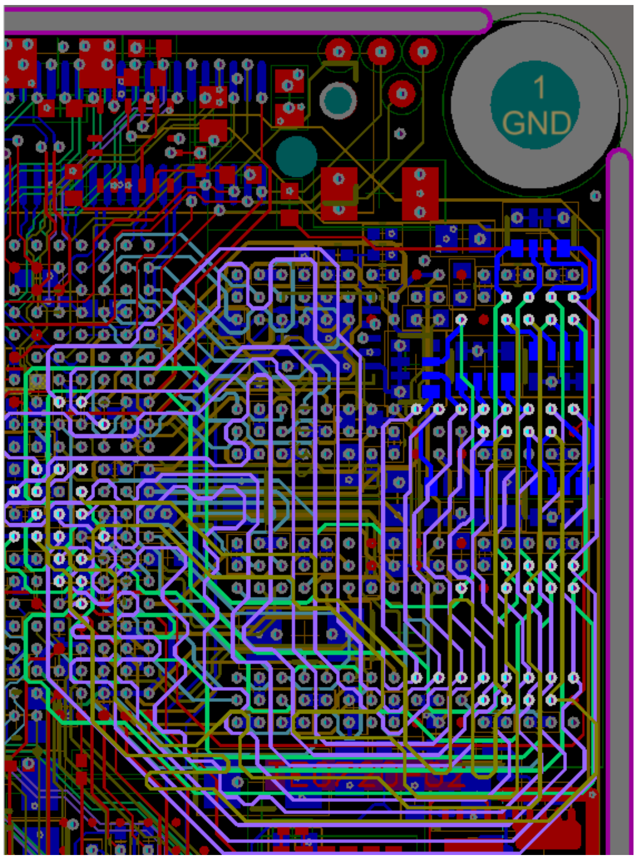
Design night 4

Not much happening, I route second memory lane and try to get Memory address and command group routed. Coming home late so kids are a sleep. My wife tells me that Andre has made some drawing from a computer he wants to be made.

Design Day 5

After breakfast I look at his drawing

"Andre, remember you helped my with DDR3 memory routing for the modules? I need similar help for the learning computer design!" I say to my son.

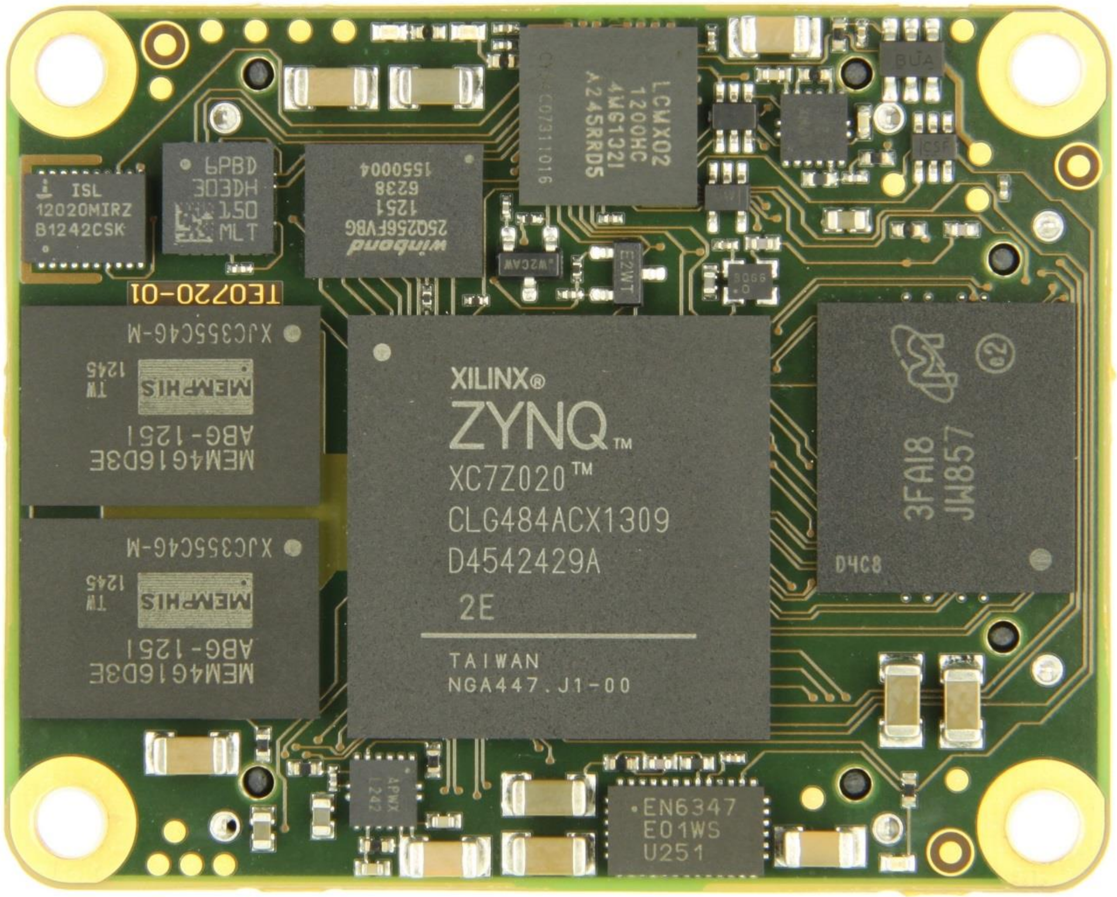


This is the part where did work a little bit together doing the length matching for DDR3 addresses.

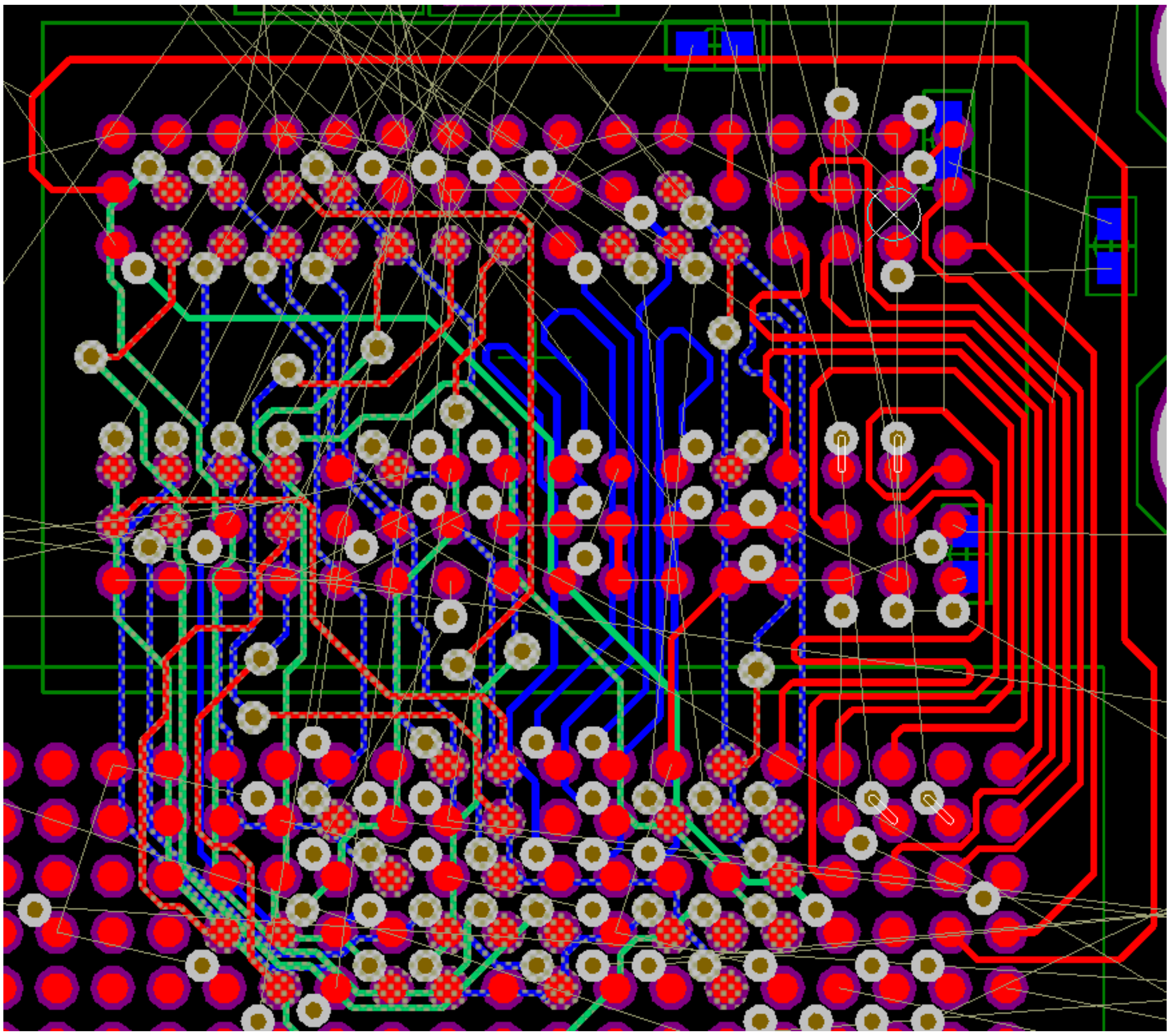
"Well can't you just use that module for the learning computer?" my son asks me.

"Yes, of course we could, TE0720 module has all that is needed and would be very easy to use also. So it is an option of course. The only problem is that it would make the price for the learning computer too high" I explain.

"We should ask Thorsten to make only one unit first before we go production, so we can find our mistakes and fix them", my son adds. Yes making prototypes is sure a good idea, I second to this.

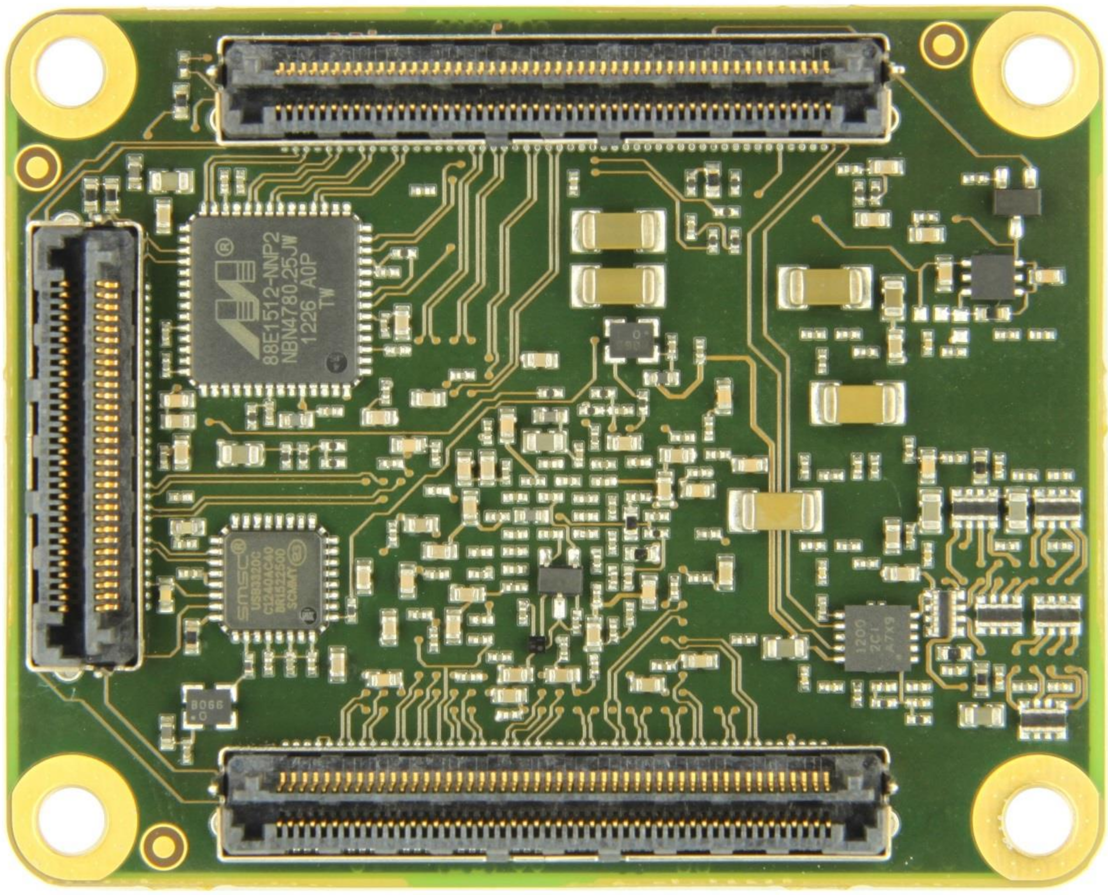


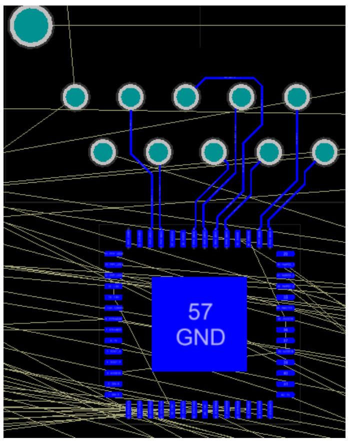
TE0720 close up view from top. Links are two memory chips. We can optimize one away for cost. In the middle is Zynq the Processor and do it all super thing, we can use smaller package, again saving a little in total cost. To the right is embedded MMC, which works like an SD Card on board, we do not this part to be on our board. The upside down chip with marking “Winbond” is 32MByte Flash, for this we use a DIP8 socket, so we can change our BIOS easy, also we save a little cost here too as we do not need that much of flash memory.



Here we have routed the DDR3 memory, Andre helped me a little to find last routes.

Well after routing the address command group it is clear the length matching is going to be hard or impossible! Well we can move the DDR3 chip bit further away to have more playground. To be done later…

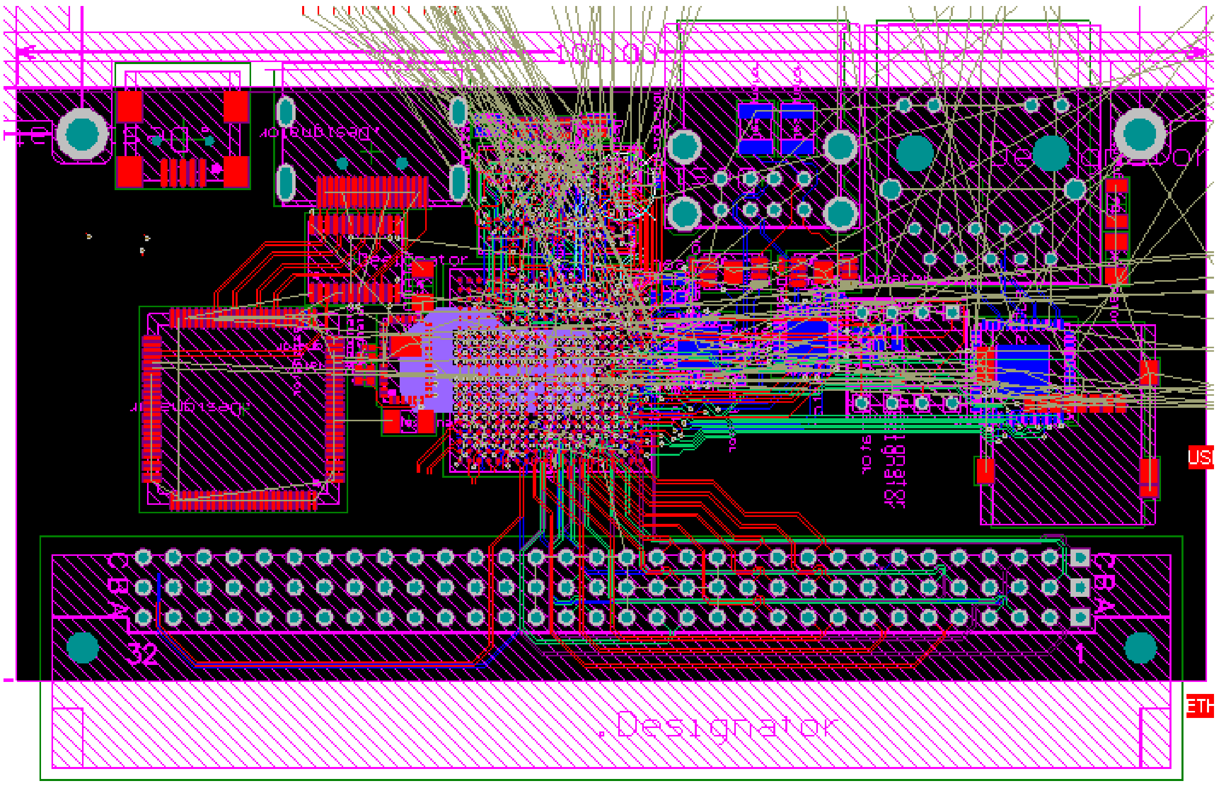

Bottom view of TE0720, the larger chip is Marvell Gigabit Ethernet PHY.



And here we have routed it on our computer cartridge! Well those are connections to the RJ45 connector, the remaining connections have to be made.

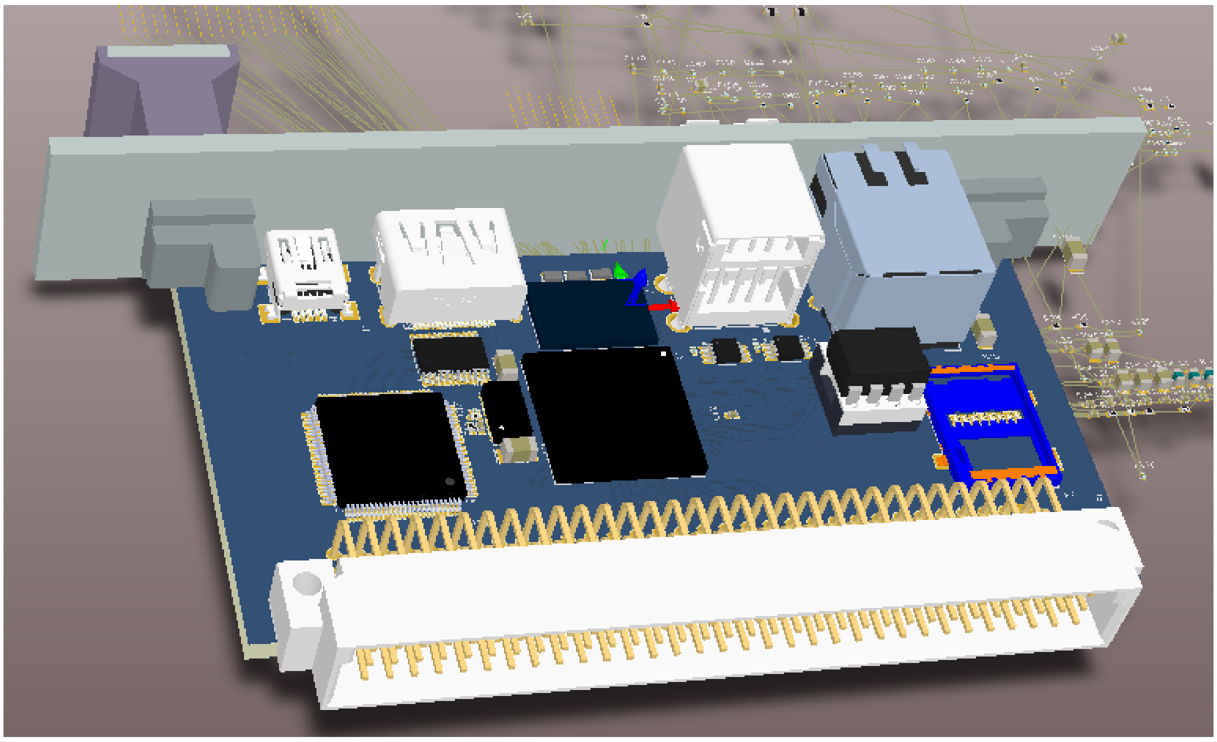
Design Night 5+1, vacation with kids

Kids are asleep or at least quiet, so I start up my old notebook that I had secretly packed into my backpack.



And this is result of a few late hours of work. Routed are both USB PHY’s, Ethernet PHY and 24 LVDS pairs (that is 48 IO pins from bank B35) from Zynq to CoC main connector. DDR3 is routed with data lines matching completed. HDMI high speed lines are routed from HDMI transmitter to ESD protection and to HDMI Connector. BIOS chip is also routed and wires are routed close to the micro SD Connector (CLK, CMD and DAT0). Zynq Core voltage regulator is placed and routed as well.

There are still some black unused areas of PCB space, what can we add there? We should not increase the cost, but if there is something that we can add, to increase the functionality we still can do it.



Top 3D view, USB and Ethernet PHY IC are placed at the bottom so they do not show.

Feature compare

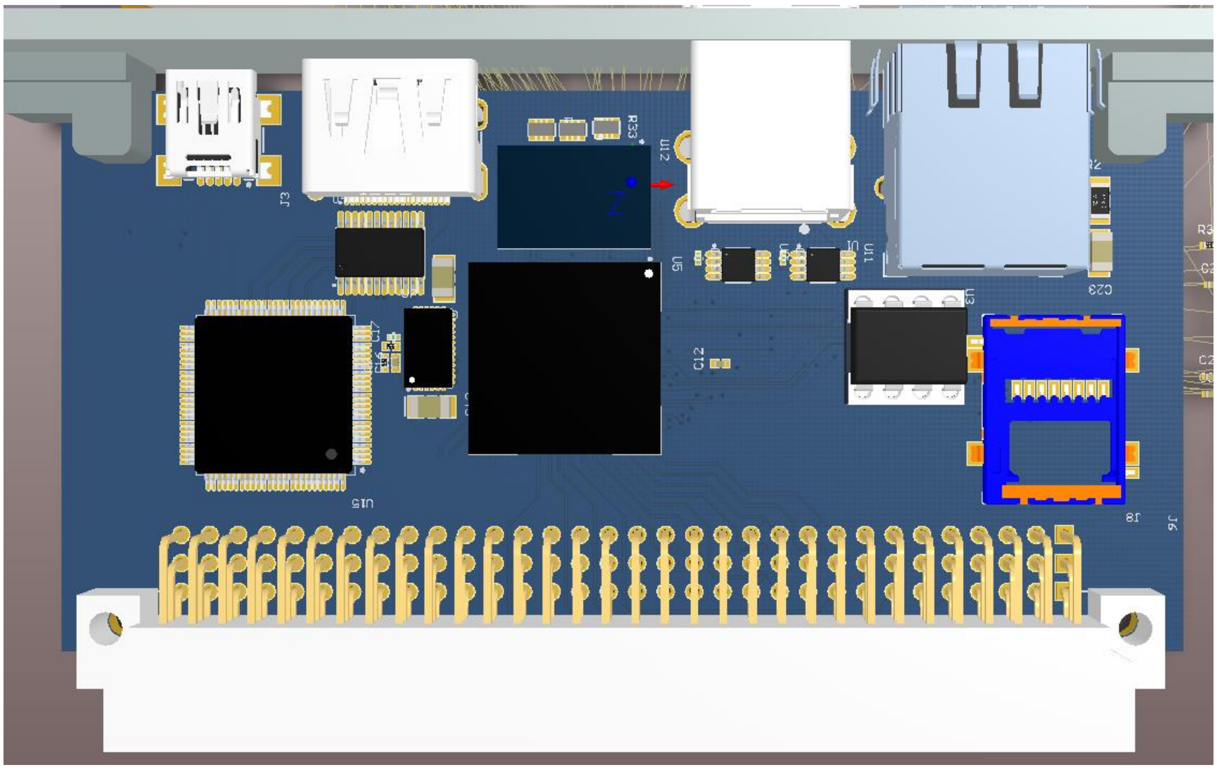
It always makes sense to compare the features with similar products already on the market, so let's compare with MicroZED.

Feature	**MicroZed**	**DADSON-1**
SoC	XC7C010	XC7C010
RAM	**1 GByte**	0.5 GByte
SPI Flash (boot rom)	8 MByte, fixed	Socketed BIOS (up to 8+ MByte)
Micro SD Card	Yes, bootable	Yes, bootable with BIOS support
Ethernet 10/100/1G RJ45	Yes	Yes
USB 2.0	1 OTG *1	**2 Host**
USB Overcurrent sense	No	**Yes, separate for both ports**
HDMI	No	**Yes, Analog Devices HDMI 1.4**
USB UART	Yes, micro USB, SiLabs CP210x	**Yes, mini USB, FTDI USB 2.0 HS**
User I/O PL FPGA	**100**	64
Pmod PS MIO	Vertical mount	No
Pmod Header PL IO	No	**Yes, Vertical or Right angle**
User Connector	2 x 100 Pin Micro Header	1x 96 Pin Connector (VG96)
FAN header	3 Pin	?
ZYNQ JTAG	14 Pin Header only	**Onboard USB JTAG**
ARM independent JTAG	Pmod slot PS MIO	Pmod slot PL I/O
CPU Clock Source	33.3333MHz	33.3333MHz
User LED (PS MIO)	Yes	Yes
Done LED (also PL LED)	Yes	Yes
User Push Button	Yes	?
Reset Button	Yes	?
Size (PCB only)	101,6 x 57,15mm	100 x 50 mm
Price EUR	150 (based on 199 USD price)	**99**

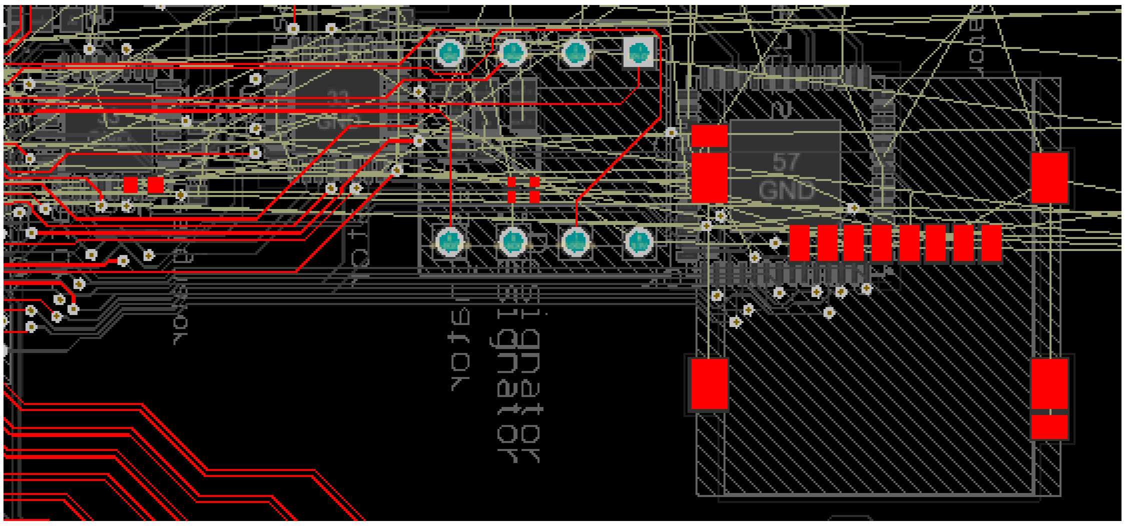
*1 MicroZED says it has USB OTG Support but the connector is standard USB host connector, so full OTG support is not possible, this is misleading and wrong information. My son Andre asked me to tell that to all, that the competition is using misleading information in marketing materials to get advantage in sales.

Andre's commentary on SiLabs USB UART: if you need to install special drivers to enable the seri al port, it may be that kids get so frustrated they do not want to do anything anymore with this thing!

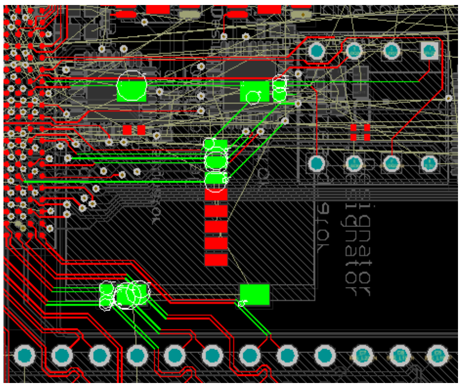
Now let's look what we have so far and what if at all we can change or add, there has been a few days pause, it usually helps see things from new perspective.



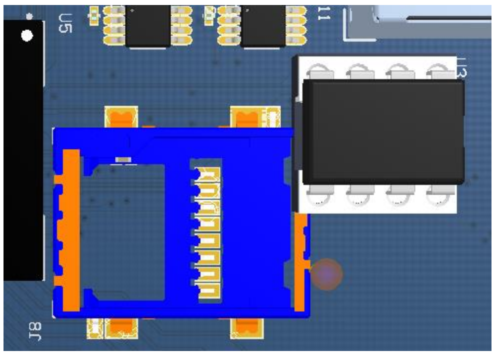
3D rendered top view, as you see there is large unused area between the main IC, connector and SD card slot. What could we place there? Unfortunately nothing what would make lots of sense, we could place LED's or Push-buttons there, yes. Can we use this space better? Look at the micro SD Card Connector at the right, it is top loader type so it could be anywhere on the board. If we could move the SD Card to the empty space more in the middle, we could "win" space at the PCB edge what is more "valuable". Is it doable?



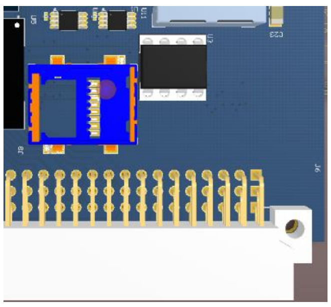
Before moving SD Card. It may be possible, let's move it and look again.



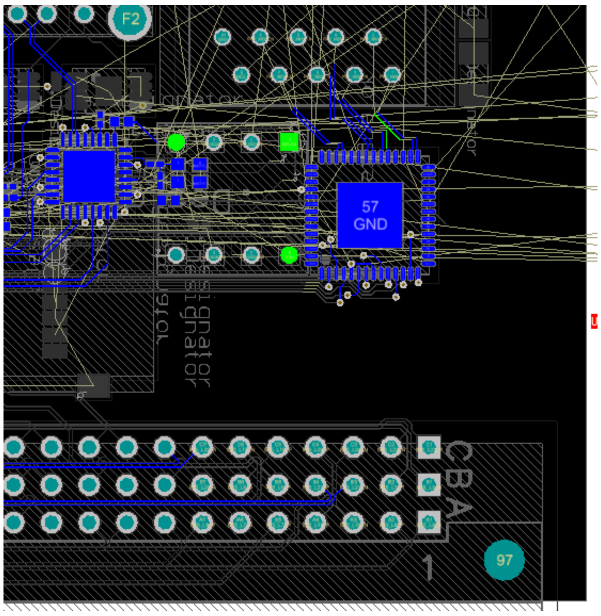
Lime color are where traces conflict, well it does not look that bad, observe that all the red wires can easily be moved to go around the SD card slot pads.



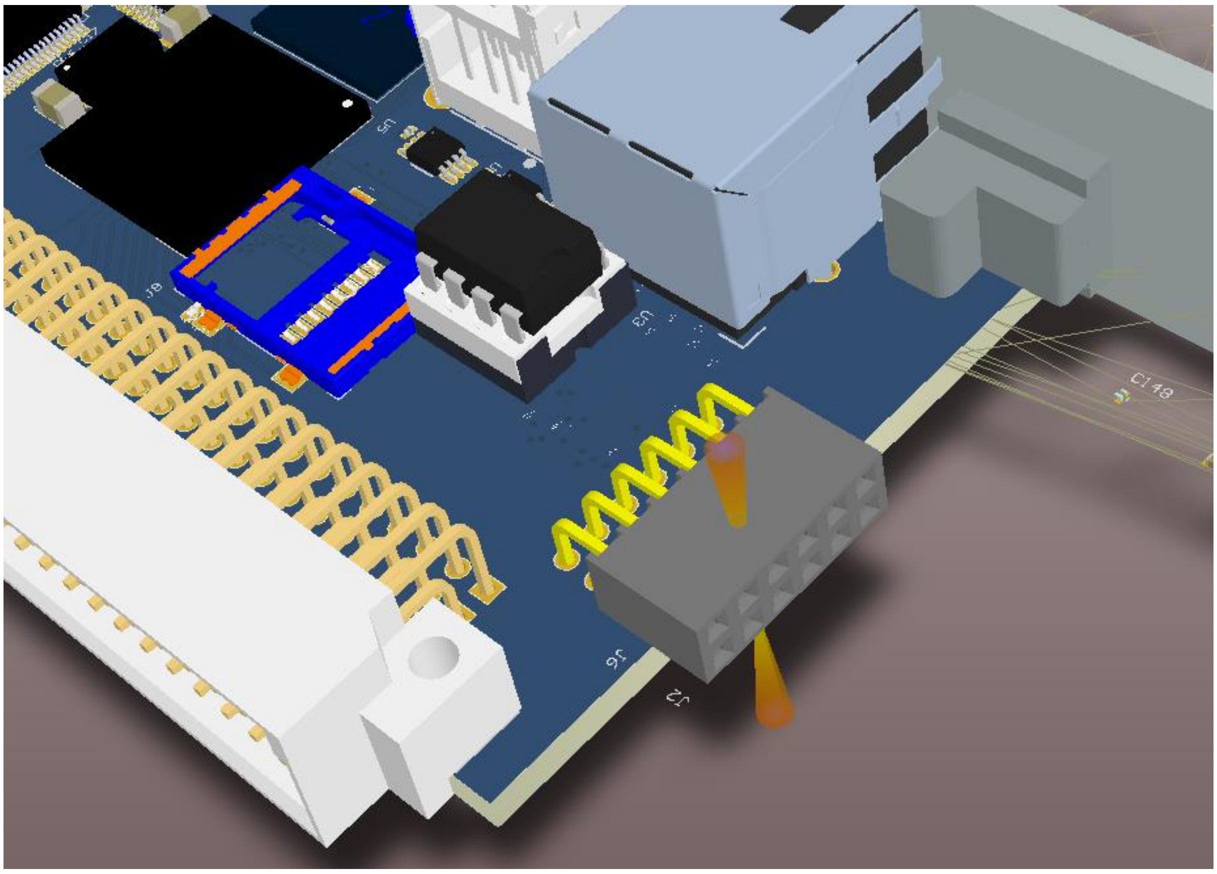
Looking in 3D, now we clearly see our BIOS socket is overlapping the Micro SD Card slot. But we only need to move the BIOS a few millimeters what again forces us to move the Ethernet PHY IC in bottom layer a little bit as well. Luckily all those changes do not make any complications for routing.



BIOS moved, we now have large empty area at the right side of board.



Now Ethernet PHY IC also moved, we have our space free, and now we have real problem, what should we do with it?



We added one PMOD slot, it could be left un-mounted or populated with straight or right angle header. And we have not added any cost, as in unassembled version it's only 12 extra holes in PCB and nothing more. This feature is added to the table now ☺

Courage

Well, I pressed “go live” button today. Decision was made a few days earlier to actually make the campaign open. Not at all satisfied with the online video and textual description but I felt that it had to be done now.

So how fare is the design? Over a week same, why? Yes I have been thinking about what is yet to be done. There are some considerations and choices to be taken. I think it’s time to make some compromises. I usually try to make things to good, this makes the doing very hard, same now – the main memory IC is placed very close to the processor. I have managed to route it, but there are length mismatches in the address group.

Thank you Sundance!

First funding at Indiegogo, balance is above 0, that’s always good.

Sundance Multiprocessor Technology
●●● embedded signal processing solutions

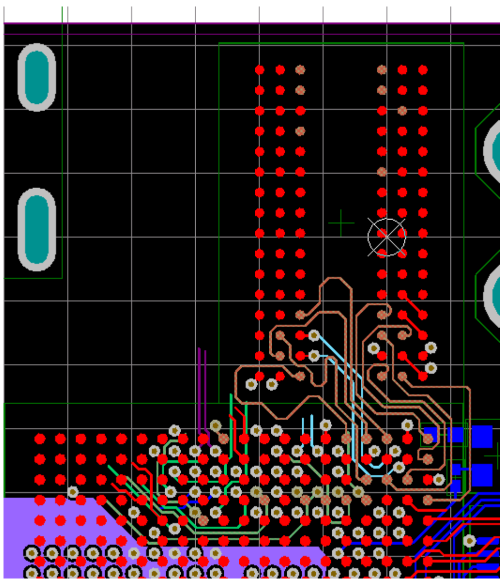
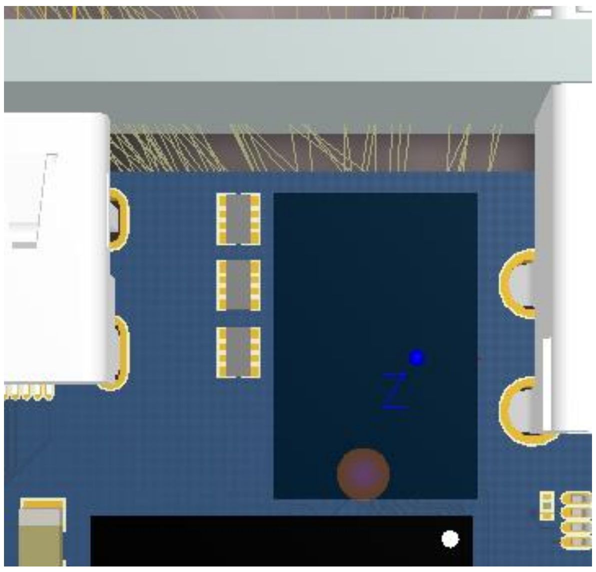
It’s not much, but as thank you I can include the logo here ☺

Thank you!

More courage, a 90 degree turn!

Today I made big change, I turned the DDR3 RAM IC 90 degrees. I was hesitating to-do it for many days. Why? Why was it necessary if at all and why did I delay with it?

I had first placed the DDR IC as close as possible to the main SoC chip, I had made almost all routing, made length matching also, well mostly. It looked promising enough that I had hope to get it properly routed completely. But, I had some trouble with address lines, there was very hard to make some of them as long as needed, there was no place for that. And all the time I did know that 90 degree turned IC would eliminate those problems. So why not making that turn? Because it was almost done the old way? It looked nice? Why? Because I did think it would make the PCB larger? Yes, for some reason I did think that when rotating the IC 90 degrees I would have to make the complete PCB board a few mm. Also I did not want to trash away the work I had done.

But look now, the RAM IC is turned 90 degrees. And it fits the same old PCB, no need to change the board outline. And with this placement the address lines coming from the SoC have some travel space before they reach the RAM, this can be used to make the length matching.

Maybe the main reason I delayed with this change was that I had to admit to myself that I failed with the routing, and that my initial placement was not good. It's always hard to admit own mistakes. Even if it's not a mistake. OK, after this change I can leave the remaining DDR3 routing for later, I am truly positive that there will be no issues getting it done.

With this change and decision made, I can proceed to finish the rest.

Hardware/Adapters

Floppy Disc Emulator

vvv

Optical Disc Emulator

Lots of old game consoles use some non-standard optical disc drive (ODD) as storage medium. Unfortunately those disc drives are very often broken and cannot be repaired or replaced. For such cases CoC can replace those optical drives.

Gamecube ODD Emulator

CoC is electrically compatible to the disc interface of Gamecube so the adapter is merely wires only type, 5V power is also directly available.

Wii ODD Emulator

CoC is electrically compatible to the disc interface of Wii so the adapter is merely wires only type, 5V power is also directly available.

Appendix A: DIY Kit content

Through-hole components for soldering

Item		Vendor	
1	Dual USB connector	Würth	
2	RJ45 Magjack with LED's	Würth	
3	R/A Button	Würth	
4	DIN 41612 Connector 96	Harting	
5	DIP8 socket		
6	1M Byte Flash BIOS DIP8	Winbond	
7	Pin header		

Glossary