
Dokumentace programu Horizon-EDA

Vydání 1.0

Lukas K.

19.01.2021

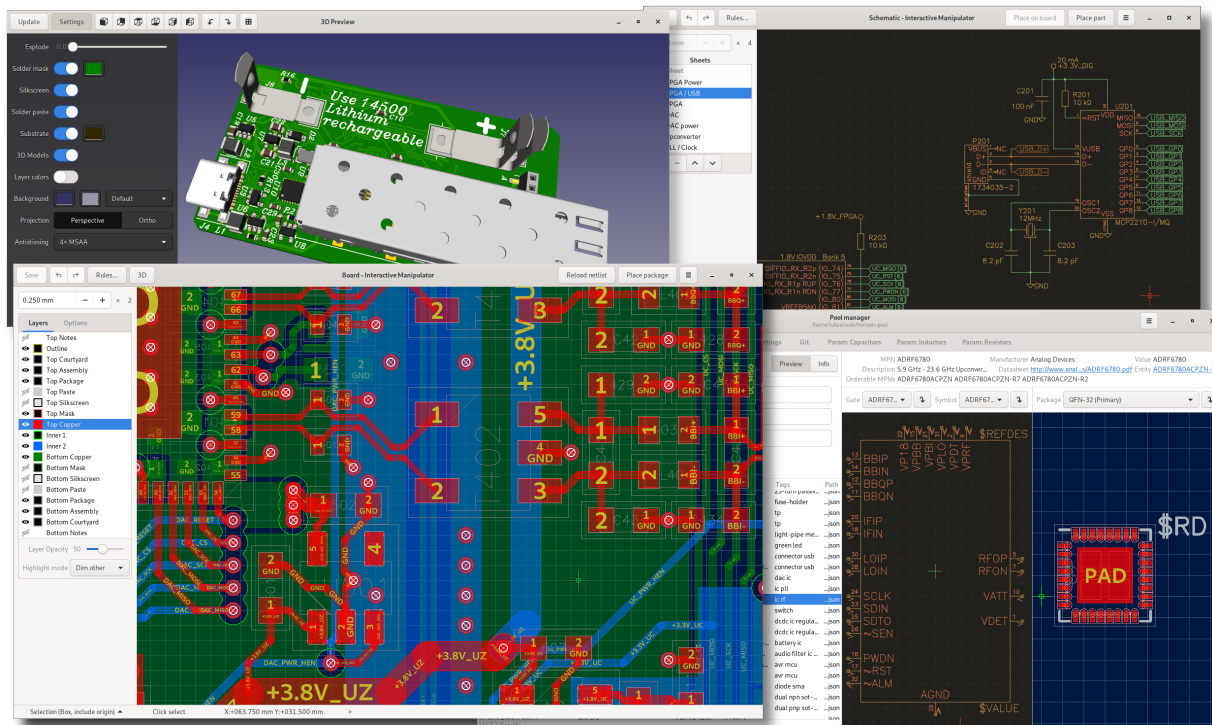
Obsah:

1	Proč další program typu EDA (Electronic Design Automation)	3
2	Hlavní výhody	5
3	Cíle	23
4	Instalace	25
5	Nastavení fondu knihoven	27
6	Vytvořte nový projekt	29
7	Kreslení schémat	33
8	Vytvořte desku	39
9	Příklad projektu	45
10	Příkazy	47
11	Mezerníková nabídka	49
12	Mřížka	51
13	Zadávaní numerických hodnot	53
14	Kreslení	55
15	Vybírání	59
16	Posouvání a manipulační značky	61
17	Hladiny (Layers)	63
18	Editor schémat	65
19	Editor desky plošných spojů DPS (PCB)	69
20	Kopírování rozmístění a umístění	71

21 Zpětné úpravy (Backanotation)	73
22 Pravidla	75
23 Proč fond knihovny	77
24 Elementy součástí	81
25 Správce fondu (Pool manager)	83
26 Vytvoření pouzdra součástky	87
27 Contribute to the Pool	89
28 Sestavení (kompilace) programu na operačním systému Windows	91
29 Sestavení (kompilace) programu na operačním systému Linux	93
30 Parametry programu	95
31 Popis funkce	99
32 Použití příkazového řádku (CLI Command Line Interface)	101
33 Modul python	103
Rejstřík	105

Horizon EDA je aplikace **Electronic Design Automation** podporující integrovaný pracovní postup pro návrh desek plošných spojů včetně správy součástí a vstupního schematu.

Podívejte se na dokumentaci programu Horizon EDA *Hlavní výhody* nebo přímo *Zkuste* nebo začněte *číst od začátku*.



Proč další program typu EDA (Electronic Design Automation)

Možná se divíte, proč jsem začal s horizontem EDA v roce 2016, když v té době již existoval funkční program KiCad.

Začněme s citací od Toma Hausherra: [PCB Design Perfection Starts in the CAD Library](#)

[...] Perfektní návrh PCB začíná v CAD knihovně

To platí také pro EDA software, kde editor schémat / DPS bude jen tak dobrý jak dobrá bude struktura knihovny. Od definice položek knihovny až po symboly, pouzdra, které jsou základem každého EDA softwaru, změna těchto definic je téměř nemožná bez významné změny téměř ve všech ostatních částí aplikace. Určitá struktura knihovny také vede další vývoj této aplikace EDA.

Používáním KiCadu pro malé a střední projekty, byly mé nejvíce bolestivé body v knihovně postrádající koncept objednatelných součástek bez duplikování symbolu pro každou součástku a schématický editor, který neví o spojích. Zatímco editor desek v KiCadu je docela dobrý jako nástroj pro návrh spojů chyběly expresivní pravidla návrhu. Zejména první a druhý bod se nedal snadno vyřešit bez velkých změn, které by následovalo mnoho diskusí, protože by to byly zlomové změny.

To mě přimělo začít přemýšlet, jak bych navrhl nástroj EDA, který by vyhověl mému přání a byl dostatečně snadný na implementaci od nuly jako show jednoho muže a umožnil splnění snu a experimentování.

Jádrem těchto myšlenek bylo udržování schématu a seznamu spojů (Netlistu) jako reprezentace samostatného návrhu, aby bylo možné použít neschematicky založené postupy takové propojovací tabulky. To vedlo k rozhodnutí definovat vývody a jejich směr signálu v něčem co se nazývá Jednotka (Unit), nikoli ve Značce (Symbolu), který je běžný u mnoha jiných EDA aplikací. Je možné mít více symbolů představujících stejnou součástku (například rezistor) bez jakéhokoli vlivu na seznam spojů. Kromě jména a směru signálu, vývod, jak je definován v jednotce, může mít také více alternativních názvů pro určení více funkcí vývodu, jak jsou obvykle k dispozici na mikroprocesorech (MCU) a programovatelných hradlových polích (FPGA).

Pro definici seznamu spojů (Netlistů) reprezentujících skutečnou součástku jsou určeny jednotky (Units) odkazující na něco, čemu se říká entita (Entity). Tento odkaz se nazývá brána (Gate). U jednoduché části, entita odkazuje pouze na jednu jednotku (Unit), která zahrnuje všechny vývody. Pro některé součástky má smysl mít více bran.

Součástky, které obsahují více instancí stejné funkce, jako například čtyřnásobný operační zesilovač pak budou odkazovat na kromě 4krát operační zesilovací jednotku také na jednotku pro napájení.

Na plošném spoji ze strany součástek jsou pouzdra definována jako do značné míry v každé aplikaci EDA - pájecí obrazce a grafické položky jako popisky, referenční označení a montážní obrys. Pájecí obrazce jsou definovány pá-

ječími místy popisujícím měděnou plošku, pájecí masku a další vrstvy v jejich tvarech a mnohoúhelnících. To velmi usnadňuje pájecí místa správného tvaru, která mohou být kreslena tak, jak jsou, bez různých obcházení, jako je použití více obrazců, které tvoří jeden skutečný obrazec. Pro každou velikost pájecího obrazce jsou pájecí místa obvykle doprovázeny krátkým skriptem napsaným ve vlastním jazyce založeném na zásobníku, abyste mohli upravit jejich velikost a také další vlastnosti, jako je poloměr rohu nebo rozšíření pájecí masky.

Pájecí místa (Pads) pouzdra (Package) jsou mapovány na vývody (piny) definované v jednotkách (Units), na které odkazuje entita v něčem co nazýváme součástka (Part). Aby bylo možné mapovat na něco objednatelné, díly mají pole pro jméno výrobce a číslo dílu výrobce (MPN) mimo jiné podrobnosti, jako je datový list odkaz na internet nebo popis.

Všechny výše uvedené odkazy mezi položkami (například entity odkazující na jednotky) jsou podle jednoznačného identifikačního klíče UUID. V Horizon EDA mají všechny přiřazené položky neměnné UUID v době jejich vytvoření. Toto UUID pak používají další položky k odkazování na tuto položku.

Na závěr tohoto úvodu:

Moje největší slabost je v tom, že nakonec obrátím libovolný elektronický projekt do výmluvy pro psaní EDA softwaru (a naopak).

(Převzato z: <https://twitter.com/mycoliza/status/824809235632447492>)

Dále: *Hlavní výhody*

2.1 Interaktivní a jednoduchý management součástí

Snadná správa součástí, pouzder součástek a symbolů schemat pomocí správce fondu knihovny součástí (*Správce fondu, Co je fond knihovny?*):

The screenshot displays the 'Pool manager' application window. The top menu bar includes 'Units', 'Symbols', 'Entities', 'Padstacks', 'Packages', 'Parts', 'Frames', 'Settings', 'Git', 'Param: Capacitors', 'Param: Inductors', and 'Param: Resistors'. The 'Parts' tab is active, showing a search interface with fields for MPN, Manufacturer, and Description, and a 'Tags' button. Below the search fields is a table of components:

MPN	Manufacturer	Description	Package	Tags	Path
3517	Keystone	5mm fuse clip	3517	fuse-holder	...
5000	Keystone	Miniature TH...	5000	tp	...
5001	Keystone	Miniature TH...	5000	tp	...
7511B11	VCC	Light Pipe, 3 ...	7511B11	light-pipe me...	...
150060G575000	Würth Elektronik	LED green cle...	0603 LED	green led	...
1734035-2	TE Connectivity	Mini USB-B r...	1734035-2	connector usb	...
10118194-0001LF	Amphenol FC1	10118194-00...	10118194-00...	connector usb	...
AD9122BCPZ	Analog Devices	Dual 16-Bit, 1...	CP-72-7	dac ic	...
ADF4356BCPZ	Analog Devices	6.8 GHz frac...	QFN-32	ic pll	...
ADRF6780	Analog Devices	5.9 GHz - 23...	QFN-32	ic rf	...
AP2280-1WG-7	Diodes Inc.	Single Chann...	SOT-23-5	switch	...
AP3015	Diodes Inc.	micro power ...	SOT-23-5	dc/dc ic regula...	...
AP3015AKTR-G1	Diodes Inc.	micro power ...	SOT-23-5	dc/dc ic regula...	...
AP9211	Diodes Inc.	Single chip Li...	U-DFN2030...	battery ic	...
AS3320	Alfa	Voltage Contr...	DIP18, 7.62 ...	audio filter ic
ATtiny5 DIP	Microchip	AVR MCU wit...	DIP8, 7.62 m...	avr mcu	...
ATtiny5 SOIC	Microchip	AVR MCU wit...	SOIC-8	avr mcu	...
B340A-13-F	Diodes Inc.	3.0A surface ...	SMA	diode sma	...
BC846AS	Diodes Incorporated	Dual NPN sm...	SOT-363	dual npn sot...	...
BC856AS	Diodes Incorporated	Dual PNP Sm...	SOT-363	dual pnp sot...	...
RI-M18A-C615M1D	Murata	Ferrite bead...	C0603

The right side of the interface shows a schematic diagram of the ADRF6780 component. It includes a pinout table with labels like BBIP, BBIN, BBQP, BBQN, IFIP, IFIN, LOIP, LOIN, RFOP, RFON, VATT, VDET, SCLK, SDIN, SDTO, ~SEN, PWDN, ~RST, ~ALM, AGND, and \$VALUE. A red square labeled 'PAD' is shown on the right, representing a padstack or footprint.

Přirazení vývodů k pájecím ploškám v editoru součástí (Part editor):

Save

Part Editor

×

MPN

ADRF6780

Inherit

orderable MPNs

ADRF6780ACPZN, ADRF6780ACPZN-R7, ADRF...

Value

Inherit

Manufacturer

Analog Devices

Inherit

Datasheet

http://www.analog.com/media/en/techn

Inherit

Description

5.9 GHz - 23.6 GHz Upconverter

Inherit

Tags

ic x rf x

+

Inherited:

Inherit

Base part

none

Entity

ADRF6780ACPZN-R7 / Analog Devices

Package

QFN-32

Change

3D Model

QFN-32-1EP_5x5mm_P0.5mm.step

Inherit

Parametric data

Table

None

Gate	Pin	Mapped
Main	AGND	✓
Main	BBIN	✓
Main	BBIP	✓
Main	BBQN	✓
Main	BBQP	✓
Main	IFIN	✓
Main	IFIP	✓
Main	LOIN	✓
Main	LOIP	✓
Main	PWDN	✓
Main	RFON	✓
Main	RFOP	✓
Main	SCLK	✓
Main	SDIN	✓
Main	SDTO	✓
Main	VATT	✓
Main	VDET	✓
Main	VP18	✓
Main	VPBB	✓
Main	VPBI	✓
Main	VPDT	✓
Main	VPLO	✓
Main	VPRF	✓
Main	~ALM	✓
Main	~RST	✓
Main	~SEN	✓

0 pins not mapped

Map

Unmap

Auto map pads

Select pads

Select pin

Copy from other part

Pad	Gate	Pin
1	Main	VDET
2	Main	VPDT
3	Main	VPRF
4	Main	AGND
5	Main	RFOP
6	Main	AGND
7	Main	RFON
8	Main	AGND
9	Main	VPRF
10	Main	VATT
11	Main	BBQN
12	Main	BBQP
13	Main	BBIP
14	Main	BBIN
15	Main	VPBB
16	Main	PWDN
17	Main	~RST
18	Main	IFIN
19	Main	AGND
20	Main	IFIP
21	Main	VPBI
22	Main	VP18
23	Main	SDIN
24	Main	SCLK
25	Main	SDTO
26	Main	~SEN
27	Main	VPLO
28	Main	LOIN
29	Main	AGND
30	Main	LOIP
31	Main	VPLO
32	Main	~ALM
PAD	Main	AGND

0 pads not mapped

2.2 Snadné vytváření součástí

Jednoduše přidejte vývody tak, jak jsou uvedeny v katalogovém listu součástky:

Pin	Name	Direction	Alt. Names	Gate
1	RESET	Input		Main
2	D	Input		Main
3	CLK	Input		Main
4	~CLK	Input		Main
5	Vee	Power Input		Main
6	~Q	Output		Main
7	Q	Output		Main
8	Vcc	Power Input		Main

Horizon-EDA se dodává také se šablonami skriptů pro import průmyslového standardu ve formátech jako je IBIS, který vám ušetří zdlouhavou práci při opisování toho, co je v katalogovém listu.

2.3 Snadná tvorba pouzder součástek

Získejte náskok při vytváření pouzder součástek výběrem z více než 20 přednastavených pájecích obrazců dle normy IPC.

Generate
IPC-7351B
Dual
Single
Quad
Grid

CHIPARRAY
LEDSC
MELF
MOLDED
CAPAE
SO
SOD
SODFL
SOJ
QFP
SON
QFN
PSON
PQFN
BGA
SOT223
SOT23
DIP
SIP
PGA

Small outline transistor (JEDEC TO-236AB)

Body

Length D2.900 mm
Width E11.300 mm
Style3 pins5 pins

Lead

Pitch e0.950 mm
Length L0.365 mm0.235 mm
Width b0.420 mm0.120 mm
Span E2.370 mm0.270 mm

Calc

DensityMostNominalLeast
Fabrication0.100 mm
Placement0.100 mm

Generate

Round-offNone0.01 mm0.02 mm0.05 mm
PadstackRectangularRounded rectangular

Lead span

Distance measured from lead termination to lead termination.

☒ Auto fit

as per IPC-7351B, Table 3-2
Flat Ribbon L and Gull-Wing Leads (greater than 0.625 mm pitch)

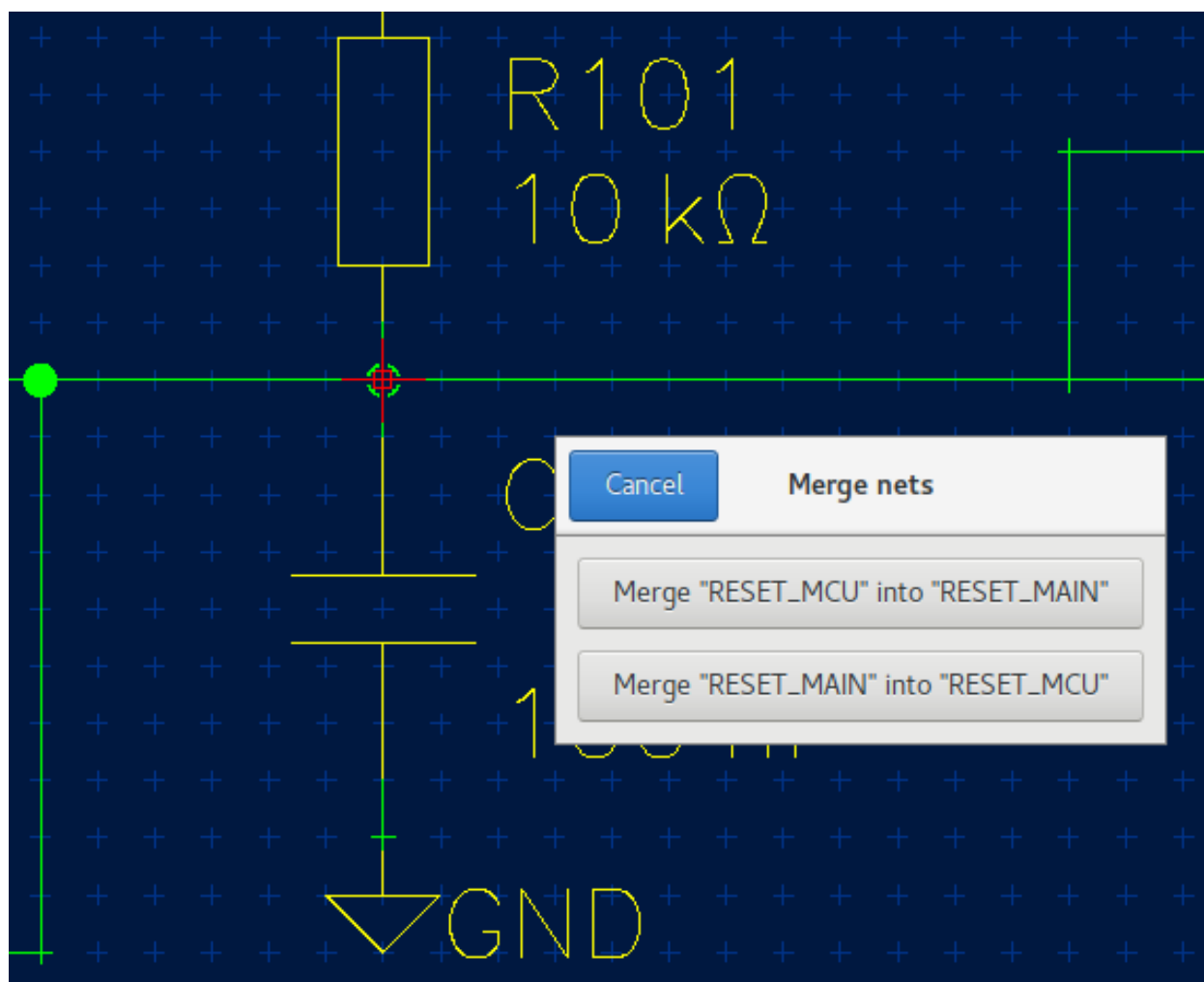
Podporován je také import pájecích obrazců z KiCadu.

2.4 Snadný pro začátečníky i pokročilé uživatele

Stačí stisknout mezerník a zobrazí se seznam všech příkazů, které můžete provést. Tyto příkazy lze spouštět přizpůsobitelnými klávesovými zkratkami pro jednotlivé klávesy nebo různými kombinacemi kláves podobnými editoru Vim.

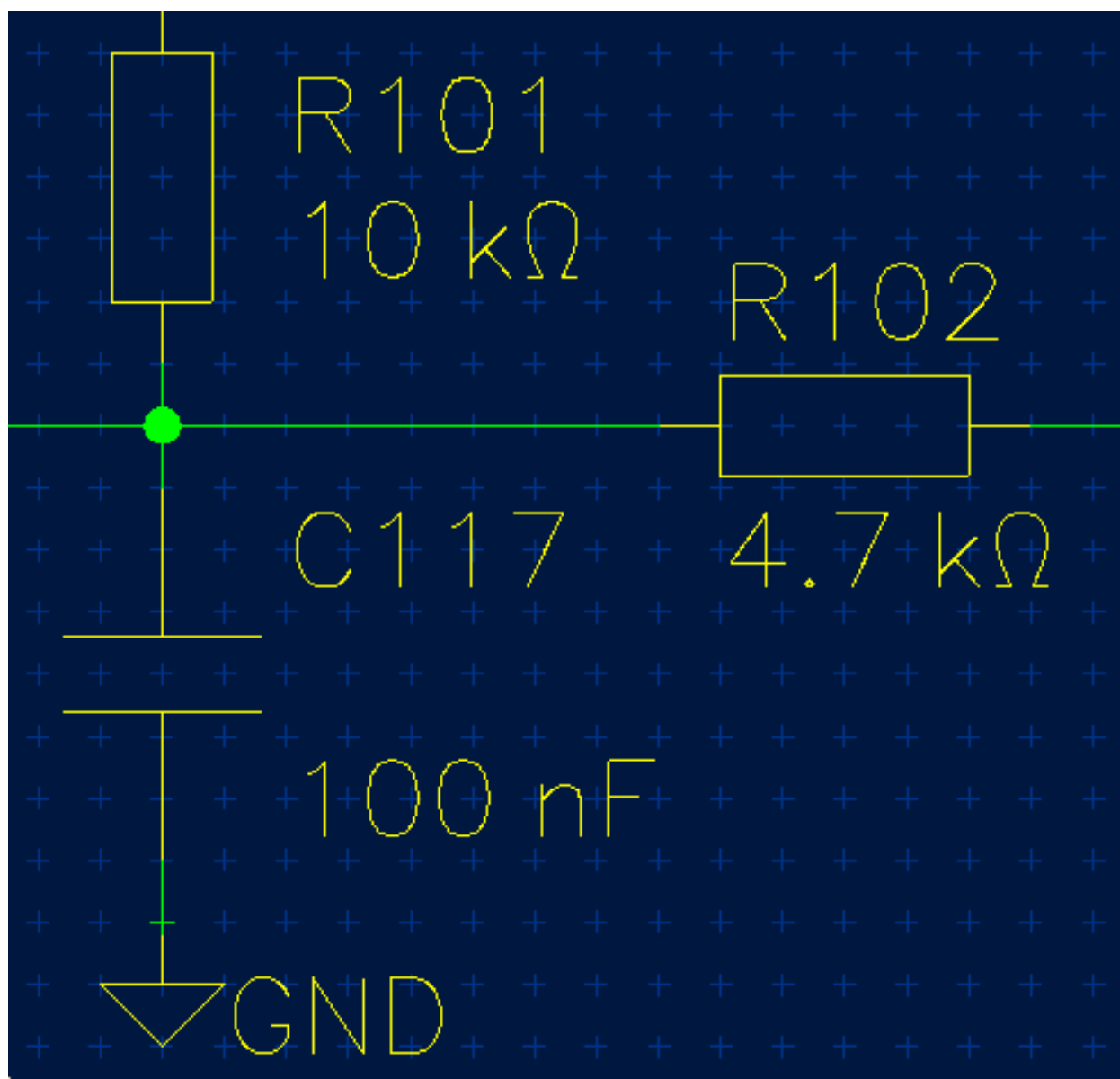
2.5 Schematický editor, který ví co děláte

Kreslení schemat není jen o linkách a štítcích. Schéma v editoru Horizon EDA ví o jednotlivých spojích a zeptá se vás na jejich sloučení:

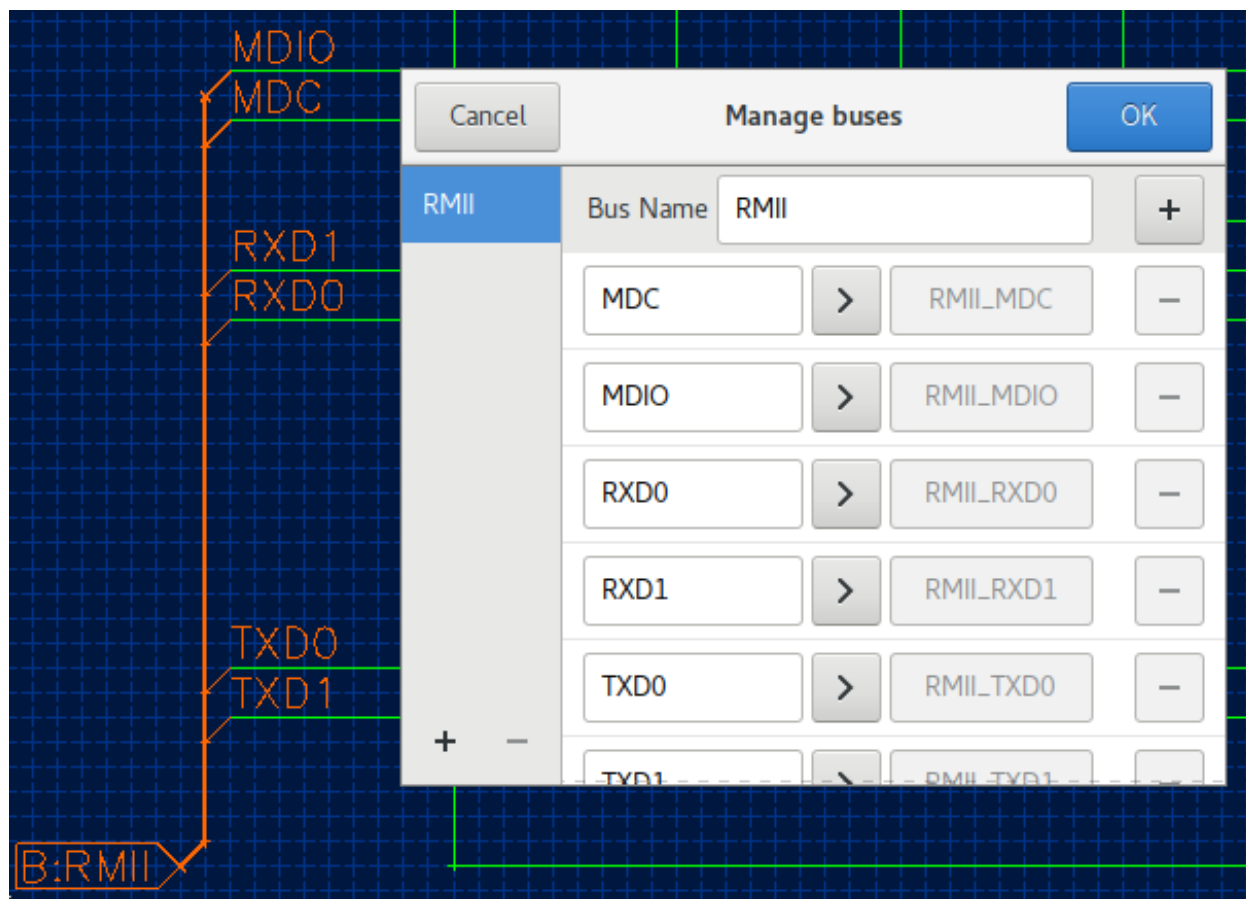


Umístí uzly tam, kde by měly být:

Také automaticky přeorientuje texty, takže se tím vyhnete těžko čitelnému referenčnímu označení:

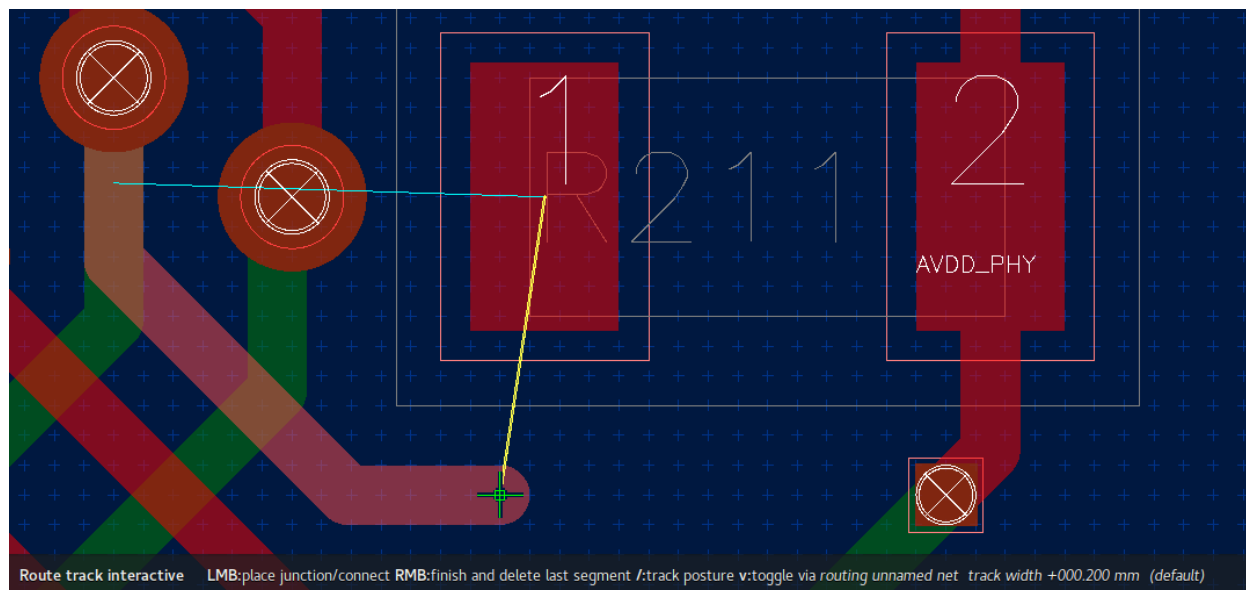


Ani sběrnice nejsou Horizonu cizí:



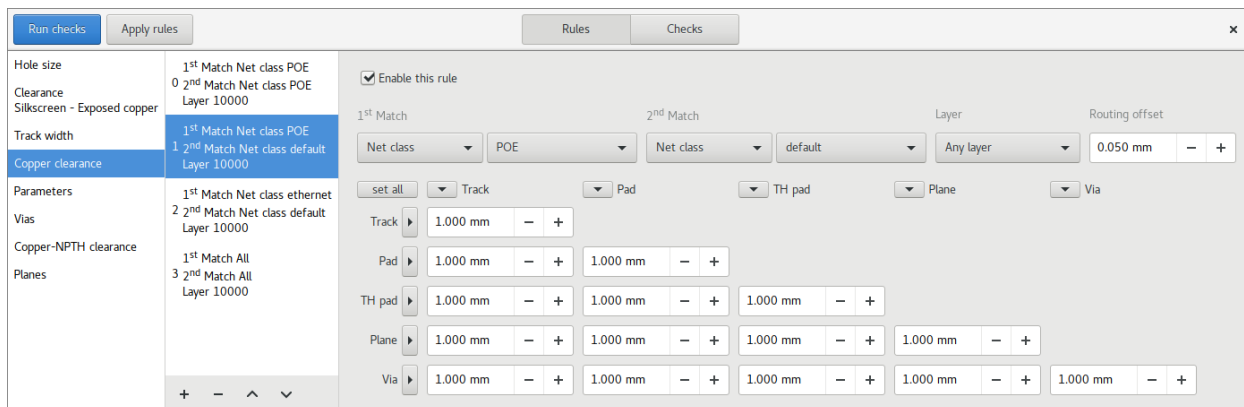
2.6 Interaktivní trasér spojů s přímou kontrolou pravidel (DRC)

Pomocí interaktivního traséru (routeru) původně vyvinutého pro KiCad, se vytváření spojů stane hroučkou. Samozřejmě respektuje vaše pravidla návrhu. Trasování diferenciálního páru je také podporováno.

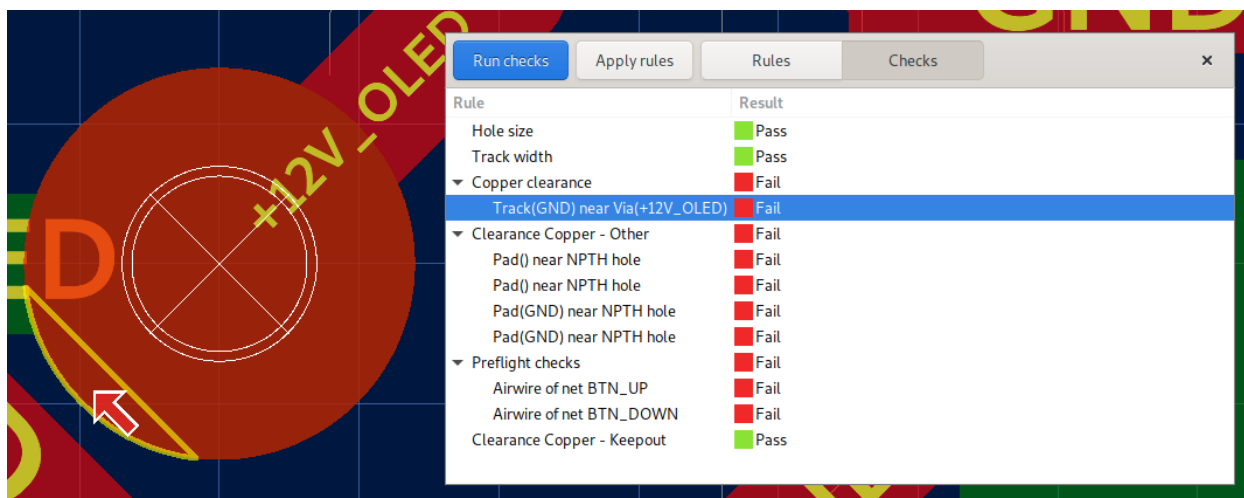


2.7 Silná pravidla

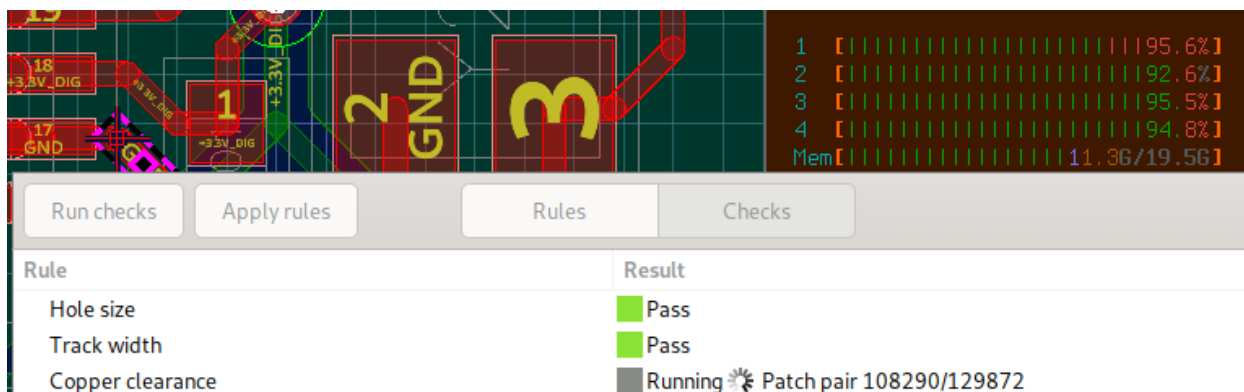
S výkonnými a flexibilními pravidly může horizon kontrolovat a upravovat váš návrh splňující nastavená kritéria:



Pokud něco nesplňuje vaše požadavky, horizon přesně řekne, co to je a na kterém místě:

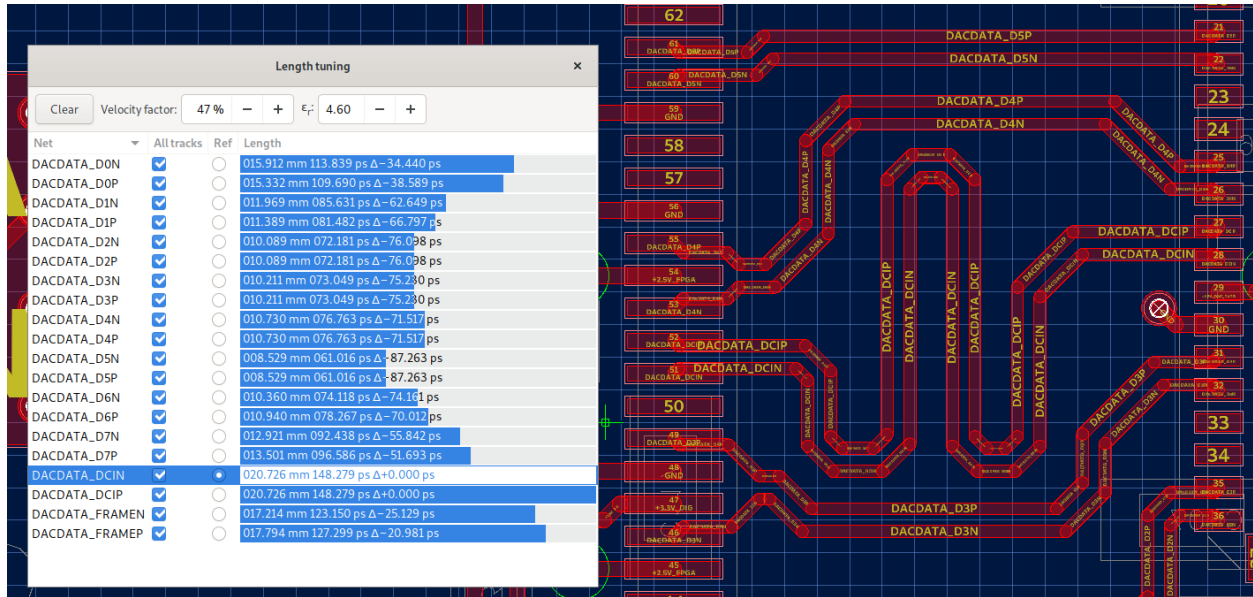


Vícevláknové DRC využívá všechna jádra CPU:



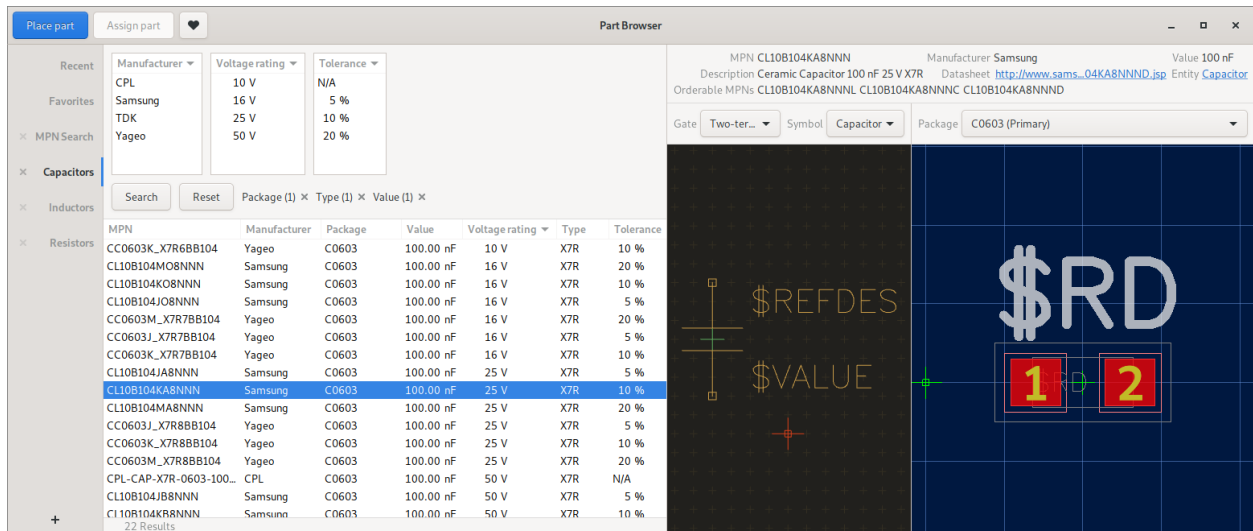
2.8 Interaktivní nastavení délek spojů

Interaktivní měření a nastavení jednotlivých spojů, diferenciálních párů nebo sběrnic:



2.9 Parametrické vyhledávání

Parametrické vyhledávání součástek vám pomůže rychle najít pasivní:



2.10 Skladové informace

Informace o skladových zásobách v reálném čase využitím informací ze serveru Kitspace's partinfo:

Part Browser

MPN LP5907MFX-1.8/NOPB Manufactu
Description 250 mA Ultra-Low-Noise, Low ... Datashe

Gate **Three-t...** ▼ Symbol **Three-t...** ▼ Pac

Digikey Stock

1,072
481
129
... 391
825
172
r Not found
r 241,460
r 348,244
N/A
N/A
... 3,293
2,577
N/A
1,198
4,540
4,126
... Not found
14 0

250-mA ultra-low-noise low-IQ low-dropout (...)

Digikey [296-41463-1-ND](#) Stock: 241,460

1	\$0.550
10	\$0.475
25	\$0.444

Digikey [296-41463-6-ND](#) Stock: 241,460

1	\$0.550
10	\$0.475
25	\$0.444

Farnell [3007714](#) Stock: 8,422

1	£0.402
50	£0.351
100	£0.211

Mouser [595LP5907MFX1.8NPB](#) Stock: 28,289

1	\$0.540
10	\$0.450
100	\$0.271

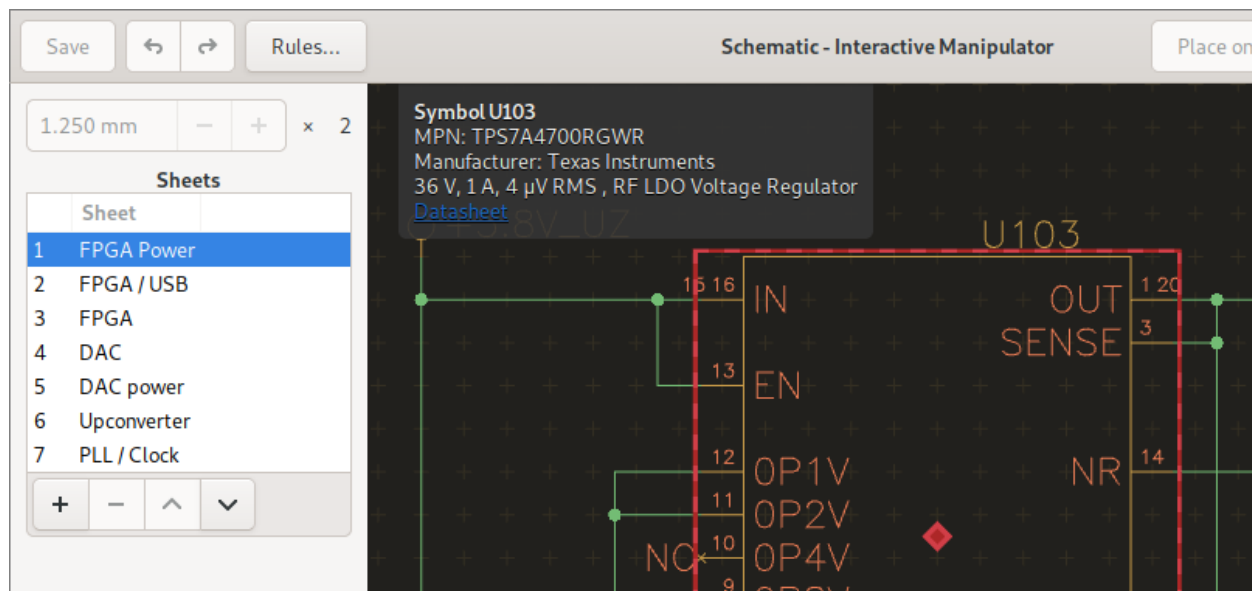
Newark [28AH3785](#) Stock: 8,382

1	\$0.540
10	\$0.450
25	\$0.390

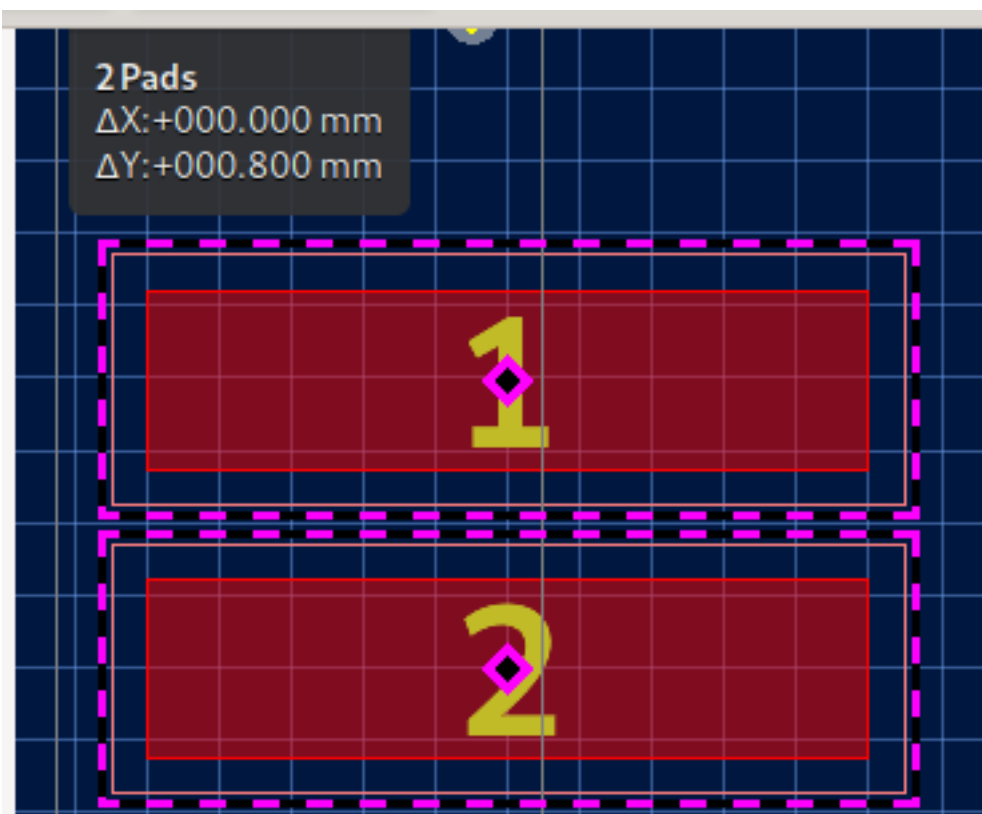
Information provided by [partinfo](#)

2.11 Inteligentní vyskakovací zobrazování

Okamžitě víte, na co se díváte, s přímým odkazem na datové listy:



Měří vzdálenost pájecích míst a mnohem více:



2.12 Průmyslové standardy výroby

Až bude váš návrh připraven k výrobě, jednoduše exportujte ve formátu průmyslového standartu Gerber a NC-vrtání ve formátu RS-274X:

Generate

Fabrication outputs

NC-Drill

Mode

Merge PTH & NPTH

Drill suffix

.txt

Gerber Settings

Outline width

0.010 mm

–

+

Output Settings

Base filename

hubble

Directory

output/gerber

Browse...

Generate Zip

Resulting filenames are the concatenation of base filename and suffix.

Gerber Layers

<input checked="" type="checkbox"/> Outline	Suffix	.gko
<input checked="" type="checkbox"/> Top Paste	Suffix	.gtp
<input checked="" type="checkbox"/> Top Silkscreen	Suffix	.gto
<input checked="" type="checkbox"/> Top Mask	Suffix	.gts
<input checked="" type="checkbox"/> Top Copper	Suffix	.gtl
<input checked="" type="checkbox"/> Bottom Copper	Suffix	.gbl
<input checked="" type="checkbox"/> Bottom Mask	Suffix	.gbs
<input checked="" type="checkbox"/> Bottom Silkscreen	Suffix	.gbo
<input checked="" type="checkbox"/> Bottom Paste	Suffix	.gbp

Wrote layer Bottom Paste to gerber file /home/lukas/code/hubble/output/gerber/hubble.gbp

Wrote layer Bottom Silkscreen to gerber file /home/lukas/code/hubble/output/gerber/hubble.gbo

Wrote layer Bottom Mask to gerber file /home/lukas/code/hubble/output/gerber/hubble.gbs

Wrote layer Bottom Copper to gerber file /home/lukas/code/hubble/output/gerber/hubble.gbl

Wrote layer Top Copper to gerber file /home/lukas/code/hubble/output/gerber/hubble.gtl

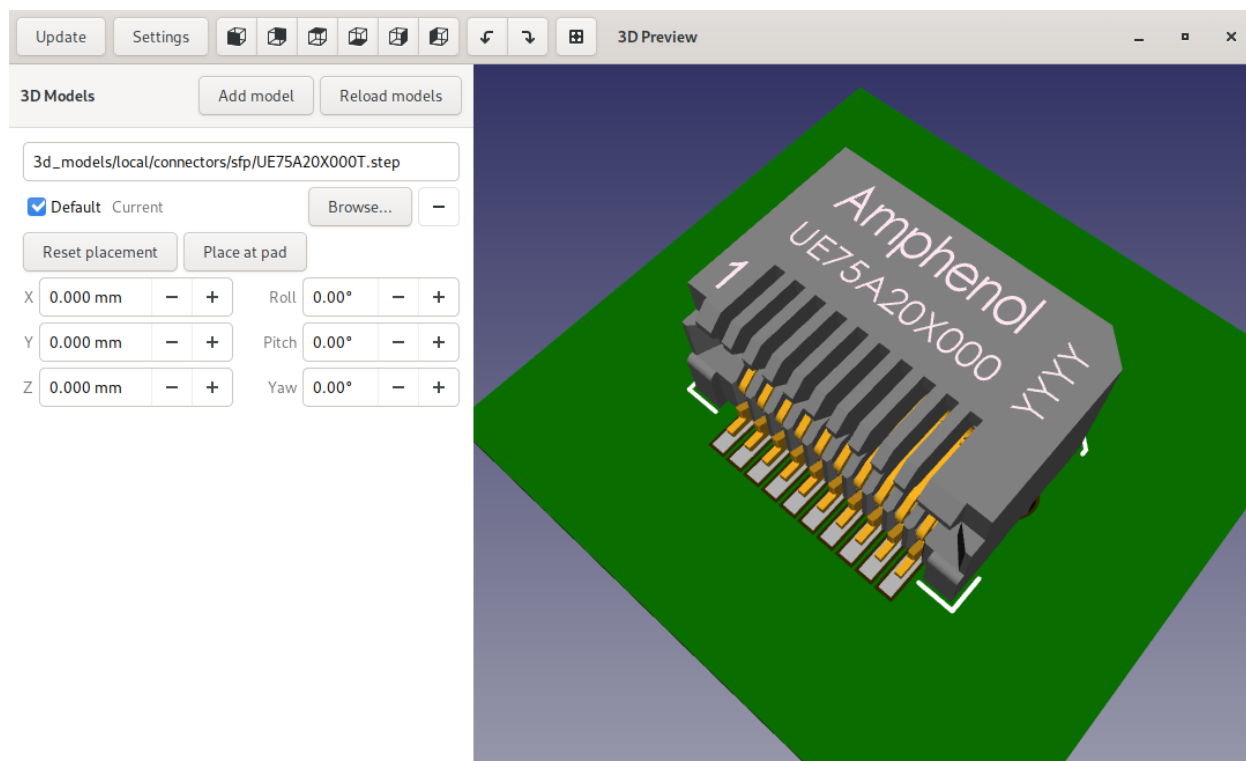
Wrote layer Top Mask to gerber file /home/lukas/code/hubble/output/gerber/hubble.gts

2.13 Integrace do programu CAD

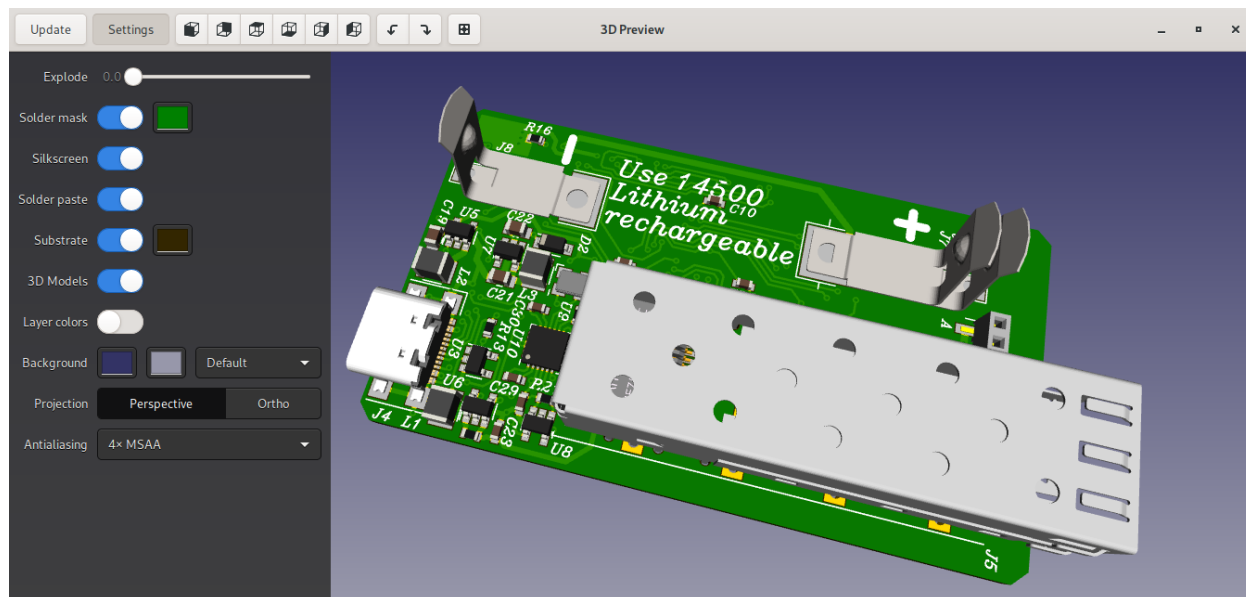
Rozšiřuje pouzdra součástek o třetí dimenzi přidáním 3D modelu v průmyslovém standardním formátu STEP:

16

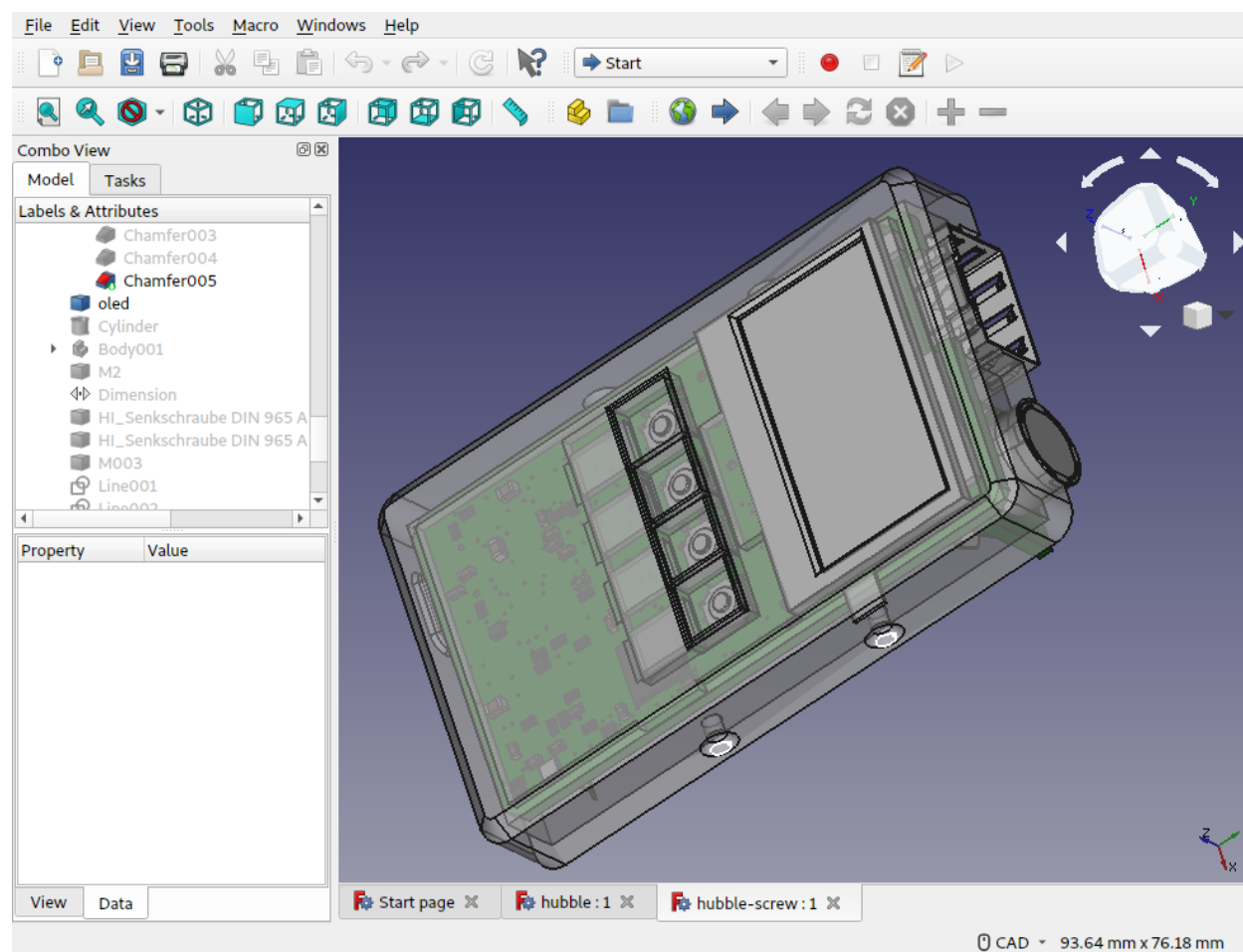
Kapitola 2. Hlavní výhody



Podívejte se na svou desku, jako byste ji drželi ve svých rukou. Za použití funkce rozpadu (Explode), můžete vrstvy zobrazit oddělené.

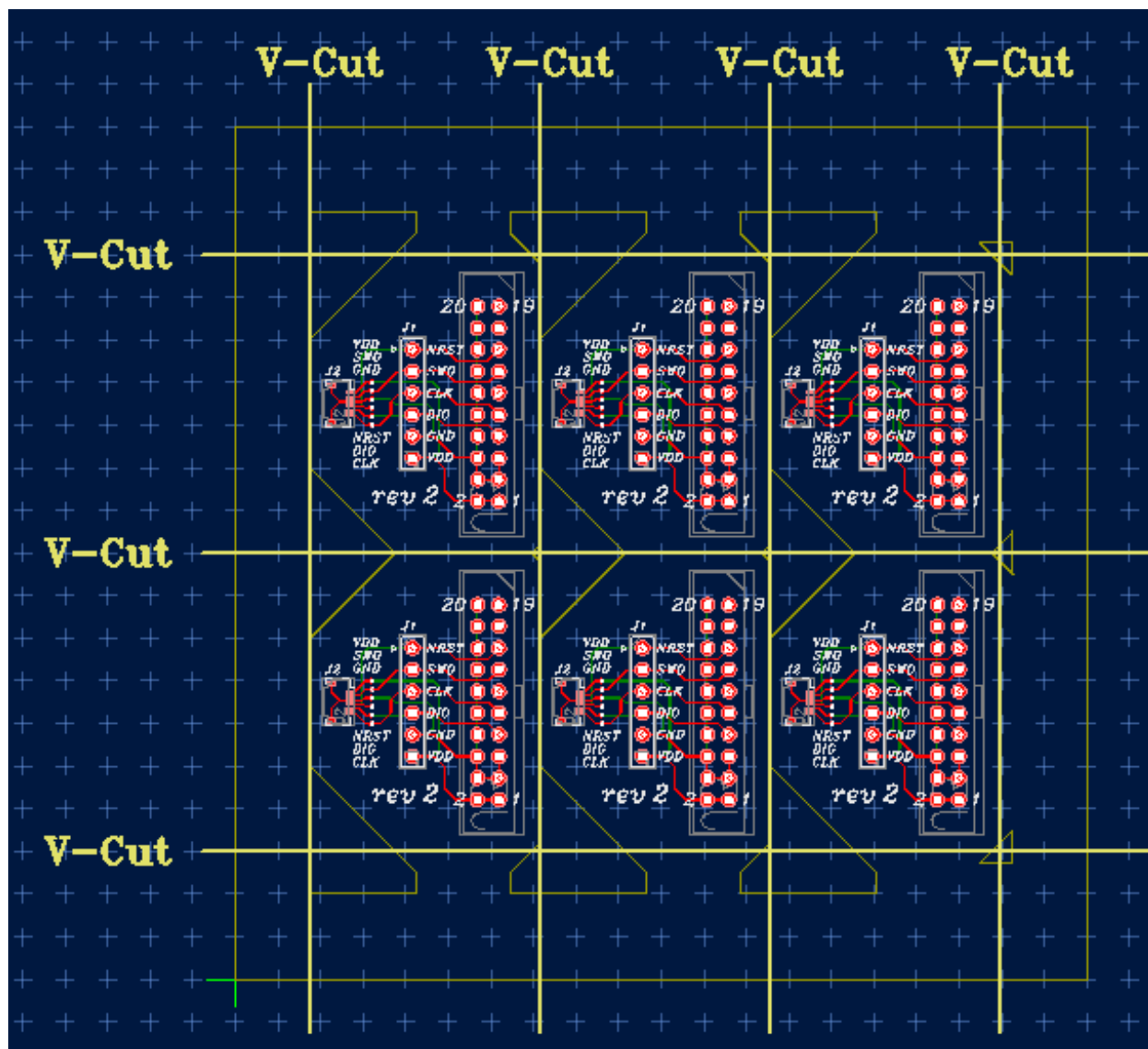


Po dokončení návrhu exportujte desku a všechny modely jako soubor STEP pro použití jako mechanického modelu v CAD programech:



2.14 Bezbolestná panelizace

Snadno lze uspořádat více kopií jedné desky nebo více desek na jednu desku (DPS) a tak ušetřit peníze při objednávce malých DPS:



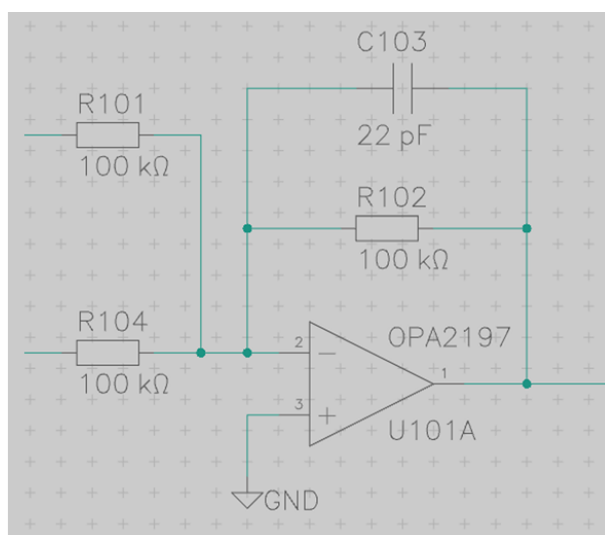
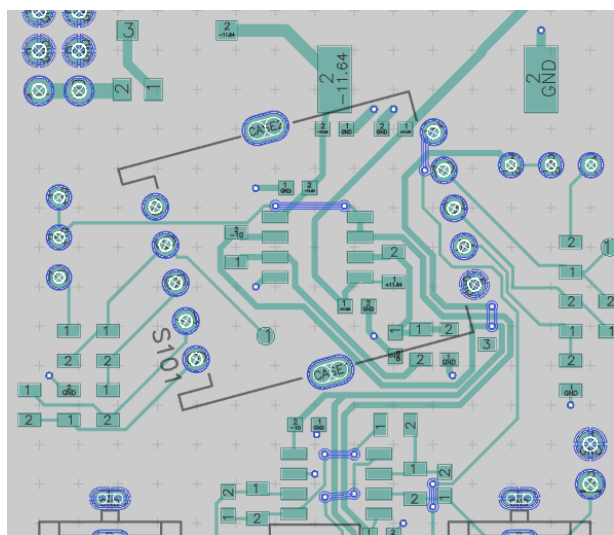
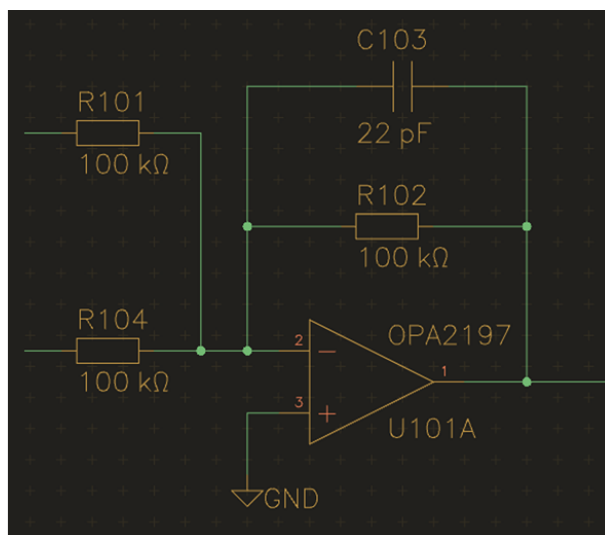
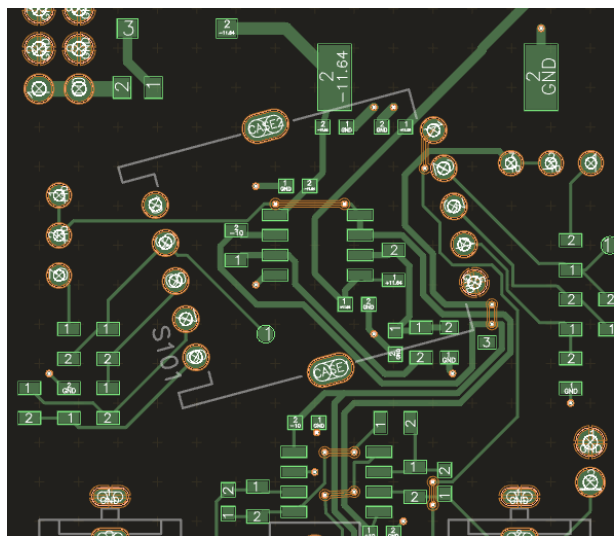
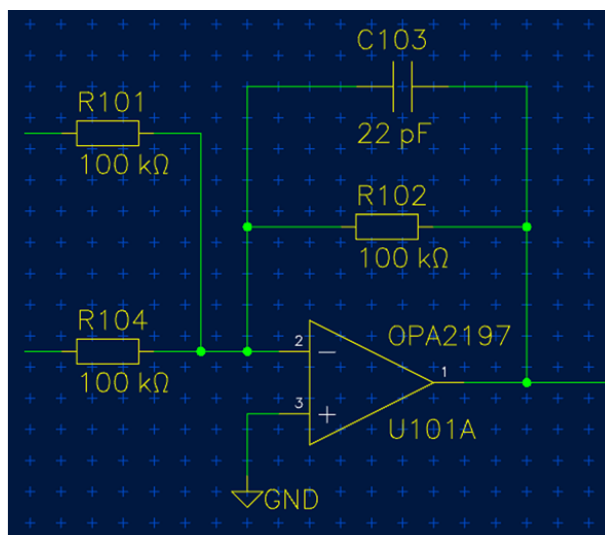
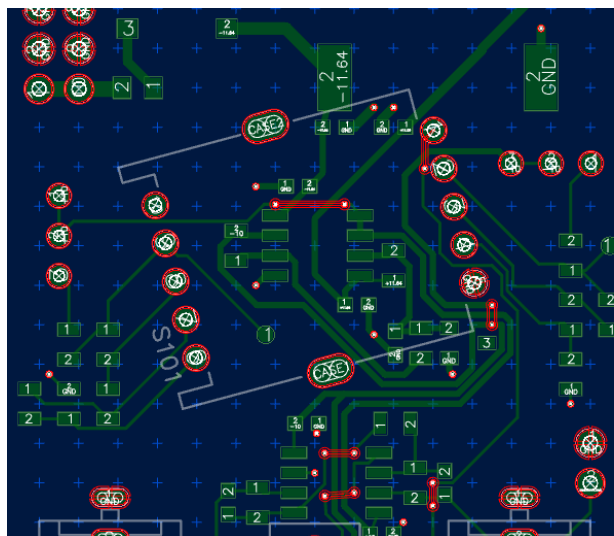
Všechny sloučené desky jsou spojeny s původním návrhem, takže se budou aktualizovat, když se původní návrh změní.

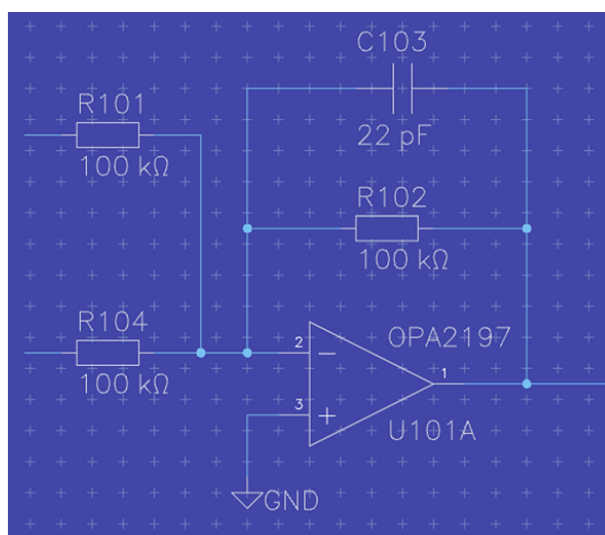
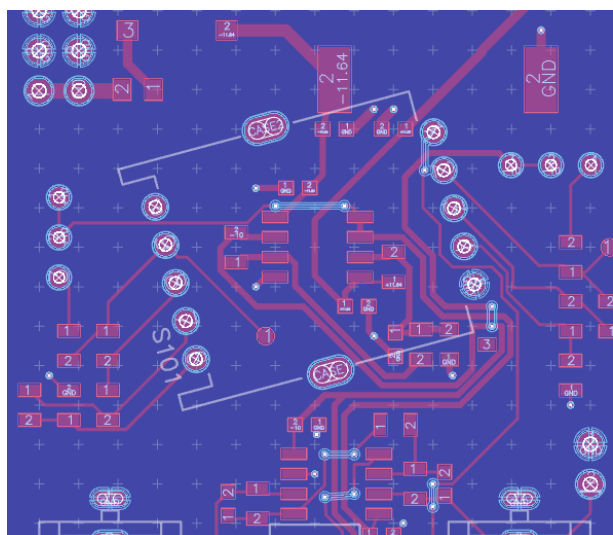
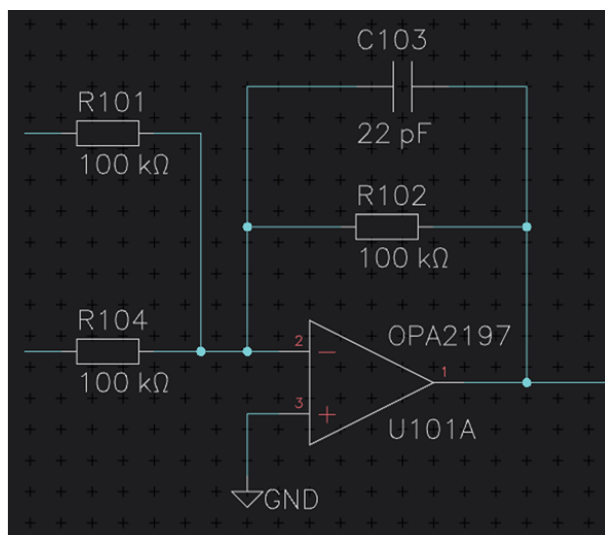
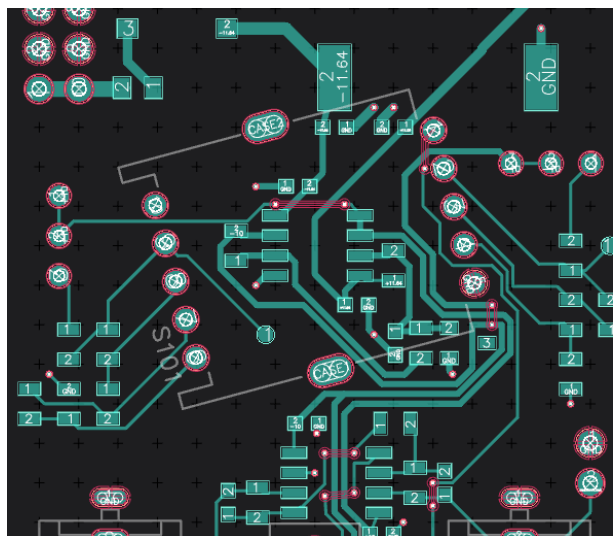
2.15 Manipulace pomocí různých vstupních zařízení

V programu Horizon EDA můžete využít moderního polohovacího zařízení notebooku. Kromě zoomování s přesností na pixel a posouvání pomocí touchpadů nebo sledování bodů, můžete přímo manipulovat s 2D a 3D pohledy pomocí gesta dotykové obrazovky, jako je přiblížení.

2.16 Přizpůsobte si vzhled

Dekorace ovlivňuje lidi a lidé jsou různí - udělejte si své vlastní nastavení barev nebo si vyberte z existujících barevných schémat.





Klávesové zkratky jsou také plně přizpůsobitelné

2.17 Je toho mnohem víc

- Akcelerované zobrazování (OpenGL)
- Neomezená možnost příkazů zpět / opakování
- Kopírování / vkládání, a to i mezi instancemi
- Vyplněné plochy spojů
- Libovolné tvary pájecích míst
- Import tvarů ve formátu DXF
- Export kusovníku (BOM)
- Exportovat soubory výběrem a umístěním

Abychom zúžili zaměření projektu, některé věci zde záměrně nejsou.

3.1 Automatický router (trasér spojů)

Napsání programu dobrého autorouteru je spousta práce, která zabere spoustu času a přitom není tak důležitá, protože zkušenosti ukazují, že autorouter je málokdy užitečný pro malé a střední desky.

3.2 Simulace

Návrh schemat pro desky plošných spojů a návrh schemat pro simulaci typu SPICE jsou velmi odlišné, protože ve schématu simulace se často zjednodušují aspekty skutečného světa, jako je nahrazení vstupu ADC náhradním obvodem. Za mě mohu říct, že jsem naprosto spokojený s LTSpice z hlediska uživatelského rozhraní a jádra SPICE a moc nevidím prostor pro nový vývoj v tomto směru.

3.3 Vykreslování

Jiné aplikace EDA nedávno získaly vlastní raytracer pro vykreslování realistické 3D vizualizace desek s obvody. Rozsah 3D zobrazení založené na OpenGL je dostačující pro kontrolu desky s problémy, jako je zapomenutá pájecí maska a získání představy o tom jak bude vypadat sestava desky. Jakoukoli lepší vizualizaci je možné zařídit exportem do 3D modelovacího softwaru, jako je Blender.

3.4 Formáty souborů

Mnoho lidí si stěžuje, že zde neexistuje žádný průmyslový standardní formát pro schémata a desky. Formáty souborů proto jsou vlastní formáty aplikací, což znamená, že musí podporovat každé tlačítko, které aplikace má. Přijetí formátu jiné aplikace v Horizontu EDA by proto vedlo k tomu, že by Horizon EDA byl špatným klonem jiné aplikace.

JSON byl vybrán jako formát serializace, protože přímo mapuje na běžné datové struktury, jako jsou mapy a pole (na rozdíl od XML) a jsou snadno manipulovatelné téměř v každém prostředí.

Dále: [*Instalace*](#)

Takže jste se rozhodli vyzkoušet Horizon-EDA? Skvěle! Zde je návod.

4.1 Stabilní verze

Poslední vývojová verze.

4.1.1 Operační systém Windows

Stáhněte a spusťte instalační program MSI z [GitHub release](#).

4.1.2 Linux

Získejte nejnovější stabilní verzi z [Flathub](#).

Pokud si chcete zkompilevat program sami, stáhněte si zdrojový tarball z [GitHub](#) a postupujte podle pokynů v *Sestavení (kompilace) programu na operačním systému Linux*.

Pro Arch Linux existuje [AUR](#) instalační balíček.

Mějte na paměti, že binární balíčky poskytované vaší distribucí mohou být zastaralé.

4.2 Vývojová verze

Obvykle funguje, ale občas se může seknout, takže ji používejte jen na vlastní nebezpečí. Doporučeno, pokud chcete získat nejnovější funkce a opravy chyb.

4.2.1 Operační systém Windows

Stáhnete nejnovější komprimovaný přeložený binární soubor z adresy [AppVeyor CI](#) a někam ho rozbalte. Pamatujte, že se jedná o 64bitové binární soubory. V případě, že překlad dosud probíhá nebo ho někdo přerušil, si můžete stáhnout předešlé verze z [the build history](#) (kliknutím na odkaz stáhnete soubor ve formátu zip)

4.2.2 Linux

Klonujte program z úložiště a podívejte se na *Sestavení (kompilace) programu na operačním systému Linux*, kde najdete pokyny, jak sestavit Horizon na linuxu.

Dále: *Nastavení fondu knihoven*

Nastavení fondu knihoven

Existují dva způsoby, jak založit fond. Pokud znáte systém pro správu verzí Git a chcete mít přímou kontrolu nad vaším úložištěm můžete použít níže uvedený pracovní postup Gitu, pokud ne, stačí použít fond manažer.

5.1 Program Git

Pokud znáte program Git, stačí někde naklonovat soubory z veřejného archivu [horizon-pool](#). Je vhodné použít program Git, aby se udržely vaše místní kopie aktuální a mohli jste případně publikovat vaše nové součástky do veřejného fondu.

5.2 Správce fondu knihovny

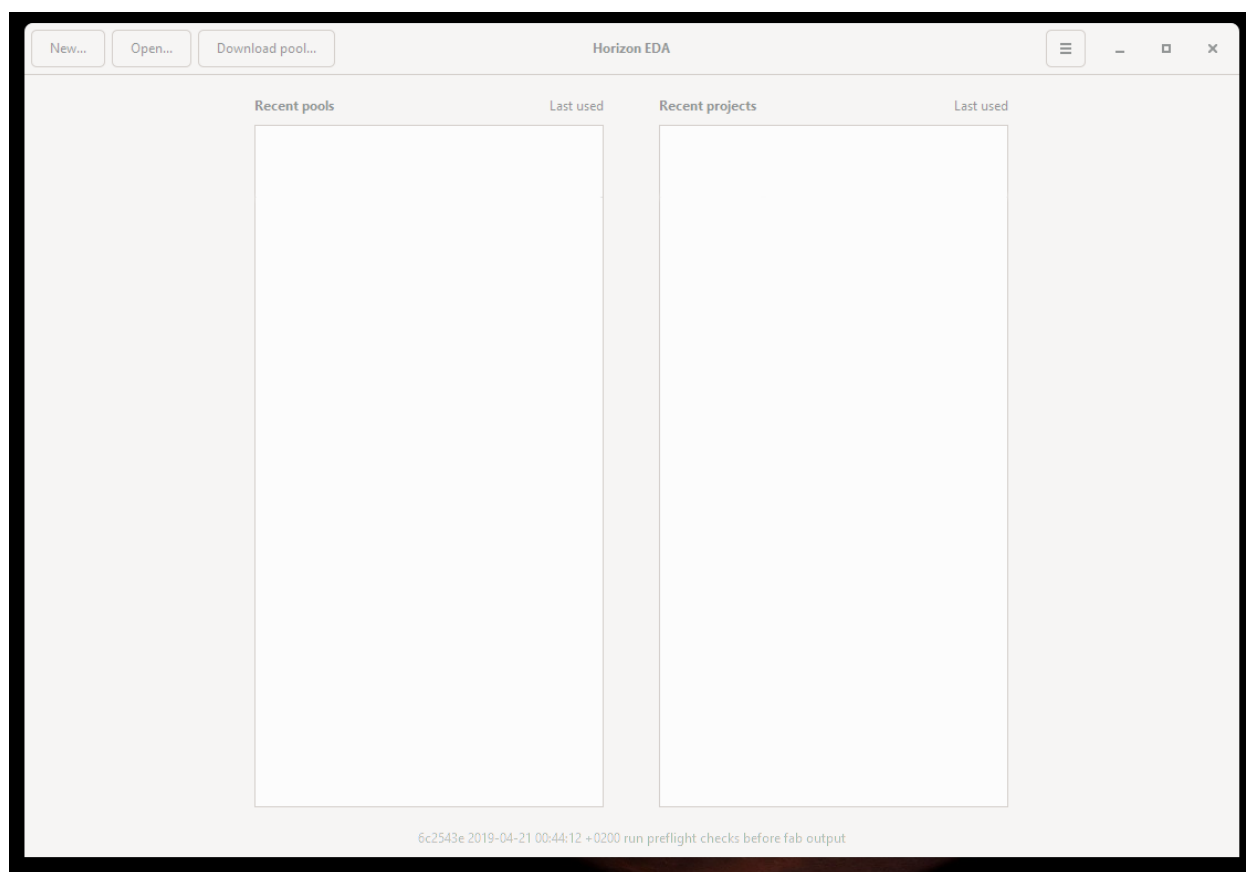
Nevíte, jak na to? Žádný problém! Poklepejte na `horizon-eda.exe` nebo spusťte `./horizon-eda` z vašeho příkazového řádku a klikněte na „Download...“ pro stažení fondu. Výchozí fond `horizon-eda/horizon-pool` je ten který potřebujete. Správce fondu vám pomůže s udržováním vašeho fondu v aktuálním stavu, viz karta „Remote“. Také vám pomůže s vytvořením žádosti o doplnění nových součástí do veřejného fondu, a tak můžete snadno přispět do veřejného fondu bez jakýchkoli znalostí programu Git.

Dále: *Vytvoření nového projektu*

KAPITOLA 6

Vytvořte nový projekt

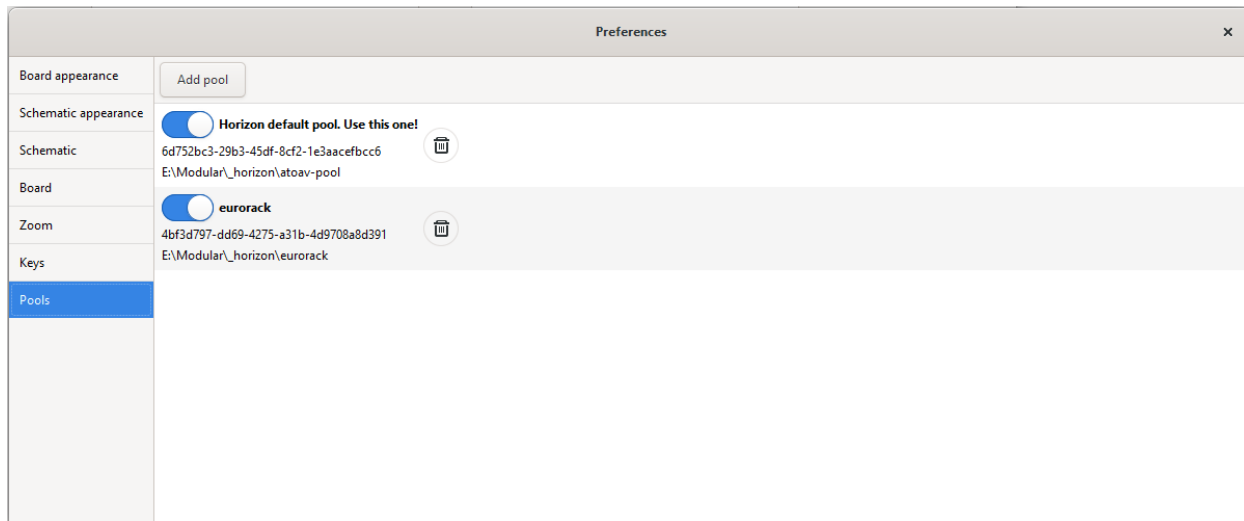
Když spustíte program Horizon-EDA mělo by se zobrazit okno podobné tomuto.



Ujistěte se, že máte nastavený a přidáný fond knihoven součástek.

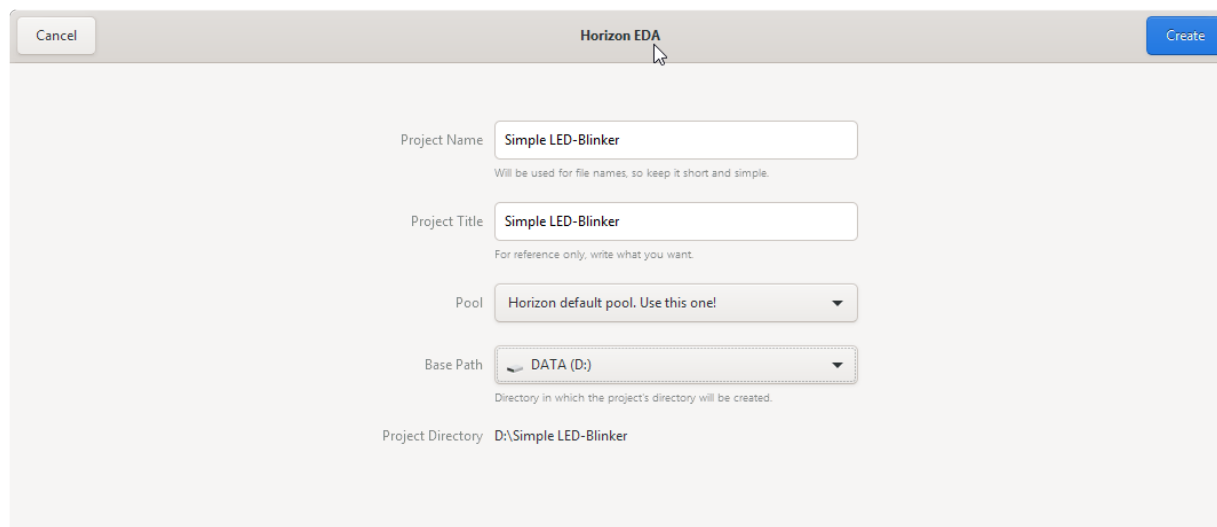
Klikněte na ikonu aplikace v levém horním rohu a otevřete dialog nastavení. Přidejte fond, který jste právě stáhli, tím,

že vyberete složku obsahující soubor `pool.json`. Když to dokončíte dialogové okno předvoleb by mělo vypadat takto:

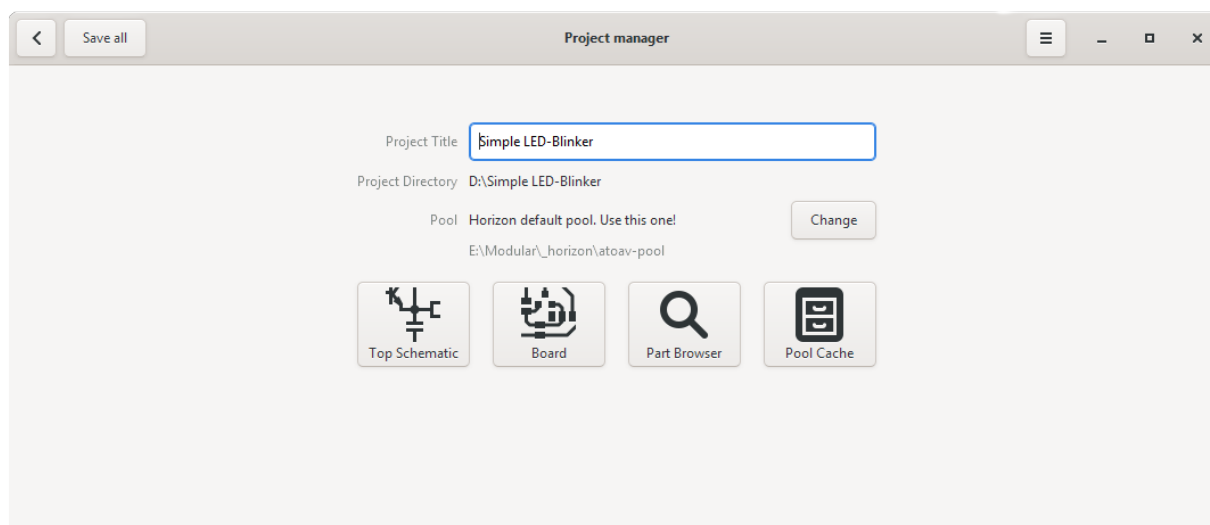


Nyní vytvořte nový projekt klepnutím na „New...“ a následně na „Project“.

V okně Dialogové okno projektu můžete vybrat název projektu a místo, kam chcete uložit složku projektu. Navíc můžete změnit aktivní fond:



V okně Správce projektů, které se nyní otevře, můžete vytvořit schéma, desku, najít si díly nebo spravovat vyrovnávací paměť fondu.

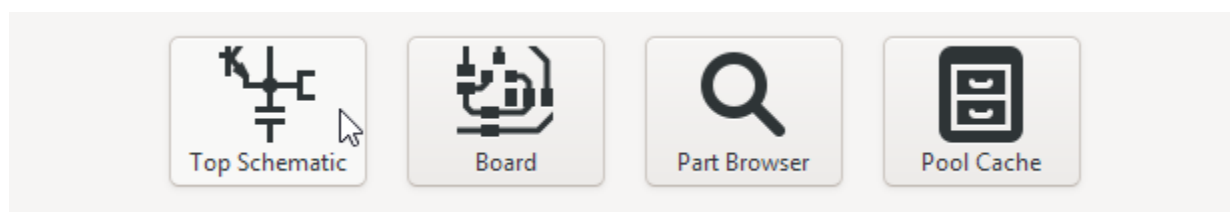


Vyrovňovací paměť fondu uchovává kopii všech součástek, které jste použili ve vašem projektu a pomůže vám chránit vaše projekty před vnějšími změnami (např. aktualizace ve fondu). Když *chcete* aktualizovat součásti nebo odstranit nepoužité části můžete udělat v okně mezipaměti fondu.

Dále: *Kreslení schémat*

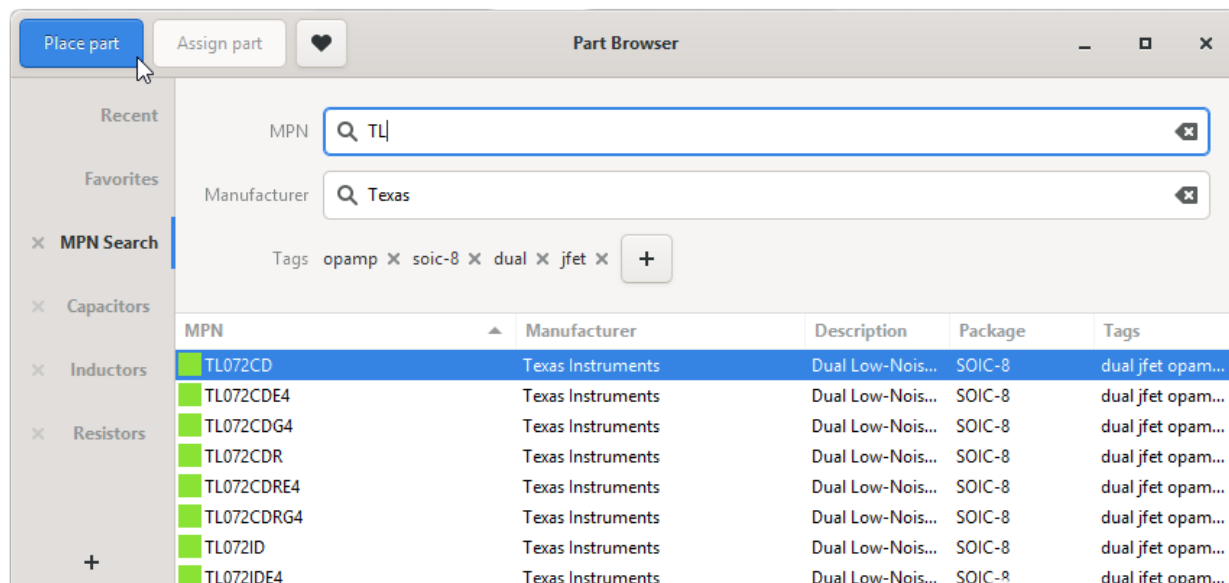
Kreslení schémat

Ve Správci projektů klikněte na tlačítko „Top Schematic“ a otevřete Editor schemat:



7.1 Umístěte součástku

Potom můžete začít umísťovat součásti kliknutím na tlačítko „Place Part“ v horní liště nebo jednoduše stiskem kláves **p p** (zapamatujte si jako počáteční písmena „Place Part“). Otevře se Prohlížeč součástek, kde můžete najít součástky, které potřebujete v celém svém fondu a umístěte je do schématu.



7.2 Pospojujete součástky

Jakmile umístíte součástky, můžete aktivovat příkaz „Draw net line“ pomocí stisku klávesy **n** a nakreslit spojení. Můžete také jednoduše spojovat vývody přetažením myši vývodu na vývod:

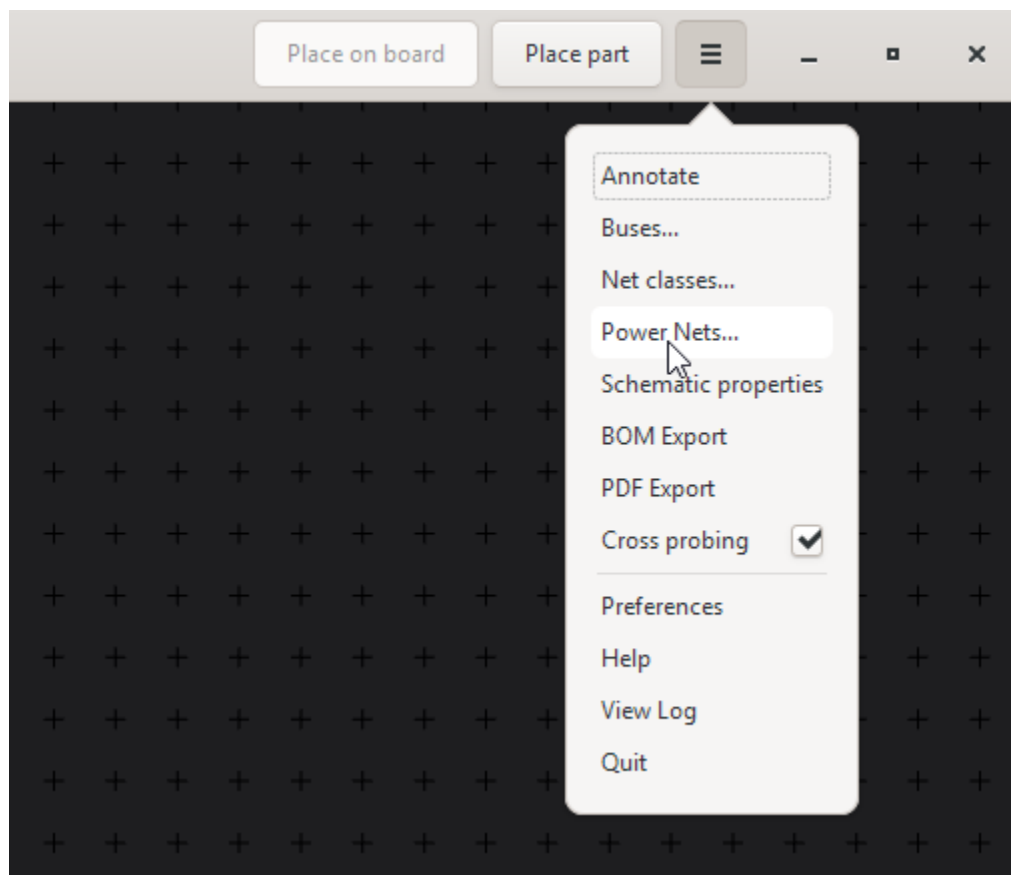
7.3 Základní posunutí

Součástky a spoje můžete přesouvat jejich výběrem a stiskem klávesy **m**, nebo pomocí stisku kláves se šipkami **←/↑/↓/→**. Otáčejte pomocí **r** a zrcadlete pomocí **e**

Pokud některou z těchto kláves zapomenete, stačí stisknout klávesu mezerník a otevřít *Spacebar Menu* a vyhledat příkaz, který jste hledali.

7.4 Spoje napájení

Protože obvody dávají mnohem větší smysl, pokud používáte spoje napájení, které můžete vytvořit kliknutím na položku nové napájecí sítě v nabídce aplikace:

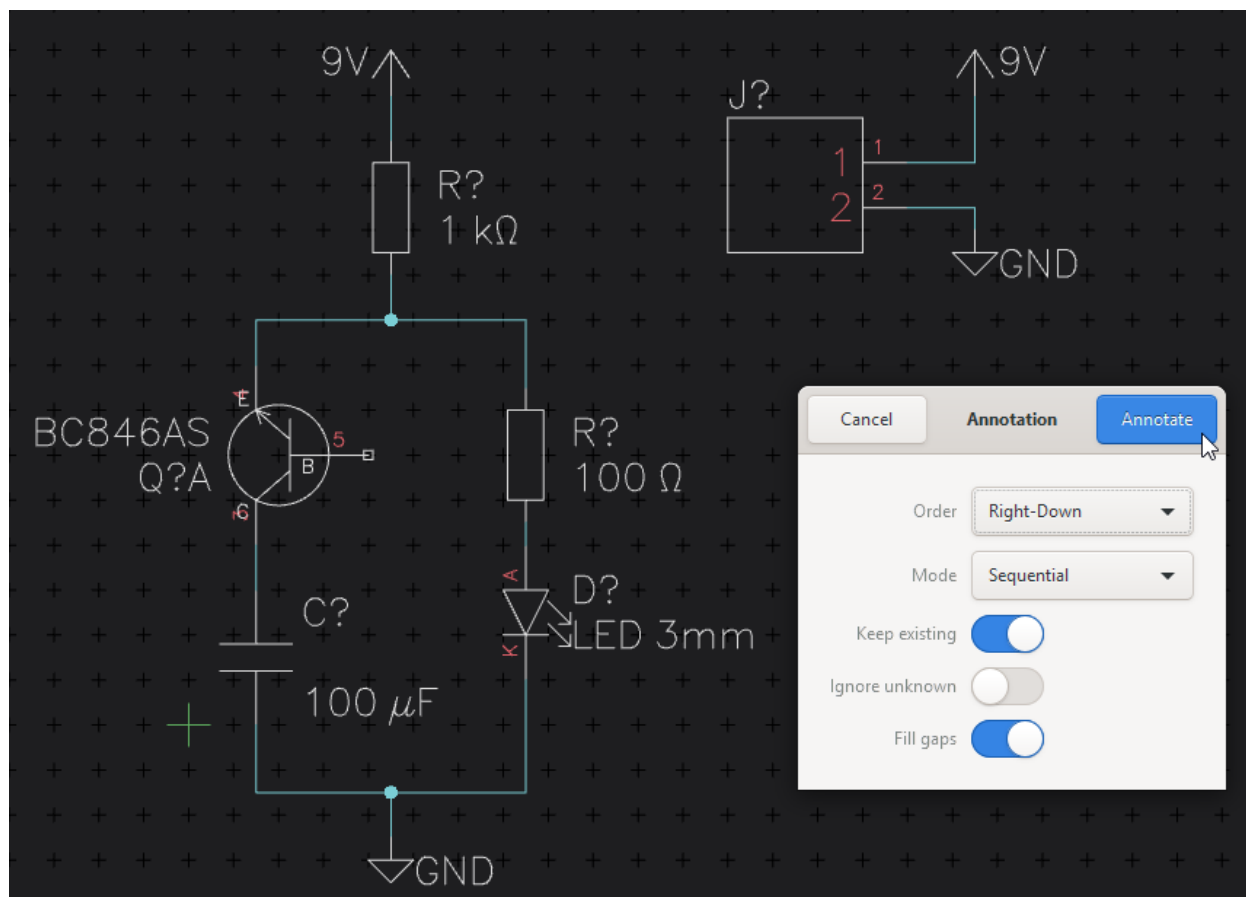


V okně, které se zobrazí, stiskněte tlačítko „Add power net“ a zadejte jméno a vyberte styl symbolu.

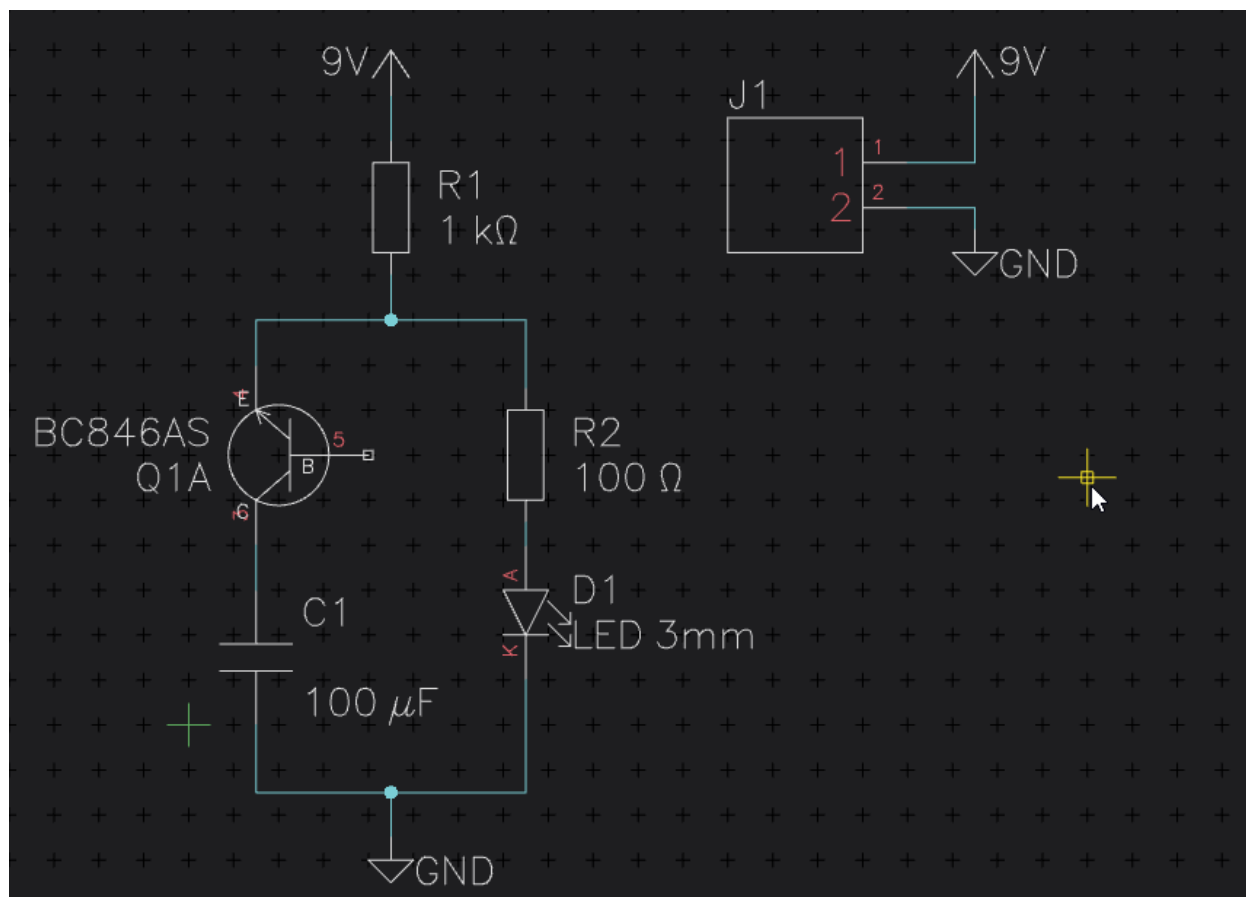
Umístěte napájecí spoje pomocí příkazu „Place power symbol“ (stiskem kláves `p` `o`).

7.5 Popisky

Spusťte příkaz pro Popisky, abyste měli všechny součástky bez popisku (označené ikonou „?“ v jejich značce) čísla položky. Takže z výchozího:



udělejte toto

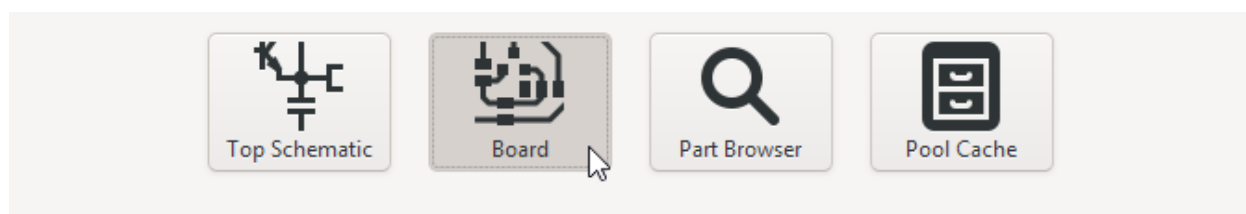


Až budete hotovi, klikněte na „Save“, abyste mohli začít tvořit desku pro schéma, které jste právě vytvořili.

Dále: [Vytvořte desku](#)

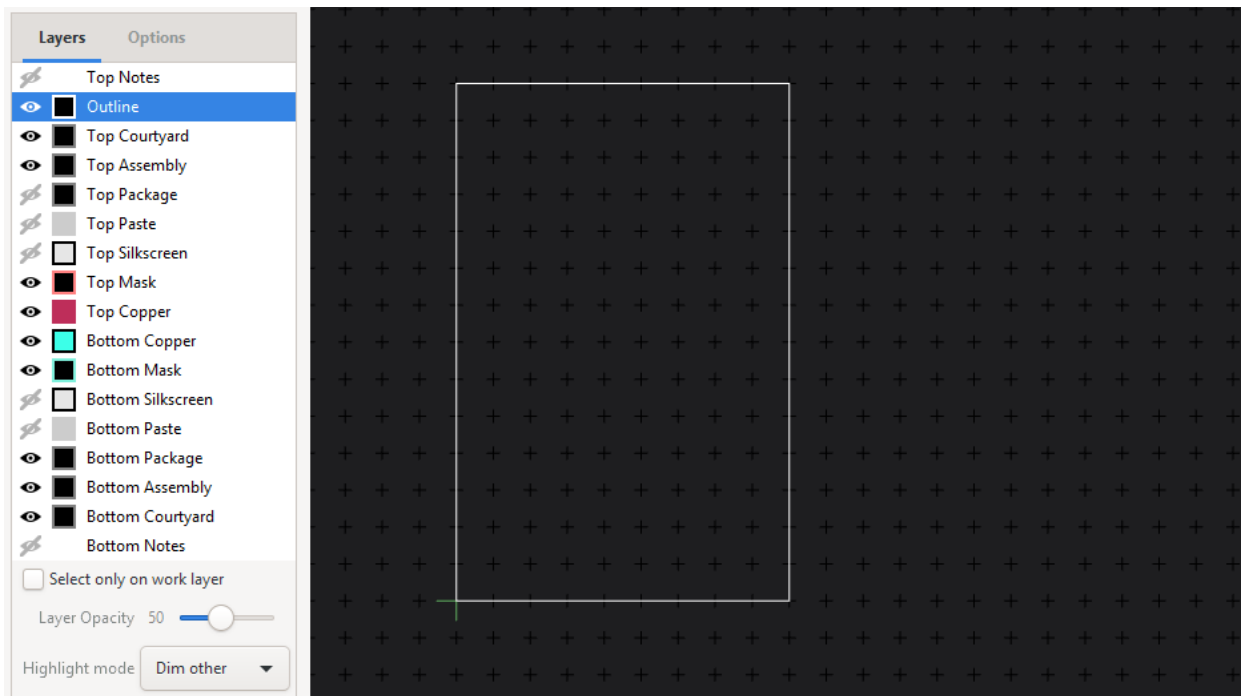
Vytvořte desku

Otevřete editor desky z hlavní nabídky projektu.



8.1 Nakreslete obrys desky

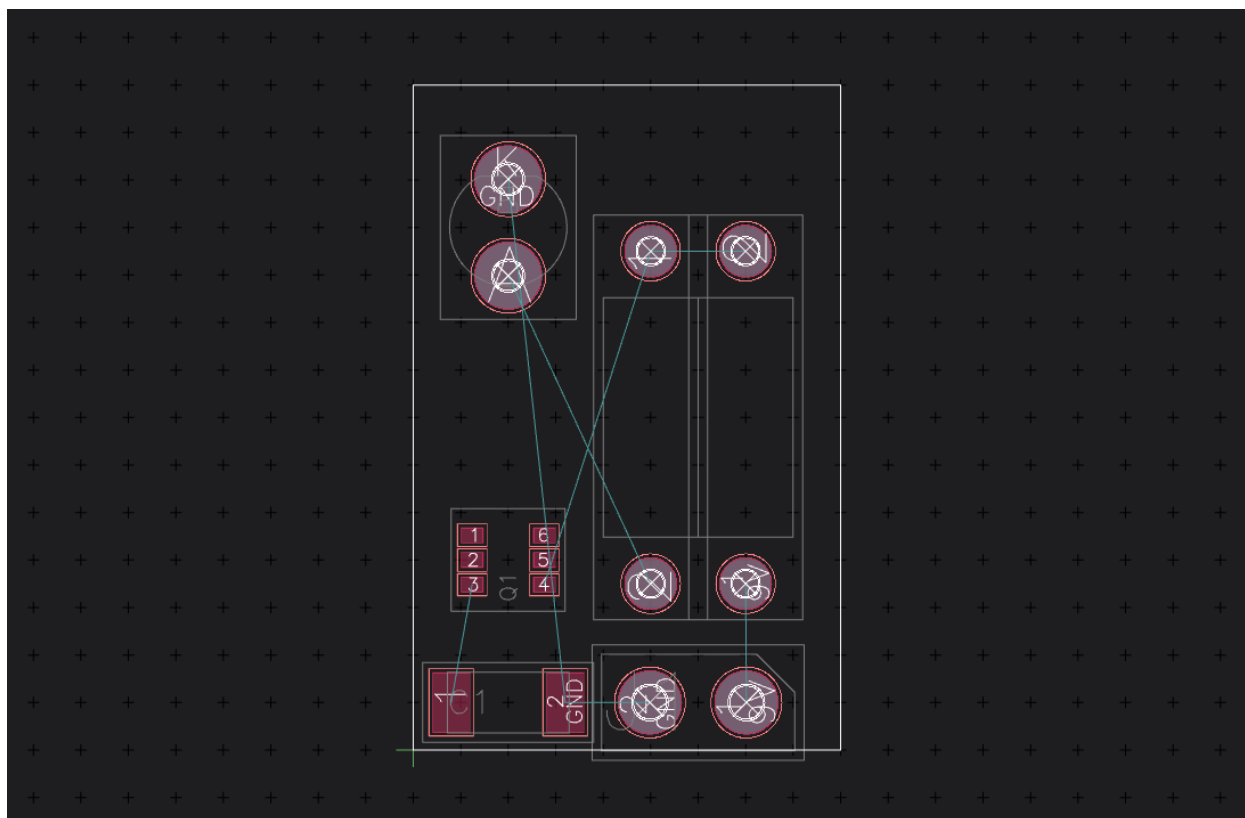
Nakreslete obrys desky tak, že nejprve vyberete hladinu „Outline“ v panelu hladin a poté vyberte příkaz „Draw polygon rectangle“ pomocí stisku kláves `d Y` (nebo použijte nabídku mezerníku):



8.2 Umístěte pouzdra součástek

Umístěte pouzdra na desku pomocí stisku kláves `p p` (tentokrát to znamená „Place Package“ (umístí pouzdro)). Pokud stále máte otevřen editor schemat můžete také vybrat Symbol a použít příkaz „Place on Board“ klepnutím na tlačítko v horním panelu.

Pokud vyberete součástky v editoru schémat, budou také zvýrazněny. v Editoru desky. Pokud změníte něco ve editoru schemat přeneste změny ze schématu na desku uložením schematu a kliknutím na ‚reload netlist‘ v editoru desky.

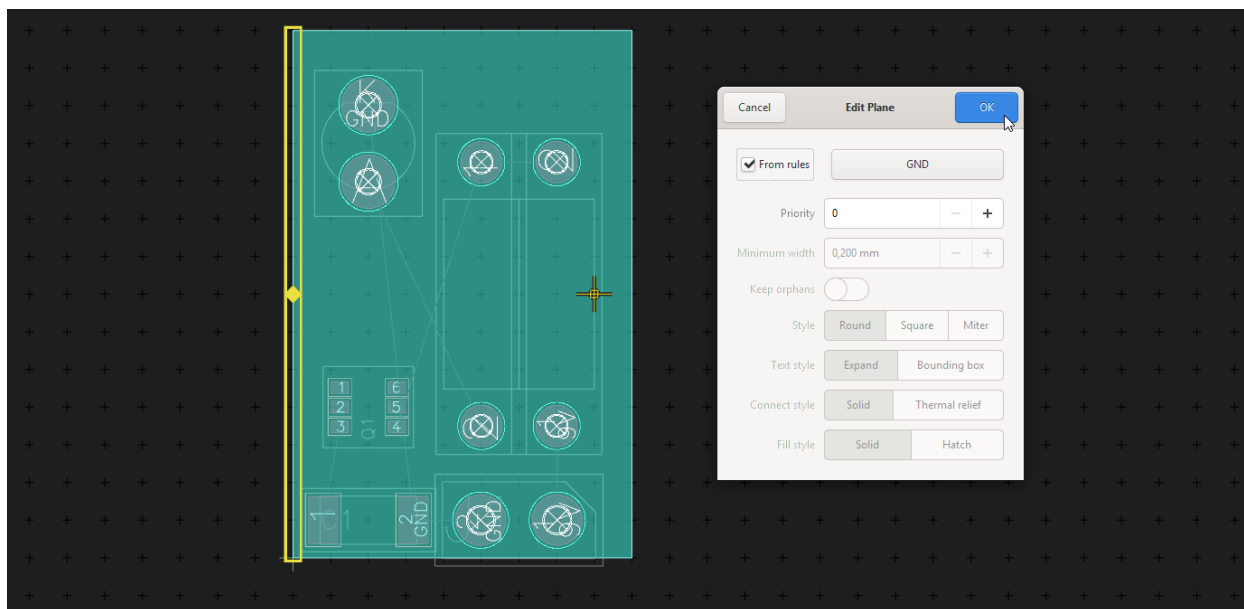


Pokud chcete umístit součástku na spodní část desky, stiskněte **e** pro přemístění na spodní stranu. Pokud hodně pracujete na spodní straně můžete použít nástroj „View bottom“ k převrácení celé desky.

8.3 Přidejte vyplněné oblasti

Pokud nyní chcete vyplnit zemní plochu, je správný čas ji přidat:

1. Vyberte hladinu, na které chcete mít rovinu (Top- nebo Bottom-Copper)
2. Nakreslete polygon stiskem kláves **d** **y**
3. Klikněte pravým tlačítkem na jednu z hran nově přidaného polygonu a vyberte „Add Plane“
4. V okně, které se zobrazí, vyberte spoj GND (pokud tam není a nepřidal jste silový spoj do schématu)



Pokud vás obtěžuje plná barva spoje, můžete přepnout na jiný styl zobrazení kliknutím na barevný čtvereček nalevo od hladiny:

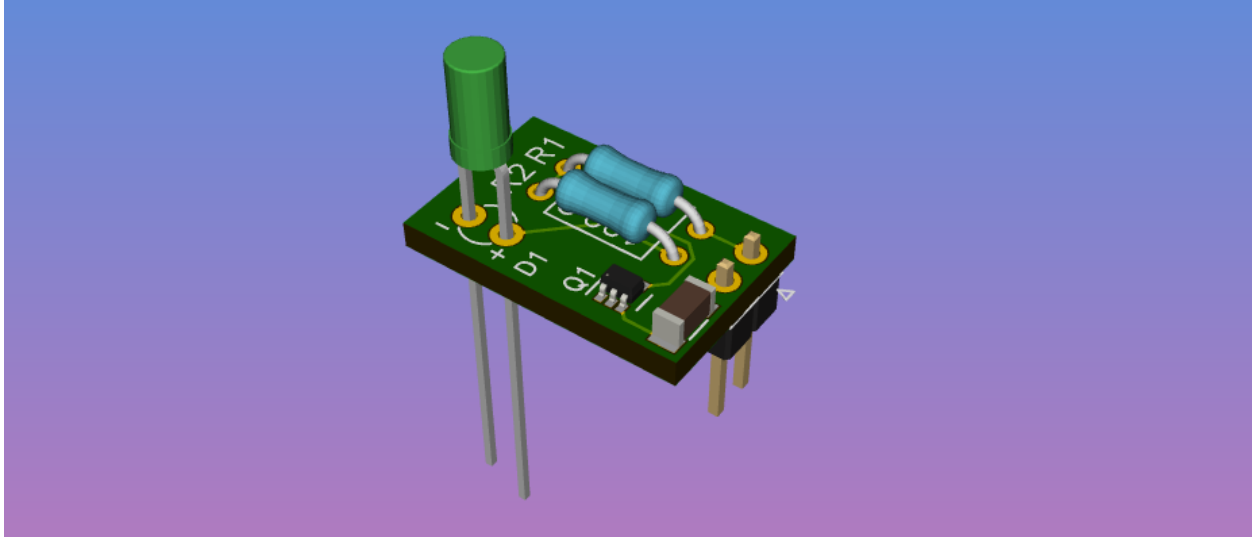
8.4 Umístěte spoje

Chcete-li trasovat spoje mezi součástkami, které jste propojili ve schématu, zadejte **x** a přetažením myši z libovolného vývodu zahajte příkaz spoj:

Trasa bude vždy směřována na vybrané vrstvě, můžete rychle vybrat horní vrstvu mědi stisknutím **1** a dolní vrstvu mědi stisknutím **2**.

Pokud chcete změnit stranu během umístění spoje, můžete umístit průchod. stisknutím klávesy **v** a potom číslo vrstvy na které chcete pokračovat.

Až budete hotovi, podívejte se na 3D pohled vaší desky a exportujte ve formátu Gerber pomocí nabídky „Fabrication Output“ v menu aplikace.



Dále: *Podívejte se na příklad projektu*

KAPITOLA 9

Příklad projektu

Místo zahájení vlastního projektu si také můžete projekt stáhnout např. [návrhové soubory projektu X-Band transmitter](#). Pro otevření, vyberte složku obsahující soubor `ddstx.hprj` v projektovém manažeru. Ujistěte se, že jsou rozbaleny všechny soubory obsažené v tomto archivu.

Dále: *Základní použití editoru: Příkazy*

Abychom zajistili jednotné uživatelské prostředí a umožnili opětovné použití kódu, editory symbolů, schémat, pájecích obrazců, pouzder a desky jsou založeny na interaktivním manipulátoru.

10.1 Jak vybrat správný příkaz

Chcete-li objekty upravit na obrazovce, použijte příkaz. Příkazy lze spouštět více způsoby:

- stiskem posloupnosti kláves příkazu. Dostupné posloupnosti kláves jsou uvedeny kliknutím na tlačítko nápovědy v pravém horním rohu nebo stiskem klávesy `?`.
- Použitím klávesy *Mezerník* napište příkaz, který hledáte nebo procházejte seznamem.
- Kliknutím pravého tlačítka myši na objekt. Zobrazí se kontextové menu se seznamem všech možných příkazů pro vybranou entitu.
- Stisknutím jednoho z vyhrazených tlačítek v horním pruhu karet.

Po spuštění příkaz obdrží veškerý vstup z klávesnice a myši, dokud není nástroj dokončen nebo dokud jej neskončíte stisknutím klávesy `ESC`. Podívejte se na lištu, která se zobrazila ve spodní části plochy, aby jste viděli, které konkrétní klávesy byly stisknuty během tohoto příkazu.

Dále: *základní ovládání: Mezerníková nabídka*

Mezerníková nabídka

Editory v Horizonu jsou založeny na rychlých a intuitivních klávesových zkratkách a podobných klávesových sekvencích. I když je to velmi výhodné pro ty, kteří je používají každý den, mohlo by to být těžkopádné pro ty, kteří právě začali nebo mají dlouhé přestávky mezi jejich elektronickými projekty.

Aby byl přechod mezi začátečníky a zkušenými uživateli snadný Má Horizon EDA mezerníkovou nabídku, která (odtud název) se zobrazí po stisku Mezerníku. V této nabídce jsou všechny Příkazy, ze kterých si můžete vybrat v seznamu. Pomůžou vám rychlejší zkratky a klíčové sekvence jsou uvedeny vedle příkazů. Takže pokud se ocitnete v pozici, kdy musíte znovu a znovu spouštět nabídku mezerníku opět si můžete snadno urychlit práci pomocí zkratek.

Tyto zkratky si můžete snadno přizpůsobit v okně předvoleb.

Dále: *Mřížka*

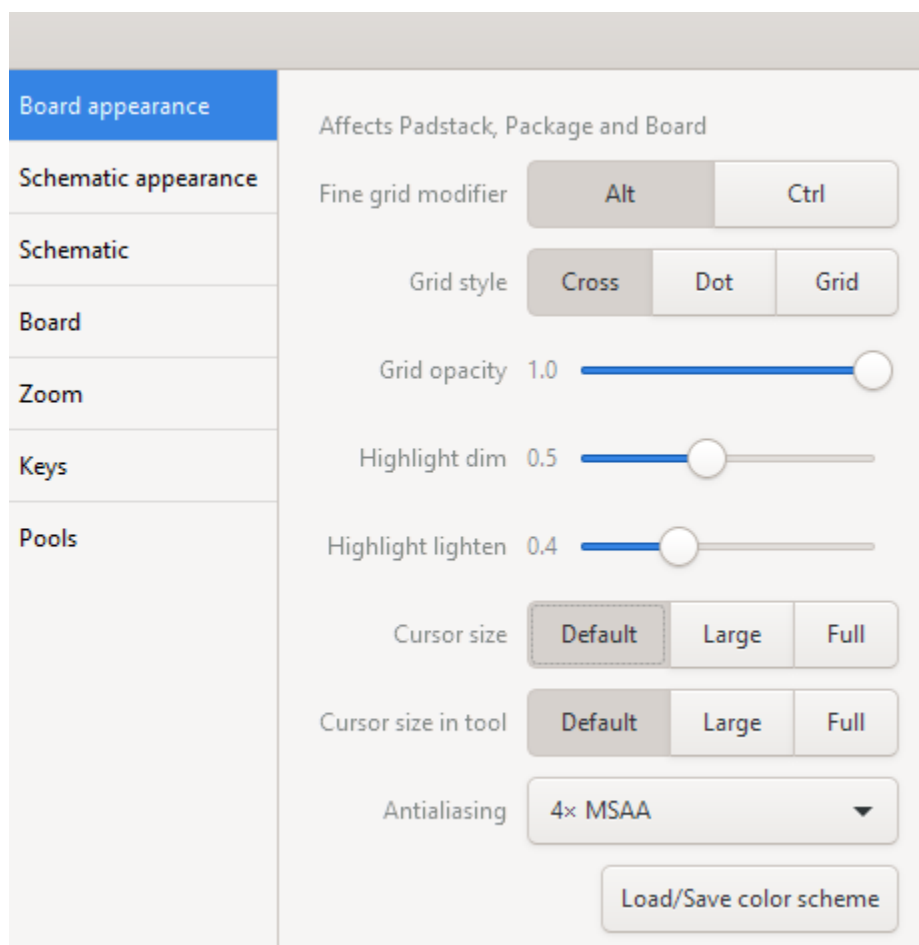
Mřížka

Všechny pohyby v Horizonu se dějí podle mřížky. Podržením klávesy `Alt` nastavíme jemnější mřížku (10krát).

Velikost mřížky můžete změnit změnou hodnoty v zadávacím poli v levém horním rohu okna. Všimněte si, že v závislosti na úrovni zoomu Velikost mřížky může být změněna v násobcích čísla dvě.

Můžete zapisovat základní matematické operace do numerických zadávacích polí, to usnadňuje dělení hodnoty dvěma, vynásobení nějakým určitým násobkem nebo přidáním hodnoty do souřadnice k vytvoření pevného odsazení. Další podrobnosti viz *[Zadávání numerických hodnot](#)*.

Můžete změnit vzhled mřížky a kurzoru v Předvolbách:

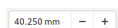


Dále: *Výkres*

Zadávání numerických hodnot

Všechny zadávací numerická pole v horizontu podporují základní matematiku se dvěma operandy a také některé další operace.

Pole vypadají takto:



Tečka i čárka jsou rozpoznány jako desetinný oddělovač bez ohledu na nastavení národního prostředí. Ve výchozím nastavení jsou všechna čísla považována za milimetry. Přípona čísla `s i` nebo `i n` bude považována za palce. Použijte `mi` nebo `mil` pro převod z tisícín.

Navíc je podporována matematika se dvěma fixními operandy, takže můžete:

Sčítat: $1+2$

Odčítat: $1-2$

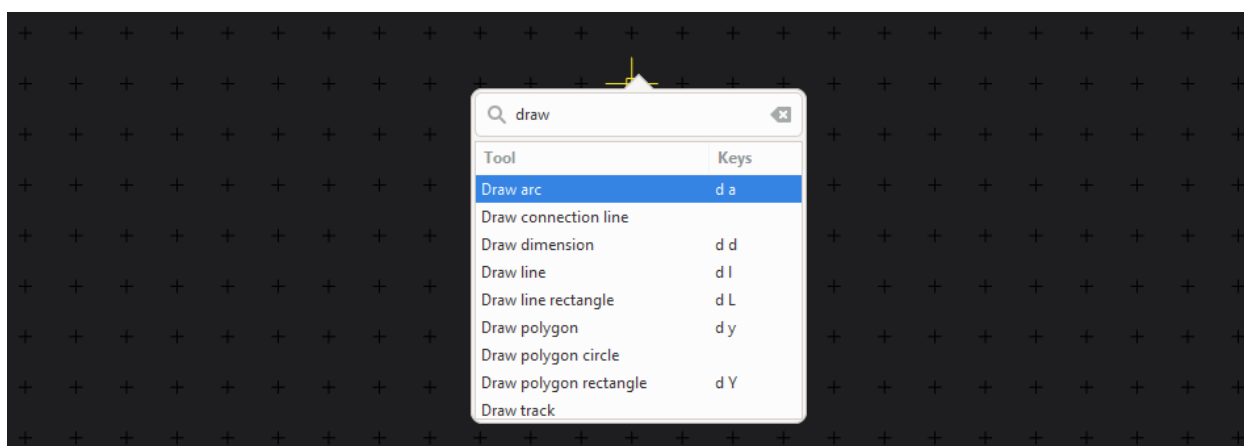
Násobit: $3*2$

Dělit: $3/2$

Průměrovat $\frac{a+b}{2}$: $3|2$

Sčítat milimetry a palce: `1in+2mm`

Různé editory v Horizon EDA sdílejí sadu různých příkazů pro kreslení. Zde vidíte ty dostupné v Editoru desky:



Kreslicí nástroje jsou rozděleny do tří hlavních kategorií:

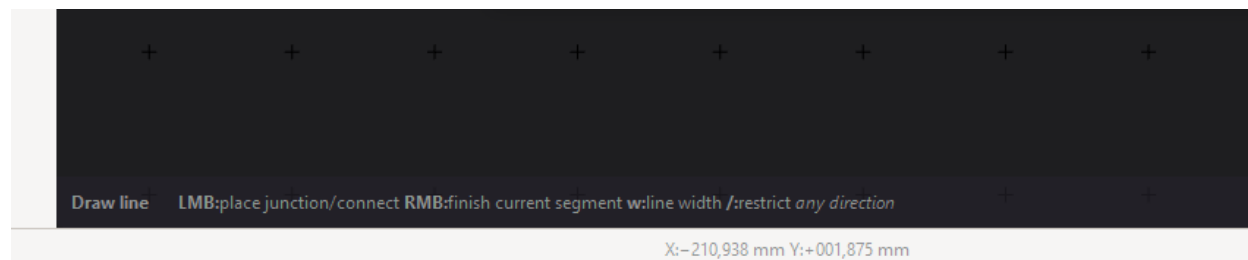
- čáry
- mnohoúhelníky
- speciální (jako stopa, kóta atp.)

Čáry se obvykle používají pro vše vizuální (např. Popisky), zatímco polygony se používají pro všechny objekty, u kterých záleží, aby byl výsledek nakreslen v uzavřeném tvaru (Pájecí místa, obrysy desky, obrys pouzdra a obrys minimálního odsazení atd.)

14.1 Kreslení čáry

Chcete-li nakreslit čáru, jednoduše vyberte příkaz „Draw line“ z mezníkového menu nebo zadejte posloupnost kláves `d l` (myšleno: „draw line“) - jednou kliknete kamkoli, začnete prvním bodem čáry přímo na místě, kde byl váš kurzor. Možná si všimnete, že se bod zachytil do rastru mřížky. Pokud chcete jemnější mřížku, podržte klávesu `Alt`, během umístění bodů. Levým tlačítkem myši umístíte další body, zatímco pravé tlačítko myši (nebo stisknutí klávesy `Esc`) dokončí čáru.

Možná jste si také všimli lišty příkazů v levé dolní části okna editoru:



V panelu příkazů se zobrazí další klávesy, které můžete stisknout pro změnu chování příkazu. V tomto případě můžete stisknout klávesu `w` pro změnu šířky spoje nebo klávesu `/` pro omezení pohybu kursoru jedním směrem.

14.2 Kreslení obdélníka

Abyste ušetřili čas, je zde také příkaz „Draw line rectangle“, který lze také vyvolat stiskem kláves `d L`. Ve výchozím nastavení se nejprve zadává středový bod obdélníku a dále jeden z rohových bodů. Stisknutím `c` můžete toto chování změnit a místo toho zadávat dva diagonálně protilehlé rohové body.

14.3 Kreslení oblouku

Nástroj „Draw arc“ je jednoduchý, nakreslí čárový oblouk pomocí předvolby tři body (v tomto pořadí): počáteční bod, koncový bod a střed. Pro spuštění příkazu můžete také použít posloupnost kláves `d a`. Pokud chcete, aby se oblouk otočil, stiskněte klávesu `e` před kliknutím na středový bod.

14.4 Kreslení mnohoúhelníku (polygonu)

Při kreslení polygonů pomocí `d y` můžete přidávat několik bodů klikáním, dokud nestisknete klávesu `Esc` nebo použijete pravé tlačítko myši. Součástí polygonu může být také oblouk. Stisknutím klávesy `a`. Stejně jako u příkazu „Draw Arc“ nastavte nejprve koncový bod a potom středový bod. Před nastavením středového bodu můžete oblouku převrátit směr pomocí klávesy `e` a nakonec kliknete na koncový bod oblouku.

14.5 Kreslení obdélníka

Podobně jako příkaz „Draw line rectangle“ je zde příkaz „Draw polygon rectangle“. Vyvolejte ho zadáním klávesy `d Y`. Stejně jako u příkazu čára můžete přepínat mezi různými režimy kreslení (Střed/roh) stisknutím klávesy `c`.

Jsou zde však některé rozdíly: poloměr rohu můžete nastavit pomocí stisknutí r a zadáním hodnoty si můžete vybrat prvek podle stisku klávesy d . Tyto prvky se používají k označení vývodu l na pouzdře součástky v hladině obrysu součástky. Můžete procházet různými prvky stiskem klávesy p a nastavte velikost prvku pomocí stisknutí klávesy s a zadání hodnoty.

14.6 Kreslení Kruhu

Pro snadné používání existuje také příkaz „Draw polygon circle“. Nejprve kliknete, na střed kruhu a druhým bodem zadáte poloměr. Poloměr můžete také zadat stisknutím klávesy r a zadání hodnoty.

14.7 Kreslení kóty

Někdy může být užitečné přidat informace o rozměrech některých součástí. Můžete to udělat pomocí příkazu „Draw dimension“. Spustíte ho pomocí stisku kláves d d , výběrem prvního a druhého bodu a přetažením. Pokud je číselná hodnota na nesprávné straně, můžete ji opravit pomocí výběru kóty a její převrácení klávesou e .

Rozměry lze také nastavit na potřebnou délku výběrem konce a z nabídky aktivovat příkaz „Enter Datum“ (stiskněte Enter). Potom můžete přichytit další objekty ke koncovým bodům kóty.

Dále: *Vybírání*

Výběr v editorech Horizontu EDA je velmi jednoduchý a nejsou zde žádná překvapení. Nezáleží na tom, jak jednoduše to vypadá, různé režimy a filtry vám mohou ušetřit dost času.

15.1 Základy

Zpočátku je aktivní režim letmého výběru „hover select“. Jednoduše vybere nejmenší objekt pod kurzorem. Levým tlačítkem myši nebo kliknutím a přetažením pro trvalé vybrání objektů. Stisknete `Esc` nebo klikněte na prázdné místo pro návrat do režimu letmého výběru.

Pokud chcete vybrat více objektů během klikání držte stisknutou klávesu `Ctrl`. Funguje to také pro zrušení výběru objektů, které jste omylem vybrali při přetahování výběrového pole.

15.2 Módy výběru

Můžete změnit chování při tažení levým tlačítkem myši do levého spodního rohu okna editoru. K dispozici jsou tři různé režimy (Box, Lasso, Paint) se čtyřmi různými charakteristikami výběru (Auto, Include Origin, Touch Box, Include Box).

15.3 Výběrový filtr

Pokud musíte vybrat více objektů, někdy může být užitečné jen vybrat určité třídy objektů. Tomu se říká filtr výběru. Můžete jej otevřít pomocí nabídky mezerníku nebo stisknutím klávesy `Ctrl + i`

Kliknutím na políčko položky vyberete pouze tuto položku. Klikněte na zaškrťovací políčko vlevo nahoře vyberte všechny položky.

Dále: *Posouvání a manipulační značka*

Posouvání a manipulační značky

16.1 Posouvání myší

Součástky a spoje můžete přesouvat jejich výběrem a stiskem klávesy `m`, současným stiskem klávesy `Alt` pro pohyb po jemné mřížce. Pohyb vybraného objektu má jeho počáteční (nebo otočný) bod přímo v místě, kde je manipulační značka, když jste spustili příkaz pro přesun.

To lze s výhodou využít, protože nám to umožňuje pohybovat objekty, které byly umístěny na jemnější mřížce, aniž by se přichytily zpět k hrubé mřížce.

16.2 Posouvání klávesnicí

Pomocí kláves se šipkami `←/↑/↓/→` můžete vybraný objekt přesunout o jeden krok mřížky požadovaným směrem. Pokud současně stisknete `Alt` přesunete díl po jemné mřížce (1/10 původní mřížky). Protože je to také příkaz, můžete jej zrušit stisknutím `Esc` a dokončit jej stisknutím `Enter` nebo kliknutím na prázdný bod.

16.3 Přesné posunování

Příkaz „Move exactly“ umožňuje zadat přesný rozměr pro přesun objektu (např. 1 mm) požadovaným směrem. Příkaz můžete spustit stisknutím `M`

16.4 Převrácení stran

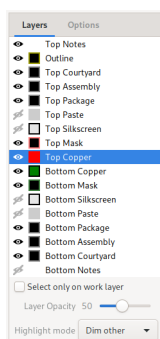
Symbole nebo součástky můžete převrátit stisknutím klávesy `e`. Použijete-li to v editoru desky, je to stejné jako převrácení součástky na druhou stranu desky (spodní/horní)

16.5 Otáčení

Objekty můžete otáčet o 90° , pokud stisknete r nebo vyberete příkaz „rotate“ Pokud chcete objekt otočit o jinou hodnotu (např. 30° nebo 45°) použijte příkaz „Rotate arbitrary“ . Otočný bod rotace je vždy umístění manipulační značky, takže se ujistěte, že je na správném místě

Hladiny (Layers)

Editor desky, pouzdra a pájecího obrazce používá níže uvedený panel k určení jak jsou hladiny zobrazeny.



Vybraná hladina se nazývá „work layer“ a je vždy viditelná a nakreslena nad všemi ostatními hladinami. Použijte Page up/down pro posunutí pracovní hladiny na další nebo předchozí hladinu. Stisknutím kláves 1 vyberete hladinu „Top Copper“ (horní vrstva mědi), klávesa 2 vybere hladinu „Bottom Copper“ (spodní vrstva mědi), klávesami 3 . . . 0 vyberete vnitřní vrstvy mědi, pokud jsou přítomny.

Kliknutím na oko přepínáte viditelnost hladiny. Mějte na paměti, že pracovní hladina je vždy viditelná, i když je nastavena jako neviditelná.

Kliknutí na barevné pole cykluje režimy zobrazení hladiny:

- Plná barva: Hladina je nakreslena obrysy a výplní, překrývající se výplně objektů ve stejné hladině se zobrazí jasněji.
- Plná barva s černým rámečkem: Hladina je nakreslena vyplněná, překrývající se objekty vypadají stejně jako nepřekrývající se objekty.
- Černá s barevným rámečkem: Hladina je nakreslena pouze jako obrysy.
- Pruhované: Stejně jako první, ale oblasti jsou spíše pruhované než vyplněné.

Chcete-li spustit editor schémat, klikněte v manažeru projektu na „Top Schematic“. Podobně jako jiné grafické editory, schematický editor v Horizon-EDA je založen na Interaktivním manipulátoru. Chcete-li umístit součástky, použijte příkaz „Place part“ nebo prohlížeč součástí v hlavním projektu. Ten je mnohem pohodlnější, protože může být trvale otevřený. Začněte kreslit propojení (vodiče) schematu stisknutím Klávesy „n“.

18.1 Umístění součástek

Chcete-li umístit součástky na desku, otevřete prohlížeč součástí buď zadáním `p p` (place part) nebo kliknutím na odpovídající ikonu v hlavním okně projektu. Pokud byla součástka již umístěna, může být nahrazena součástkou stejné entity výběrem a použitím tlačítka „Assign part“ v okně prohlížeče součástek.

18.2 Spoje a segmenty spojů (Nets)

Na rozdíl od některých jiných schématických editorů Horizon-EDA si eviduje jednotlivá propojení a nejde jen o nakreslenou čáru (vodič), která se nakonec při generování seznamu spojů (netlistu) transformuje na vodič (spoj). Propojení může být reprezentováno jedním nebo více segmenty vodičů. Segment spoje je sada propojení vodičů, uzlů, vývodů atd. všechny propojené vodiči (čárami spoje). Editor sleduje, které vodiče patří do kterého segmentu a oznamuje, když se operace chystá sloučit dva segmenty.

Když se podíváte na spoj v editoru vlastností po výběru spoje, spoj v editoru vlastností je „celý“ spoj, nejen segment spoje. Proto přejmenování spoje nemění propojení. Pro připojení vývodu na segment jiného spoje, použijte nástroj pro přesunutí segmentu do jiného nebo nový spoj „Move net segment to other/new net“.

Štítek spoje (Net label) pouze zobrazuje název spoje, ke kterému je připojen, ale nemění ho. Upozorňuje vás na ne konzistenci v systému schématu, které by mohlo mít za následek nežádoucí propojení, editor schématu umístí varování na porušené položky.

18.3 Symboly napájení (Power Nets)

Nejjednodušší způsob vytvoření symbolu napájení ve schematu je použít příkaz pro správu napájení „Manage Power Nets“. Příkaz je k dispozici v menu hamburger. Poté použijte příkaz „place power symbol“, pro umístění symbolu napájení pro tento spoj. Symboly napájení „napájí“ jejich vodiče a připojené segmenty. Můžete si vybrat ze tří stylů napájecích symbolů ve výše uvedeném příkazu. Symboly antény a tečky lze umístit buď nahoru nebo dolů. Symbol GND může směřovat pouze dolů.

18.4 Sběrnice (Buses)

Pro seskupení souvisejících vodičů použijte sběrnice (Buses). Po vytvoření sběrnice přidejte vodiče. Můžete buď přiřadit existující vodiče nebo nově pojmenovat automaticky vytvořený vodič kliknutím na tlačítko se šipkou vedle ní.

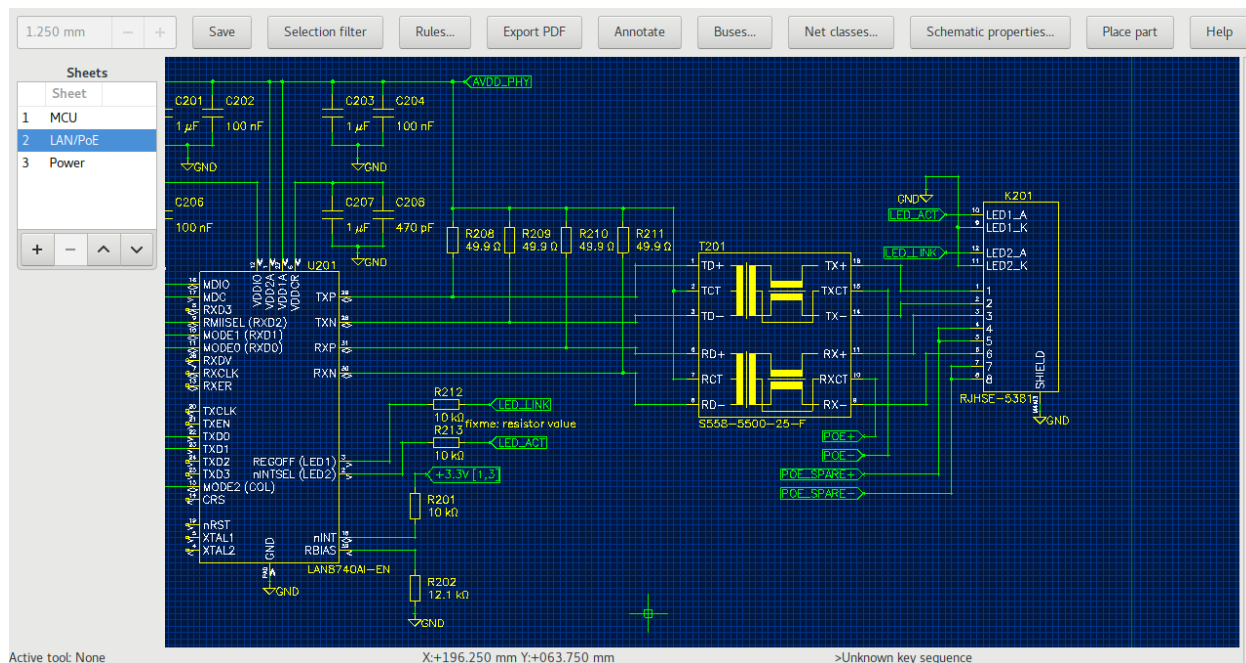
18.5 Diferenciální páry (Diff. pairs)

Chcete-li vytvořit diferenciální pár, vyberte dva vodiče, které chcete spárovat a spusťte příkaz „Set diff. pair“. Můžete také vybrat jeden vodič a budete požádáni o druhý. Chcete-li oddělit vodiče, použijte příkaz „Clear diff. pair“. Doporučuje se přiřadit oběma vodičům stejnou hladinu (Netclass) např. „100diff“, abyste jim mohli snadno nastavit stejná pravidla.

18.6 Přímě na desku (To board)

Chcete-li usnadnit umístění pouzder součástek na desku plošných spojů DPS (PCB), jednoduše vyberte požadované symboly a stiskněte „To board“. Tím se přepnete na editor desky dále spusťte nástroj „place package“ s pouzdry pro vybrané symboly. Možná bude nutné nejdříve znovu načíst seznam spojů (netlist) v Editoru desky, aby editor desky načel nové komponenty.

18.7 Snímek obrazovky



Editor desky plošných spojů DPS (PCB)

Chcete-li spustit editor desky plošných spojů (dále jen desky), klikněte na „Board“ ve správci projektu. Stejně jako všechny ostatní grafické editory je i editor desky v Horizon-EDA založen na Interaktivním manipulátoru

Editor desky si udržuje interní kopii seznamu spojů (netlist). Chcete-li aktualizovat seznam spojů, stiskněte „Save“ (Uložit) v editoru schémat pro zápis seznamu spojů na disk, pak klikněte v editoru desky na „reload netlist“ pro aktualizaci seznamu spojů nebo znovu otevřete editor desky.

19.1 Vyplněné oblasti (Planes)

Pro přidání vyplněných oblastí nejprve nakreslete do hladiny spojů (copper) mnohoúhelník (polygon) požadovaného tvaru. Poté pomocí nástroje „Add plane“ přiřadíte oblast. Pro zobrazení vyplněných oblastí je nutné aktualizovat tyto oblasti pomocí příkazu „Update all Planes“. Oblasti s nižší prioritou se vyplní jako první.

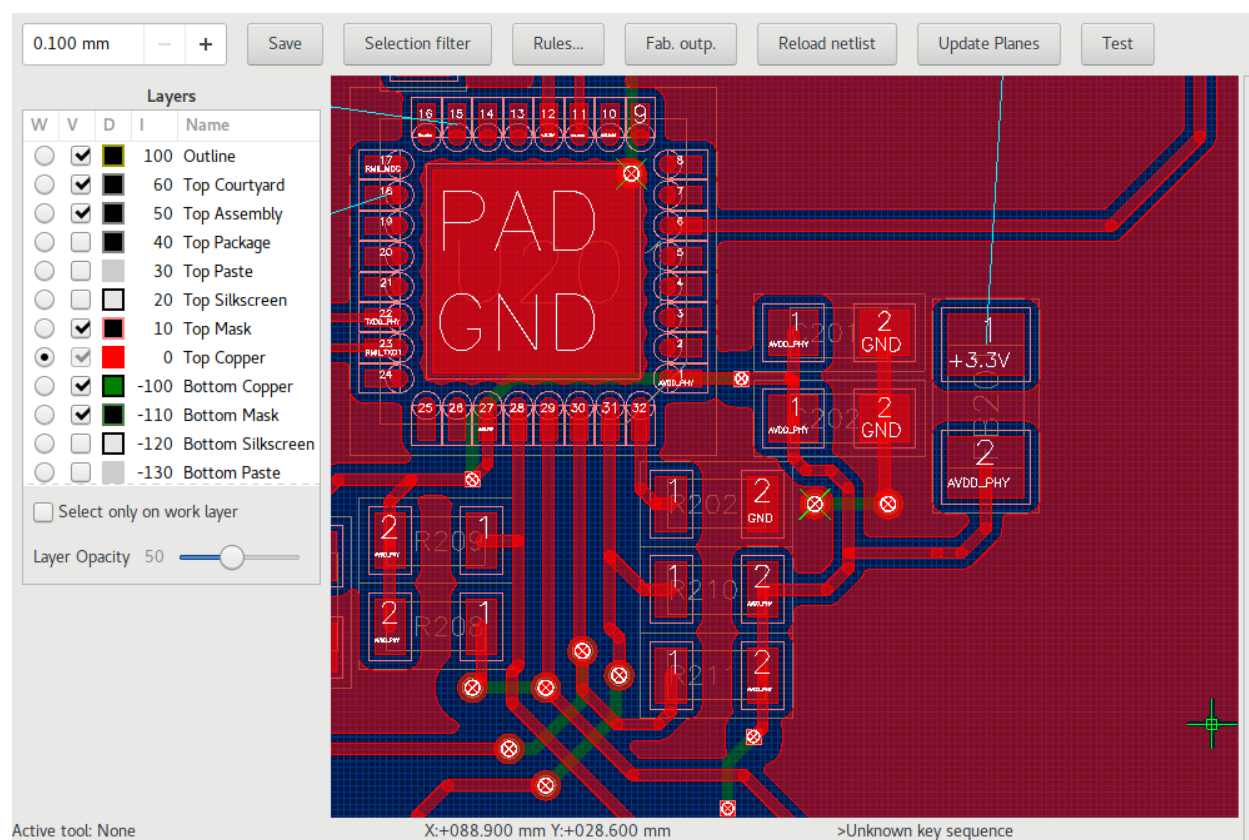
19.2 Prokovené otvory (Vias)

Před vložením prokoveného otvoru vytvořte výchozí pravidlo pro tyto otvory v okně pravidel „Rules“ a přiřadíte jim požadovaný tvar (padstack).

19.3 Diferenciální páry

Jedná se o speciální útvar trasy signálového páru se stanovenými parametry vzájemných elektrických vlastností. Chcete-li vytvořit diferenciální pár, podívejte se na *Diferenciální páry*. Před trasováním diferenciálního páru, vytvořte pravidlo určující šířku stopy a mezeru tohoto páru. Chcete-li vést trasu, použijte nástroj „Route diff“.

19.4 Snímek obrazovky



Kopírování rozmístění a umístění

20.1 Motivace

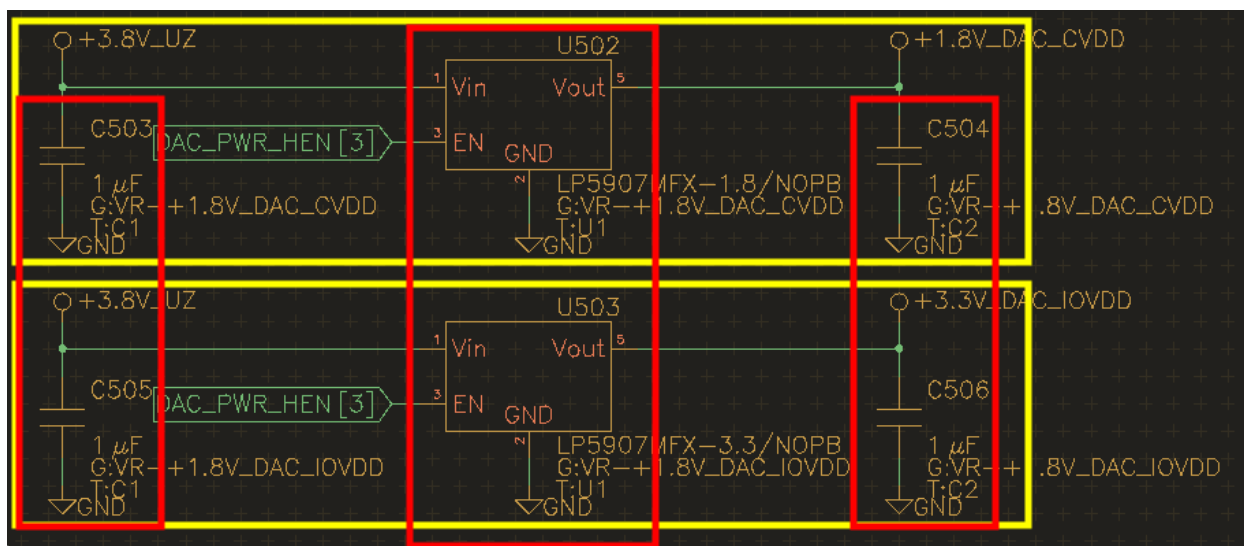
Návrh desky plošného spoje často obsahuje podobné, ale ne shodné celky jako jsou např. regulátory napětí. Nebylo by hezké, kdyby stačilo udělat rozmístění jednou a potom zkopírovat do ostatních kopií? Technologie Horzion-EDA vám to umožní v jednoduchém dvoustupňovém procesu.

20.2 Skupiny a značky

Aby tato funkce fungovala, musíte nejprve sdělit Horizonu-EDA, jak spolu komponenty souvisí. Toho je dosaženo přiřazením skupin a značek ke komponentám. Každá sekce, tj. Všem komponentám spojeným s jedním regulátorem napětí, přidělte jednu skupinu. Vyberte všechny symboly jedné sekce a pomocí příkazu „Set new group“ přiřadte všechny do nové skupiny. Pro zviditelnění skupin a značek na schematu použijte „Toggle group & tag visibility“. Skupina komponent a značka se poté zobrazí pod referenčním označením.

Abychom odlišili v Horizonu-EDA související komponenty v každé skupině, dostanou přiřazené shodné značky. Protože nově umístěná součást již bude mít shodnou značku a skupinu a značky zůstanou zachovány při kopírování / vkládání jiné kopie stejného okruhu budou pravděpodobně mít vhodné značky již nastaveny. Chcete-li změnit značku na součásti, použijte příkaz „Set tag“.

Až budete hotovi, schéma by mělo vypadat zhruba takto (obdélníky přidány pro názornost). Všechny komponenty uvnitř žlutého obdélníku patří do stejné skupiny, vše uvnitř červeného pole patří ke stejné značce.



Chcete-li se ujistit, že jste udělali přiřazení správně, můžete použít akci „Highlight group/tag“.

20.3 Deska

Umístěte a trasujte nějakou skupinu jako obvykle.

20.3.1 Kopírovat umístění

Pro každou skupinu umístěte pouzdro součástky tak, aby ostatní pouzdra součástek byly odkázány na požadované místo a umístěte okolo všechny ostatní pouzdra součástek. Poté vyberte všechny součástky skupiny, které chcete znovu použít a spusťte nástroj „Copy placement“. Klikněte na referenční pouzdro (kterékoli pájecí místo nebo středový bod) v již umístěné skupině a všechny vybrané součástky budou umístěny shodným způsobem.

20.3.2 Kopírování rozvržení

Vyberte všechny spoje (ostatní objekty budou ignorovány), které chcete kopírovat v natrasované skupině, spusťte příkaz „Copy tracks“ a klikněte na libovolné pouzdro (libovolné pájecí místo nebo středový bod) v cílové skupině.

Zpětné úpravy (Backannotation)

Někdy můžete potřebovat propojit vývody součástek na základě jejich umístění na desce, jako například konektory nebo vývody FPGA. Nebylo by krásné, pokud by jste mohli definovat tato spojení přímo v editoru desky bez přechodu tam a zpět mezi editory desky a schémat pro každý spoj? S Horizonem-EDA je to možné!

21.1 Jak na to

Pomocí příkazu „Draw connection line“ (spustit přes mezerník) propojte jednotlivá pájecí místa (plošky), které požadujete propojit. Potom použijte funkci „Backannotate connection lines“ pro zpětnou změnu v editoru schemat. Nově vytvořená připojení se zobrazí jako prázdné šítky nového přerušeného spoje. Po uložení schématu a aktualizaci seznamu spojů (Reload netlist) v editoru desky, budou spoje automaticky nahrazeny naznačenými spoji, které je nutno následně ručně propojit stopou mědi. Pozn. Pro správnou činnost této funkce je nutné mít spuštěn editor desky a editor schemat současně.

21.2 Omezení

Protože tato funkce není přidána příliš dlouho, některé možnosti nejsou zatím podporovány:

- Nelze propojit dva existující spoje
- Řetězové spoje (propojení dvou spojů s jedním vývodem) nepřinesou očekávaný výsledek

Pravidla

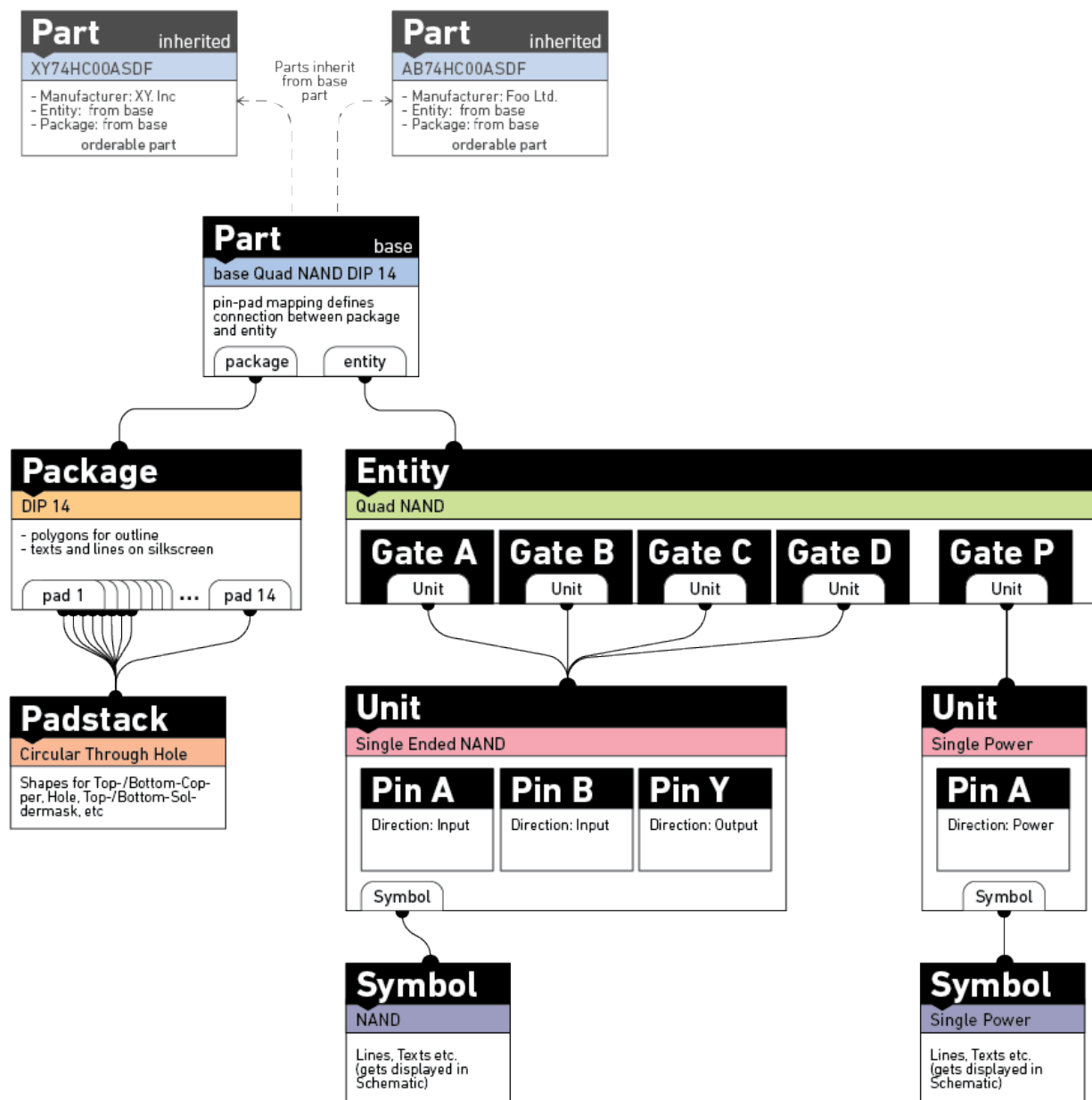
Horizon-EDA používá pravidla pro specifikaci vazeb pro DRC (Design rule checking) pravidla kontroly návrhu desky i pro vstup různých nástrojů, jako je interaktivní trasér. Pravidla jsou vyhodnocena shora dolů a použije se první pravidlo odpovídající všem kritériím. Je tedy na vás, abyste se ujistili, že pravidla jsou uspořádána od konkrétnějších na méně konkrétní. Vždy je dobré mít obecné pravidlo úplně dole.

Proč fond knihovny

Takže co vlastně je fond knihovny (Pool)? Mnoho aplikací EDA uspořádává pouzdra, symboly a podobné komponenty v knihovnách. Často jsou chaotické a správa verzí je obtížná, protože mnoho nezávislých prvků je sloučeno do jediného souboru. Zejména pozdější spolupráce je obtížná.

V Horizonu neexistují žádné knihovny. Místo toho všechny neprojektové prvky (symboly atd.) jsou uloženy ve fondu. Podobně jako „centrální knihovna“, tento přístup je běžnější pro firemní aplikace EDA.

Nicméně nové však je, že součástky v tomto fondu jsou *složený* z více dalších prvků, které odpovídají různým přirozeným vlastnostem součástek:



Například můžete definovat jednu entitu „Čtyřnásobného NAND hradla“ a znovu použít pro každou nová součástku obsahující toto NAND, aniž byste museli znovu a znovu definovat Gates (Brány). Jednotka Čtyřnásobného NAND hradla se skládá z několika jednotek samostatných členů NAND a jedné jednotky napájení. Pokud si nyní přejete vytvořit Dvojitou entitu NAND, stačí znovu použít již existující jednotky a to vám zaručuje soulad s ostatními součástkami NAND ve vašem fondu.

Každý z těchto prvků, které tvoří součástku, je uložen v jediném json souboru v příslušné složce, tj. /symbols, /entities, /units, /parts atd. Přesné umístění v těchto složkách není důležité, pouze musí být soubor json uložen ve správné složce: Symboly v /symbols, jednotky v units a tak dále. Pouze soubory, které mají příponu „.json“ lze zpracovat aktualizátorem fondu. Aby bylo vyhledávání součástek pohodlnější, metadata všech json Souborů jsou agregovány do databáze sqlite. Aktualizaci zajistí tlačítko ‚Update pool‘ ve správci fondu.

Přirozeně je fond se zaměřením na skladbu organizován pomocí značek namísto hierarchického systému, protože ten často vede ke (zbytečnému) zmatku v hlediscích, jako je to, zda se mají díly seskupit podle výrobce nebo dalších

atributů.

Přestože si můžete vytvořit svůj vlastní fond, důrazně se doporučuje používat fond na adrese <https://github.com/horizon-eda/horizon-pool/>. Chcete-li do něj přidat nové díly, jednoduše odešlete žádost o přidání (merge request). Viz také: :doc:`Přispívání do fondu <pool-contribute>`

Elementy součástek

Výsledná součástka se skládá z více prvků, které jsou uloženy nezávisle na sobě ve fondu.

24.1 Součástky (Parts)

Na vrcholu struktury fondu je součástka (Part). Chcete-li se vyhnout zdvojení, součástka může zdědit svoji definici z jiné součástky. To je určeno pro použití ve skupině součástí, které se liší pouze v některých vlastnostech jako odpor nebo výstupní napětí pro pevné regulátory napětí. Každá součástka může být doplněna parametrickými daty, aby bylo vyhledávání snadnější. Tato funkce je zatím ve vývoji, protože není k dispozici žádné uživatelské rozhraní pro součástky založené na parametrických datech. Součástky také obsahují mapování Entity vývodů k pájecím obrazcům pouzder.

Kromě dat je tato součástka spojena jak s entitou, tak s pouzdrem. Součástka uchovává, jaké vývody každé brány na mapě entity mají pájecí místa v pouzdře (např. v případě NAND by mapovaly vstup A na odpovídající vývod na pouzdře DIP 14).

24.2 Pouzdra (Packages)

Pouzdro definuje průmět (otisk) půdorysného tvaru součásti na desku plošného spoje. Pokud výrobce součástky definuje doporučený tvar průmětu, použijte tento. Používejte pouze obecné pouzdra, pokud odpovídající neexistují. Podrobnosti o pouzdrech viz. [Vytvoření pouzdra součástky](#).

24.3 Entity (Entities)

Entita je reprezentace součásti v seznamu spojů (netlist). Díly, které jsou logicky stejné jako např. různé typy konektorů USB proto mohou všechny sdílet stejnou entitu, např. „USB konektor se stíněním a ID“. Vlastní součástky se sestávají z jedné nebo více Jednotek (Units).

24.4 Jednotky (Units)

Jednotka ve skutečnosti definuje logické vývody součástky. Pouze pro součástky, které sestávají z jednoho celku „gate“ jako např. regulátory napětí, jejich entity jednoduše odkazují na jednu jednotku (Unit). Pro části sestávající z více celků „gate“ jako duální operační zesilovač nebo velký mikroprocesor, každý celek „gate“ odkazuje na jednu jednotku. Mít jednotky oddělené od entit umožňuje sdílení více entit stejnou jednotkou. Předpokládá se tedy, že entita jednoho logického celku „gate“ může odkazovat na stejnou jednotku např. čtyřikrát. Kromě jména má vývod i směr (pro ERC) a volitelně alternativní názvy vývodů (pinů) pro popis vývodů, které mají více funkcí, jak je běžné např. u Mikroprocesorů.

24.5 Symboly (Symbols)

Symbol se používá ve schématu k reprezentaci jednotky. Na rozdíl od jiných EDA aplikací, symbol právě zobrazuje vývody z jeho jednotky a nedefinuje je.

24.6 Padstacks

V Horizonu-EDA pájecí obrazec (pad) součástky odkazuje na definici pájecího místa (padstack) příslušného vývodu, která určuje tvar pájecího místa (plošky). V programu existují dva druhy pájecích míst: Globální pájecí místa jsou k dispozici pro použití na všechna pouzdra součástek a měly by pokrývat většinu případů použití. V případě pouzdra součástky vyžadující speciální speciální pájecí obrazec, je možné jej vytvořit pomocí tlačítka „Create padstack for package“ na záložce pouzdra „Packages“ ve správci fondu.

Pájecí místo se skládá z těchto částí:

- Obrazec měděné vrstvy
- Obrazec pájecí masky
- Obrazec pájecí pasty
- Otvor (volitelné)

Upřednostňovaným způsobem definování geometrie pájecích míst je použití takových tvarů, které jsou vhodné pro export do gerber souboru. Pro se souhlasení více než jednoho parametru pouzdra lze použít pájecí místa, která mění svoji velikost. Lze využít také globální parametry specifické pro aplikaci, jako je rozšíření pájecí masky a zúžení masky. Chcete-li získat představu o tom, jak to doopravdy funguje podívejte se na globální pájecí místa ve fondu knihovny.

Chcete-li použít parametry na geometrii pájecího místa, je to popsáno v kapitole [Parametry programu](#), které se postarají o změnu parametrů tvarů pájecích míst.

Správce fondu (Pool manager)

Správce fondu a Průvodce součástmi pomáhají se správou komponent jako jsou symboly, entity a součástky ve fondu. Pravděpodobně budete používat Správce fondu pro vytváření nových dílů. Chcete-li otevřít správce fondu, spusťte `horizon-ed` („`.exe`“) a vyhledejte soubor `pool.json` fondu, který chcete upravit. Podle toho, jakou komponentu chcete vytvořit, je k dispozici několik pracovních postupů:

25.1 Zdědění nové součásti z existující součásti

Když součástka, kterou se chystáte vytvořit, již existuje v jiné variantě (jiná hodnota, nebo jiný teplotní rozsah), ale jinak shodná, tak by nová součástka měla být zděděna ze stávající součástky. Chcete-li to provést, vyberte požadovanou základní součástku na záložce „Parts“ a klikněte na „Create Part from Part“. Po zadání umístění souboru nové součástky se zobrazí Editor součástek. Zrušte zaškrtnutí možnosti „zdědit“ (inherit) pro atributy, které chcete změnit a uložte novou součástku.

25.2 Vytvoření nové součásti ze stávající entity

Tento pracovní postup je vhodný, pokud již entita pro novou součást existuje. Odporů nebo LED v nestandardních pouzdrech jsou typickým příkladem. Pro vytvoření nového pouzdra viz [Vytvoření pouzdra součástky](#). Na záložce „Parts“ klikněte na „Create Part“ pro vytvoření nové součásti. Potom zadejte entitu, pouzdro a umístění souboru součástky, kterou můžete pozměnit a mapovat pájecí místa na vývody v Editoru součástí.

Máte-li vhodnou entitu a pouzdro, můžete pokračovat a vytvořit součástku. Na kartě „Parts“ klikněte na „Create Part“ a vytvořte novou součástku. Poté, co zadáte entitu i pouzdro a umístění součástky, otevře se Editor součástek a můžete součástku upravit a namapovat vývody Entity do pájecí obrazec pouzdra.

25.3 Vytvoření zcela nové součásti

Mnoho součástek, jako jsou mikroprocesory (MCU), FPGA, ADC a další zázraky dnešního světa vyžadují vytvoření nových jednotek a entit. To by bylo manuálně velmi zdlouhavé, v tomto vám může pomoci průvodce součástmi (Part

Wizard). Poté, co vyberete pouzdro součástky (pro vytváření pouzder součástek viz [Vytvoření pouzdra součástky](#)) na záložce „Packages“ klikněte na „Part Wizard...“. Budete uvítáni seznamem všech pájecích míst pouzdra.

Vyplňte názvy vývodů podle katalogového listu součástky. Vložte pouze primární jméno vývodu (např. PB5) na MCU do sloupce zcela vlevo a vložte všechny ostatní názvy (jako UART0_TX, TA0) oddělené mezerou do sloupce „Alt. names“. Pokud je vaše součástka *opravdu* velká (jako FPGA nebo velký MCU), může potřebovat, aby se ve schématu zobrazil více než jeden symbol. Vyberte všechny vývody, které chcete mít se stejným symbolem a typem v alternativním názvu vývodu. V případě, že je více podložek elektricky identických (např. několik vývodů GND), můžete je seskupit jejich výběrem a kliknutím na tlačítko „Propojit podložky“ na dolním panelu nástrojů. Tímto způsobem bude pro vybrané položky vytvořen pouze jeden vývod.

U opravdu velkých součástek s více než 100 vývody může být ruční vkládání příliš zdlouhavé. Chcete-li se tomu vyhnout, je možné použít průvodce součástmi (Part wizard) a importovat názvy vývodů ze souboru s příponou json. Tento soubor může být generován např. nějakým skriptem v jazyce Python nebo podobném. Struktura souboru json by měla vypadat takto:

Soubor CSV pro import vývodů může vypadat takto:

```
1,PB0,bidirectional,Main,TXD,SDA
2,PB1,bidirectional,Main,RXD,SCL
3,TDI,input,Main
4,TDO,output,Main
5,GND,power_input,Main
```

Celý formát CSV je pad,pin,direction,gate,alt1,alt2,...

Definování stejných vývodů ve formátu json lze provést pomocí:

```
{
  "1": {"pin": "PB0", "alt": ["TXD", "SDA"], "gate": "Main"},
  "2": {"pin": "PB1", "alt": ["RXD", "SCL"], "gate": "Main"},
  "3": {"pin": "TDI", "direction": "input", "gate": "Main"},
  "4": {"pin": "TDO", "direction": "output", "gate": "Main"},
  "5": {"pin": "GND", "direction": "power_input", "gate": "Main"}
}
```

Pro CSV i json položky se stejným pin-gate budou sloučeny. Další platné hodnoty pro direction jsou open_collector, passive a not_connected.

Vhodnými zdroji údajů se jmény vývodů a pájecích míst jsou:

- IBIS modely
- BSDL soubory
- katalogové listy PDF dané součástky

Jakmile vyplníte názvy vývodů, klikněte vlevo nahoře na „Další“ pro postup na další obrazovku. Vyplňte položky podle vaší součástky. Pokud si nejste jisti, co tam má být, podívejte se na stávající součástky ve fondu. Pokud je vaše součástka k dispozici vícekrát v téměř identických variantách, které se liší pouze v aspektech jako teplotní rozsah nebo možnost balení (Páska / Role, Trubka, atd.) vytvoří součástku, kterou se chystáte použít. Pro vytvoření dalších variant postupujte podle pokynů v horní části. Postarejte se o správné zadání umístění jednotek / symbolů / entit a částí tak, aby končily v podsložkách jejich příslušné složky ve fondu.

Pro každý celek (gate) klikněte na „Edit Symbol“ pro spuštění interaktivního manipulátoru vytvoříte symbol pro tuto jednotku (unit). Použijte příkaz „Map pin“ pro umístění vývodů do symbolu a „Draw line rectangle“/„Edit line rectangle“ pro nakreslení těla symbolu. Nezapomeňte dát symbolu smysluplné jméno a umístěte texty „\$REFDES“ a „\$VALUE“.

Když nakreslíte všechny symboly a vyplníte všechny údaje, kliknutím na „Finish“ konečně vložíte součást do fondu.

25.4 Kam ukládat komponenty

Při vytváření nových symbolů, součástek a podobně správce fondu / průvodce tvorbou součástek vás dříve nebo později požádá o název souboru nebo složky (v případě pouzder) k uložení nové součástky. Technicky, vámi specifikovaná cesta pouze musí splňovat dva požadavky:

- Musí být ve správné složce nejvyšší úrovně, tj. pro každou schematickou značku musí být někde ve složce /symbols atd. Pájecí obrazec specifický pro a pouzdro musí být umístěno do složky /padstack.
- Název souboru musí mít příponu .json

Pro získání představy jak to všechno prakticky vypadá se podívejte do veřejného fondu na adrese <<https://github.com/carrotIndustries/horizon-pool/>>‘__

25.5 Databáze fondu

Fond uchovává údaje (názvy souborů, UUID, jména atd.) V SQLite databázi usnadňující vyhledávání. Normálně si správce fondu aktualizuje databázi pokaždé, když se něco změní. Nicméně, pokud externě manipulujete / odstraňujete soubory, musíte kliknout na „Aktualizovat“ (Update Pool) databázi, která zahrne provedené změny.

Vytvoření pouzdra součástky

Rozložení desky plošných spojů může být tak dobré jak dobře jsou vytvořeny pájecí obrazce (footprints) pro jednotlivá pouzdra součástek, které program používá, proto je důležité vytvořit vysoce kvalitní podklady obrazců těchto obrazců (footprints).

V Horizonu-EDA se pouzdro součástky se skládá z těchto částí:

- Pájecí místa (pads), na které se součást připájí
 - Spoje - měděné vrstvy tras jednotlivých propojení v jednotlivých (horní / dolní / vnitřní) vrstvách
 - Otvory (pro díly pro pájení do otvorů TH (Thru hole))
 - Vynechání nepájivé masky
 - Masky pájecí pasty
- Obrys pouzdra součástky (Package outline)
- Montážní obrys a referenční označení (Assembly outline)
- Textový popis součástek (Silkscreen)
- Obrys zástavby součástky (Courtyard outline)

26.1 Pájecí obrazce (pads)

Každý pájecí obrazec je definován několika pájecími místy (padstack) uspořádanými do různých tvarů a parametry aplikovanými na tento tvar. Podrobnosti o pájecích místech viz. *Pájecí místa / plošky (Padstacks)*. Tyto pájecí obrazce jsou pravděpodobně nejdůležitější vlastností pouzdra součástky, je vhodné s nimi začít. Pájecí místa můžete umístit ručně pomocí příkazu „Place pad“ nebo je nechejte umístit automaticky podle běžně používaných vzorů pomocí příkazu „Footprint gen.“, který je k dispozici na horním panelu. Po umístění pájecích míst mají stále svou výchozí velikost, která patrně není ta, co potřebujete. Chcete-li velikost změnit, vyberte parametry, které chcete upravit a pomocí nástroje „Edit pad“ přidejte parametry do pájecího místa. V závislosti na vybraném tvaru pájecího místa jsou přiřazeny určité parametry. Nejčastěji se používají šířka a výška pájecího místa. Použijte tlačítko zaškrtnutí vedle parametru pro parametry, které se mají použít na všechny vybrané podložky.

26.2 Obrys pouzdra součástky (Package outline)

Obrys pouzdra se používá k vizualizaci obrysu součásti. Dá se říct, že by tedy měla sledovat jmenovité rozměry součásti. Můžete použít nástroj „import DXF“ pro import výkresu DXF získaného ze STEP modelu nebo jinak. Protože účel obrysu součásti je čistě vizuální, můžete použít buď čáry, nebo mnohoúhelníky. Vývody přidejte, pouze pokud výrazně přispívají ke vzhledu součásti.

26.3 Montážní obrys

Montážní obrys končí na výkresu sestavy (zatím není implementováno) a jeho účelem je pomoc při montáži a kontrole PCA. Montážní obrysová vrstva tedy obsahuje pouze tyto položky: Mnohoúhelník označující obrys součásti, volitelně vývody, pokud se výrazně přispívají ke vzhledu součásti a referenční označení součásti. Na rozdíl od obrysu pouzdra je montáž pouze hrubá náhrada tvaru součásti. Musí zahrnovat nějaký druh vizuálního označení umístění prvního vývodu součásti. Použijte příkazy „Draw polygon rectangle“ a jeho možnosti dekorace pro kreslení takového obrysu. Pro referenční označení, vložte text obsahující „\$ RD“ v takové velikosti, aby se vešel do obrysu sestavy, i když je předpona o 4 číslice delší.

26.4 Textový popis součástek

Účelem textového popisu je objasnit umístění dílů a orientace při ruční montáži a vizuální kontrole, také by měla pomoci pokud je součást citlivá na orientaci, použijte nějakou značku prvního vývodu. Nepoužívejte tečku pro označení prvního vývodu, místo toho sklopte nebo prodlužte grafickou značku. Doporučená tloušťka čáry je 0,15 mm. Také vložte text „\$ RD“, s tloušťkou čáry 0,15 mm do hladiny popisu.

26.5 Obrys zástavby součástky

Obrys zástavby označuje prostor potřebný pro plochu kolem součástky, která nesmí být obsazena jinými součástkami, aby byl ponechán dostatečný prostor pro montáž. Vzhledem k tomu, že velikost obrysu zástavby je třeba upravit v závislosti na uživatelských výrobních požadavcích, musí být nastaven pomocí parametru programu. Při zvětšení obrysu zástavby 0 mm je obrys zástavby (obdélníkový) kolem pájecích míst a obrysu pouzdra součástky. Chcete-li vytvořit obdélníkový obrys zástavby, který lze parametrizovat, proveďte toto:

Pomocí příkazu „Generate courtyard“ vygenerujete počáteční obrys zástavby na 0 mm rozšíření. Pokud to nemá za následek požadovaný mnohoúhelník, použijte příkaz „Draw polygon rectangle“ pro nakreslení počátečního obrysu a nastavení jeho skupiny parametrů na „courtyard“ pomocí editoru vlastností na pravé straně okna.

Otevřete okno „Parameters“ a klikněte na „Insert courtyard program“. Je-li vše jde správně, měl by se přidat program obrysu zástavby a jeho parametr „Courtyard expansion“ nastavený na 0,25 mm.

Contribute to the Pool

Oficiální fond je uložen na adrese <https://github.com/horizon-eda/horizon-pool/> do fondu přispívají jeho uživatelé. Existuje několik způsobů, jak můžete pomoci. Nejprínosnější je odesláním součástí, které jste vytvořili.

Aby byl ve fondu pořádek, přidávejte pouze díly, které můžete skutečně koupit, jejich odpovídající symboly, entity atd. Takže nepřidávejte část nazvanou 7805, místo toho přidejte MC7805BDTRKG od ON Semiconductor. Když jste vytvořili něco, co byste chtěli sdílet, můžete použít *Pool Manager* a nahrát svůj výtvar do oficiálního fondu:

27.1 Používání Integrace GitHub

Aby integrace GitHub fungovala, musí být fond stažen pomocí tlačítka „Download...“ na úvodní stránce správce fondu. Správce fondu naklonuje veřejný fond do složky `.remote` v systému vašeho místního fondu. Pokud vše půjde dobře, neměli byste nikdy zasahovat do této složky. K dispozici jsou dvě operace pro správu vaší místní kopie aktualizace a zahrnutí vašich součástí do veřejného fondu.

27.1.1 Aktualizace fondu

Tento příkaz aktualizuje vaše kopie globálního fondu ve složce `.remote` od posledního potvrzení a zeptá se vás, jaké změny chcete použít ve vašem místním fondu.

27.1.2 Vytvoření žádosti o zahrnutí součástí do veřejného fondu

Nejprve přidejte součástky / entity / atd. do seznamu „items to be merged“, poté vyplňte název a tělo žádosti o zahrnutí. Správce fondu bude automaticky přidávat položky, které jsou nutné, aby nebyly porušeny závislosti. Takže když vytvoříte zcela novou částku s novou jednotkou, entitou a pouzdrům přidejte ji do seznamu pro přidání součástí. Nezapomeňte přidat nové symboly. Poté, co se ujistíte, že to je to, co chcete, klikněte na „Create pull request“. Budete vyzváni k zadání přihlašovacích údajů pro server GitHub.

27.1.3 Pomoc při kontrole

Přidávání součástek je skvělá věc, ale kontrola součástek, které ostatní vytvořit je také dobrá věc. Více očí, které porovnájí součástku s jejím datovým listem sníží šanci na chybu. Pokud jste úspěšně vyrobili DPS s konkrétními součástkami, můžete také informovat o pážitelnosti a to je silný ukazatel toho, že součást nemá žádné kritické chyby.

Sestavení (kompilace) programu na operačním systému Windows

28.1 Nainstalujte program MSYS2

Stáhněte si a spusťte instalátor programu msys2 z <http://msys2.github.io/> Mám testováno pouze s 64bitovou verzí, 32bit by měla fungovat také. Ujistěte se, že cesta, kterou jste vybrali pro instalaci neobsahuje žádné mezery (v názvech složek).

28.2 Spusťte konzolu MSYS

Spusťte položku nabídky Start „MSYS2 mingw 64 bit“, která by se měla zobrazit v okně konzoly. Všechny níže uvedené kroky se týkají toho, co by jste měli napsat do toho okna.

28.3 Nainstalujte aktualizace

Napište

```
pacman -Syu
```

pokud vám řekne, že chce restartovat před restartováním, zavřete okno konzoly a po restartu znovu spusťte program `pacman -Syu`.

28.4 Nainstalujte související součásti

Napište / vložte

```
pacman -S mingw-w64-x86_64-gtkmm3 git base-devel \
mingw-w64-x86_64-yaml-cpp mingw-w64-x86_64-boost \
mingw-w64-x86_64-sqlite3 mingw-w64-x86_64-toolchain \
mingw-w64-x86_64-zeromq mingw-w64-x86_64-glm zip \
mingw-w64-x86_64-libgit2 mingw-w64-x86_64-occe \
mingw-w64-x86_64-podofo mingw-w64-x86_64-libzip --needed
```

Až budete vyzváni, stačí stisknout klávesu Enter. Posad'te se a počkejte, až instalátor dokončí instalaci téměř kompletního linuxového vývojového prostředí.

Než budete pokračovat, můžete přejít do jiné složky. Je to jednoduché zadejte “ cd“, mezeru a přetáhněte složku, do které chcete přejít do okna konzoly.

28.5 Klonovat zdrojový kód programu Horizon-EDA

```
git clone http://github.com/horizon-eda/horizon
cd horizon
```

28.6 Sestavení (kompilace) programu ze zdrojového kódu

```
make -j 4
```

Číslo 4 na konci můžete upravit podle počtu procesorů ve vašem systému pro rychlejší kompilaci. Očekávejte 100% zatížení procesoru (CPU) po dobu několika minut dle výkonu počítače. Z důvodu zapnuté volby vkládání ladících symbolů mají výsledné spustitelné soubory značnou velikost.

28.7 Spuštění programu

Po kompilaci nebudete moci dvakrát kliknout na výsledné spustitelné soubory protože požadované knihovny DLL nejsou ve složce známé systému Windows. Budete muset spustit z příkazového řádku prostředí Mingw například pomocí příkazu `./horizon-eda`. Aby fungovalo stahování fondu, musíte zkopírovat soubor `/mingw64/ssl/certs/ca-bundle.crt` do složky obsahující `horizon-eda.exe`.

28.8 Vytvoření archivu

Chcete-li vytvořit archiv ZIP, jak je dostupný ke stažení, spusťte příkaz `./make_bindist.sh`.

Sestavení (kompilace) programu na operačním systému Linux

Sestavení programu Horizon-EDA v Linuxu je podobně jednoduché jako ve Windows

29.1 Nainstalujte související součásti

Ujistěte se, že máte nainstalované tyto nezbytné související knihovny:

- Gtkmm3 >= 3.20
- cairomm-pdf
- libsvg
- util-linux
- yaml-cpp
- sqlite
- boost
- zeromq
- glm
- libgit2
- curl
- opencascade / opencascade community edition
- zeromq with C++ bindings: <https://github.com/zeromq/cppzmq>
- podofo
- libzip

pro různé distribuce jsou zde předpřipraveny příkazy pro případnou instalaci knihoven. Pro verzi Ubuntu >= 17.04:

```
sudo apt install libyaml-cpp-dev libsqlite3-dev util-linux librsvg2-dev \
    libcairomm-1.0-dev libepoxy-dev libgtkmm-3.0-dev uuid-dev libboost-dev \
    libzmq5 libzmq3-dev libglm-dev libgit2-dev libcurl4-gnutls-dev liboce-ocaf-dev \
    libpodofo-dev libzip-dev
```

V systému Arch Linux:

```
sudo pacman -S yaml-cpp zeromq gtkmm3 cairomm librsvg sqlite3 libgit2 curl \
    opencascade boost glm podofo libzip
```

Na Fedoře 25/26/27:

```
sudo dnf install git make gcc gcc-c++ pkg-config cppzmq-devel OCE-devel\
    gtkmm3-devel libgit2-devel libuuid-devel yaml-cpp-devel sqlite-devel librsvg2-
↳devel\
    cairomm-devel glm-devel boost-devel libcurl-devel podofo-devel libzip-devel
```

Na openSUSE Tumbleweed:

```
sudo zypper in git make gcc gcc-c++ pkg-config cppzmq-devel oce-devel\
    gtkmm3-devel libgit2-devel libuuid-devel yaml-cpp-devel sqlite3-devel librsvg-
↳devel\
    cairomm-devel glm-devel boost-devel libcurl-devel libpodofo-devel binutils-gold_
↳libzip-devel
```

Na FreeBSD 12:

```
sudo pkg install git gmake pkgconf e2fsprogs-libuuid sqlite3 yaml-cpp \
    gtkmm30 cppzmq libgit2 boost-libs glm opencascade podofo libzip
```

29.2 Sestavení (kompilace) programu ze zdrojového kódu

```
make -j 4
```

29.3 Spuštění programu

Výsledné binární soubory jsou samostatné a nevyžadují žádné externí datové soubory jako ikony nebo podobně. `horizon-eda` je hlavní spustitelný program. Spouštějte jej ze složky kde byl sestaven pomocí příkazu:

```
./horizon-eda
```

Parametry programu

Jak již bylo popsáno jinde v dokumentu, Horizon-EDA podporuje parametrizovatelné tvary pájecích míst a (v omezené míře) pouzder součástek. Chcete-li použít dané parametry na existující geometrii, každé pájecí místo a podobné objekty je možné změnit pomocí krátkého makra.

Tyto makra jsou psány ve vlastním zásobníkovém jazyce. Uživatelé kalkulaček HP se měli cítit jako doma. Protože v programu není možné provádět žádné smyčky, budou tyto programy ukončeny v daném čase. Zásobník obsahuje 64bitová celá čísla se znaménkem. Koncepčně roste shora dolů.

30.1 Syntaxe

Na nejvyšší úrovni je program tvořen značkami. Značky jsou oddělené libovolným množstvím mezer.

Typy značek:

- Celá čísla: číslo, volitelně opatřené znaménkem
- Rozměr: číslo s volitelnou zlomkovou částí, s příponou „mm“ plovoucí řádová značka před značkou mm se vynásobí 1×10^6 , od vnitřní jednotka měření programu Horizon-EDA je 1nm
- Matematické operátory, jako například: + - * /
- Řetězce znaků
- Parametry začínají znakem [a končí] jakákoliv značka mezi těmito dvěma znaky bude přidána jako parametr předchozího příkazu

30.2 Obecné příkazy

30.2.1 Nulový operand

`get-parameter` [`<parameter>`] načte paramter a vloží jej na zásobník

30.2.2 Jeden operand

Before the operation, the stack looks like this:

```
. .  
. .  
+---+  
| a |  
+---+
```

Operators:

```
      | pushes  
dup  | a a  
chs  | -a
```

30.2.3 Dva operandy

Before the operation, the stack looks like this:

```
. .  
. .  
+---+  
| a |  
+---+  
| b |  
+---+
```

Operators:

```
      | pushes  
+     | a+b  
-     | a-b  
*     | a*b  
/     | a/b  
dupc  | a b a b (Duplicate coordinate)
```

30.2.4 Tři operandy

Before the operation, the stack looks like this:

```
. .  
. .  
+---+  
| a |  
+---+  
| b |  
+---+  
| c |  
+---+
```

Operators:

```
      | pushes  
+xy   | a+c b+c  
-xy   | a-c b-c
```

30.3 Příkazy pro pájecí místa

Aby program mohl manipulovat s objektem (tvaru atd.), musí být přiřazena třída parametrů. `## set-shape set-shape [<třída parametrů> <form>]` Nastaví tvar na zadaný formulář nebo jej přesune na určenou pozici Platného formuláře:

- `obdélník`, zobrazí výšku, šířku
- `kruh`, zobrazí průměr
- `obround`, zobrazí výšku, šířku
- `pozice`, zobrazí polohu y, x

30.3.1 Zadání otvoru

`set-hole [<třída parametrů> <shape>]` Nastaví díru na specifikovaný tvar z přednastavených možností:

- `kulatý`, zobrazí průměr
- `slot`, zobrazí délku, průměr

30.4 Polygonové příkazy (pájecí místa a pouzdra součástek)

30.4.1 Zadání polygonu

`set-polygon [<třída parametrů> <shape> <x0> <y0>]` Nastaví polygon před připraveného tvaru se středem na (x0, y0) z vybraných možností:

- `obdélník`, zobrazí výšku, šířku
- `kruh`, zobrazí průměr

30.4.2 Zadání vrcholů polygonu

`“ set-polygon-vertices [<třída parametrů> <n_vertices>] “` Načte `n_vertices` souřadnic vrcholů ze zásobníku a vytvoří z nich mnohoúhelník.

30.4.3 Vytvoření polygonu

`expand-polygon [<třída parametrů> <x0> <y0> <x1> <y1> ... <xn> <yn>]` Vytvoří polygon určený hodnotami souřadnic v parametrech načtených ze zásobníku.

30.5 Příklad programu (pro SMD obdélníkové pájecí místo)

```
get-parameter [ pad_width ]
get-parameter [ pad_height ]
dupc dupc
set-shape [ pad rectangle ]
get-parameter [ solder_mask_expansion ]
```

(continues on next page)

(pokračujte na předchozí stránce)

```
2 *  
+xy  
set-shape [ mask rectangle ]  
  
get-parameter [ paste_mask_contraction ]  
2 *  
-xy  
set-shape [ paste rectangle ]
```

31.1 Interaktivní manipulátor

Primárním rozhraním Horizontu je tzv. „Interaktivní manipulátor“. Je to jednotný editor symbolů, schémat, pájecích míst (padstacks), pouzder součástek (packages) a desky plošných spojů (PCB).

31.1.1 Pracovní plocha (Canvas)

Na pracovní ploše se vykreslují objekty, jako jsou symboly schématu, pouzdra součástek nebo jednotlivé spoje. Výstupem vykreslení jsou segmenty čar a trojúhelníky, které jsou následně vykresleny pomocí grafického procesoru (GPU). Chcete-li vykreslit na pracovní ploše jiné než grafické (OpenGL) objekty, pracovní plocha poskytuje nástroje pro získání více informací o tom, co je vykresleno. Zatím program umožňuje výstup ve formátu Gerber, 3D náhled na desku a kontrolu dle pravidel návrhu (DRC - Design rule checking)

31.1.2 Jádru (Core)

Protože některé dokumenty, jako jsou symboly a schémata, obsahují ty samé typy objektu (např. texty), schémata a seznam spojů (netlist) musí být modifikované synchronizovaně, musí dojít k zapouzdření. Jádro je spojovací článek mezi dokumentem, pracovní plochou a příkazy.

31.1.3 Příkazy (Tools)

Pro každou akci, kterou může uživatel udělat, existuje příkaz. Vyvolaný příkaz začne přijímat vstupy z klávesnice a myši a podle funkce upravuje dokument pomocí jádra. V případě potřeby může příkaz vyvolat další dialogy pro vyžádání dodatečných informací od uživatele.

31.1.4 Editor vlastností (Property editor)

Jednoduché úpravy parametrů objektů, jako je šířka čáry, neumožňují přímo jejich nástroje, proto jádro poskytuje rozhraní úprav vlastností. ovládací prvky editoru vlastností jsou automaticky generovány z popisu objektů.

Použití příkazového řádku (CLI Command Line Interface)

Správce projektu a správce fondu do značné míry eliminovali potřebu spouštět interaktivní manipulátor a další nástroje přímo z příkazového řádku, ale je to stále užitečné pro vývoj.

Všechny níže uvedené příkazy vyžadují proměnnou prostředí `HORIZON_POOL`, nastavenou na složku fondu (obsahující soubor `pool.json` a `pool.db`)

32.1 Použití horizon-imp

Režim symbolů:

```
horizont-imp -y <soubor symbolu>
```

Schematický režim:

```
horizon-imp -c <soubor schematu> <soubor bloku>
```

Režim padstack:

```
horizon-imp -a <soubor pájecího obrazce>
```

Režim balíčku:

```
horizon-imp -k <soubor pouzdra součástky>
```

Režim desky:

```
horizon-imp -b <soubor desky> <soubor bloku> <přes složku>
```

32.2 Použití horizont-pool

Většina příkazů `-edit` a `-create` spustí příslušný `$EDITOR`, který vytvoří odpovídající soubor ve formátu YAML.

```
horizon-pool create-unit <unit file>
horizon-pool edit-unit <unit file>
horizon-pool create-symbol <symbol file> <unit file>

horizon-pool create-entity <entity file> [<unit file> ...]
horizon-pool edit-entity <entity file>

horizon-pool create-package <package file>
horizon-pool create-padstack <padstack file>

horizon-pool update #Recreates the pool's SQLite database.
```

Po vytvoření souborů nezapomeňte spustit `horizon-pool update`

32.3 Použití horizont-prj

Pomocí těchto příkazů můžete vytvářet prázdné bloky, schémata atd.

```
horizon-prj create-block <block filename>

horizon-prj create-schematic <schematic filename> <block filename>

horizon-prj create-board <schematic filename> <block filename>
```

Některé funkce Horzion EDA jsou k dispozici v modulu Python pro použití ve skriptech.

33.1 instalace

Modul python není zahrnut v překladu pomocí `build all`. Chcete-li jej sestavit, spusťte překlad pomocí `make build/horizon.so`. To vyžaduje mít instalovány hlavičkové soubory Pythonu verze 3. Potom můžete použít v Pythonu `sys.path` a importovat pomocí `import horizon`.

33.2 Použití

```
import horizon

#open project
p=horizon.Project("/path/to/project.hprj")

#open schematic
sch = p.open_top_schematic()

#export PDF
pdf_settings = sch.get_pdf_export_settings()
pdf_settings['output_filename'] = '/tmp/sch.pdf'
sch.export_pdf(pdf_settings)

#export BOM
bom_settings = sch.get_bom_export_settings()
bom_settings['output_filename'] = '/tmp/bom.csv'
sch.export_bom(bom_settings)

#open board
```

(continues on next page)

(pokračujte na předchozí stránce)

```

brd = p.open_board()

#export gerber
gerber_settings = brd.get_gerber_export_settings()
gerber_settings["output_directory"] = "/tmp/gerber"
brd.export_gerber(gerber_settings)

#export pick&place
pnp_settings = brd.get_pnp_export_settings()
pnp_settings["output_directory"] = "/tmp/pnp"
brd.export_pnp(pnp_settings)

#export STEP
step_settings = brd.get_step_export_settings()
step_settings["filename"] = "/tmp/pca.step"
brd.export_step(step_settings)

#run DRC
rules=brd.get_rules()
#modify rules if needed
rule_ids = brd.get_rule_ids()
#if needed, remove unneeded checks from rule_ids
result = brd.run_checks(rules, ids)

#export 3D rendering (see next section)
exporter = brd.export_3d(1920, 1080) #width, height
exporter.view_all()
exporter.load_3d_models() #optional
exporter.render_to_png("brd.png")

```

Chcete-li dále upravit nastavení exportu, podívejte se na vrácené příkazy. pomocí metody `get_*_export_settings`.

33.3 Použití 3D vykreslování

Použijte `brd.export_3d(1920, 1080)` nebo podobně pro vytvoření `Image3DExporter` objektu

class Image3DExporter

render_to_png (*filename*)

Vykreslit obrázek formátu png

render_to_surface ()

Vykreslit plochu formátu pycairo

Typ návratové hodnoty `cairo.Surface`

load_3d_models ()

Načte 3D modely pokud jsou k dispozici

view_all ()

Obnoví pohled na horní stranu

I

`Image3DExporter` (*vestavěná třída*), [104](#)

L

`load_3d_models()` (*metoda `Image3DExporter`*), [104](#)

R

`render_to_png()` (*metoda `Image3DExporter`*), [104](#)

`render_to_surface()` (*metoda `Image3DExporter`*),
[104](#)

V

`view_all()` (*metoda `Image3DExporter`*), [104](#)