Zadání semestrální práce z předmětu PLO

Vytvořte v jazyce VHDL-93 nebo VHDL-2008 (syntaxe dle standardu IEEE Std. 1076-1993 nebo IEEE Std. 1076-2008) číslicový systém realizující generátor monoskopu s vloženým obrazem z paměti. Generovaný VGA signál musí být minimálně v rozlišení 800x600 obrazových bodů (časování viz příloha) zobrazující svislé barevné pruhy a uprostřed pak obraz 320x240 vkládaný z paměti ROM (případně RAM). Barva každého bodu obrazu je dána 4-bity (tj. 16 barev/bod). Barva prvního pruhu (nejvíce vlevo) je dána vstupem *start_color* (dle *palette index*). Každý další pruh má barvu podle *palette index+1* předchozího pruhu. Šířka barevných pruhů je určena vstupem *stripe_size* (4/8/16/32 bodů).

Zdrojové texty sem. práce musí splňovat následující formální požadavky:

- Syntaxe dle IEEE Std. 1076-1993 nebo IEEE Std. 1076-2008
- Modulární struktura
- Návrh musí být časově omezen (SDC) – definice hodinového signálu, setup a hold pro VGA DAC, atd.
- Dodržování zásad „štábní kultury“ tj. zejména:
  - smysluplné názvy identifikátorů
  - dobře čitelný strukturovaný zdrojový text (zarovnání bloků, atd.)
  - komentované zdrojové texty

Deklarace top level entity systému musí mít následující tvar (je nutné zachovat jména všech identifikátorů a jejich pořadí):

```
entity vga_gen is
  generic (
    -- Video Timing 800x600 @ 60 Hz, pixel clk 40.0 MHz
    PIX_FREQ : INTEGER := 40000000;
    H_PIXELS : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(800, 12);  -- horizontal display width in pixels
    H_FP     : UNSIGNED(9 DOWNTO 0)  := TO_UNSIGNED(40, 10);   -- horizontal front porch width in pixels
    H_PULSE  : UNSIGNED(9 DOWNTO 0)  := TO_UNSIGNED(128, 10);  -- horizontal sync pulse width in pixels
    H_BP     : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(88, 12);   -- horizontal back porch width in pixels
    H_POL    : STD_LOGIC := '1'; -- horizontal sync pulse polarity (1 = positive, 0 = negative)
    V_PIXELS : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(600, 12);  -- vertical display width in rows
    V_FP     : UNSIGNED(5 DOWNTO 0)  := TO_UNSIGNED(1, 6);     -- vertical front porch width in rows
    V_PULSE  : UNSIGNED(5 DOWNTO 0)  := TO_UNSIGNED(4, 6);     -- vertical sync pulse width in rows
    V_BP     : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(23, 12);   -- vertical back porch width in rows
    V_POL    : STD_LOGIC := '1'  -- vertical sync pulse polarity (1 = positive, 0 = negative)
  );
  port (
     clk : in std_logic;      -- vstup hodinoveho kmitoctu 50.0 MHz nebo 27.0 MHz z XTAL osc.
     reset_n : in std_logic;  -- vstup asynchronniho resetu (aktivni v nule)
     start_color : in std_logic_vector(3 downto 0);  -- barva prvního pruhu
     stripe_size : in std_logic_vector(1 downto 0);  -- sirka pruhu 00=4px, 01=8px, 10=16px, 11=32px
     vga_clk : out std_logic  -- hodinovy vystup pro D/A VGA prevodnik (zapis na vzestupnou hranu)
     vga_vsync, vga_hsync : out std_logic; -- vertikalni a horizontalni synchronizace VGA
     vga_r, vga_g, vga_b : out std_logic_vector(9 downto 0); -- R, G, B data pro D/A VGA
     vga_sync : out std_logic;  -- log. 0 = synchronizace na zelene barve vypnuta
     vga_blank : out std_logic  -- log. 1 = RGB vystupy povoleny
     -- pripadne dalsi vstupy/vystupy umistete az za tento komentar
  );
end;
```

Pokud budete potřebovat, aby top level entita měla další signály, přidejte jejich deklaraci za posledním signálem (tj. za signál *vga_blank*).

Hodnoceno bude následující:

- fyzická funkčnost navrženého systému, tj. splnění minimálních funkčních vlastností
- modularita, struktura kódu, SDC, tj. splnění formálních požadavků
- propracovanost a funkčnost testbenche (VHDL-93 nebo VHDL-2008)
- porozumění kódu při obhajobě semestrální práce
- bonusová implementace: podpora vyšších rozlišení (1024x768, …)
- na obhajobu práce si připravte krátkou prezentaci (3-4 minuty) Vašeho řešení obsahující:
    - blokové schéma vašeho řešení
    - co jednotlivé bloky umí, případně jaká mají vylepšení/nedostatky oproti zadání, atd.
    - parametry vašeho řešení (kolik obsahuje LUT, FF, paměti, f_{max}, atd.)

Příloha - úvod do generování VGA signálu

Obraz v rozlišení 800x600 bodů je složen ze 600 řádek a každá řádka má 800 bodů, které tvoří viditelný obraz. K vykreslení obrazu obsahuje analogový monitor vychylovací obvody. Tyto obvody umožňují směrovat elektrony emitované elektronovými děly po celé ploše stínítka (zleva–doprava, shora–dolů). Tyto vychylovací obvody potřebují dva synchronizační signály na spuštění a zastavení vychylování. Vykreslování obrazu probíhá vykreslením první řádky (zleva–doprava) a následným posunem na další nižší řádku (řádky se tedy zobrazují shora–dolů). Další informace generování VGA signálu a jeho časování pro různá rozlišení naleznete např. na následujících odkazech:

https://martin.hinner.info/vga/timing.html

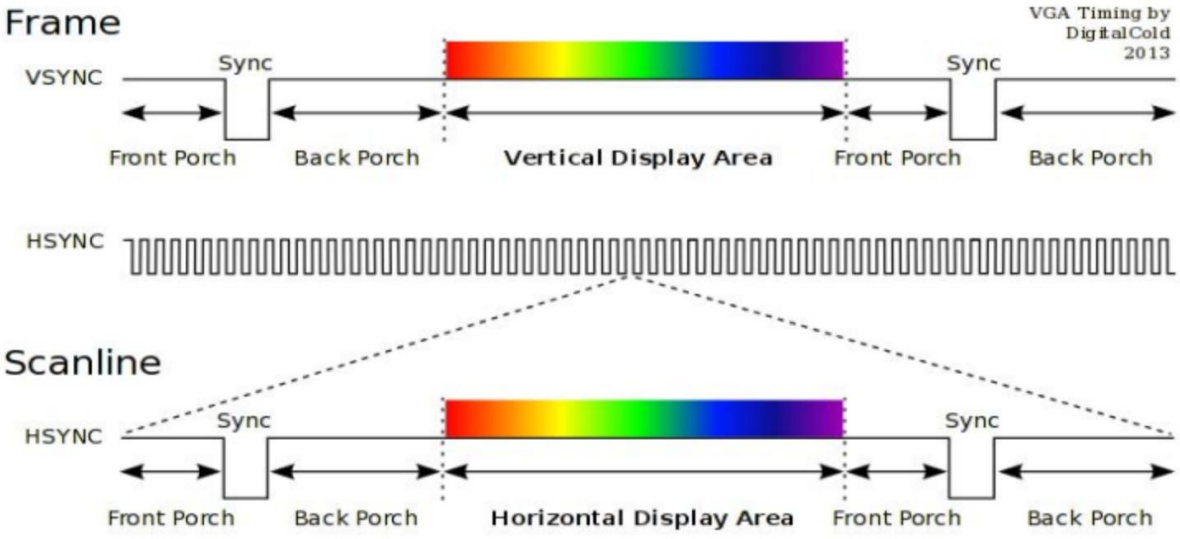
https://tomverbeure.github.io/video_timings_calculator

https://www.ultrasemi.com/en/tool/video_timing_calculator

Pulzy horizontální synchronizace označují začátek/konec řádky složené z jednotlivých bodů. Určují tak, že monitor zobrazuje body mezi levým a pravým okrajem viditelné plochy obrazovky. Podobně pulsy na signálu vertikální synchronizace označují začátek/konec obrazu složeného z řádek. Tím je zajištěno, že monitor zobrazí řádky mezi horním a dolním okrajem viditelné plochy obrazovky.

Návod na realizaci generátoru VGA signálu 800x600:
Při obrazových hodinách 40 MHz je doporučeno generovat obraz následovně. Realizovat dva čítače, jeden jako čítač bodů v řádce (horizontální) a druhý jako čítač řádek (vertikální). Rozsah čítače bodů v řádce je 1 až 1056 (800+40+128+88=1056 bodů v řádce), rozsah čítače řádek je pak 1 až 628 (600+1+4+23=628 řádek). Pro získání obrazového hodinového signálu 40MHz využijte nejlépe fázový závěs a generujte hodinový signál z XTAL oscilátoru 50 MHz nebo 27 MHz. Pro výstupní hodinový kmitočet pro VGA DAC použijte hodinový signál o stejné frekvenci jako pro obrazová data ale případně fázově posunutý, tak aby došlo k dodržení setup a hold časů pro VGA DAC (např. -60º).

Princip generování obrazu včetně synchronizačních pulzů:



Pro rozlišení 1024x768 můžete využít např. následující časování:

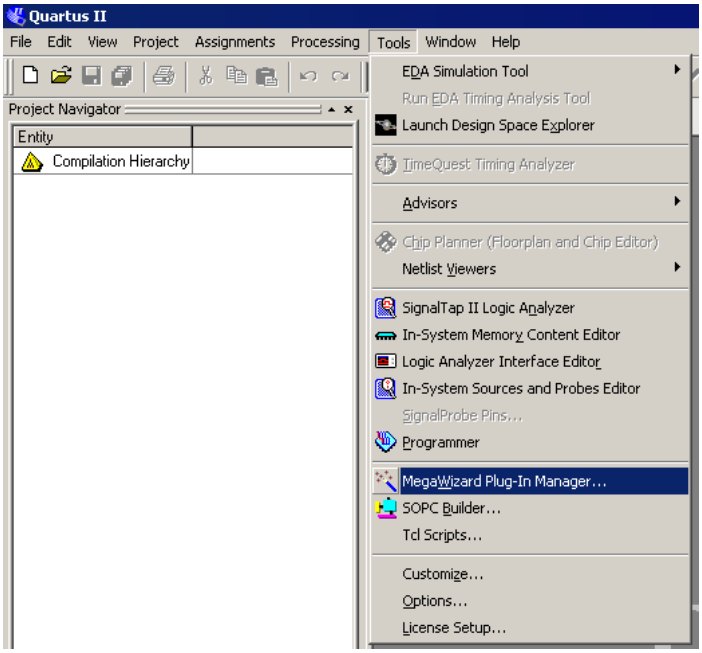
```
-- Video Timing 1024x768 @ 60Hz, pixel clock 65 MHz
PIX_FREQ : INTEGER := 65000000;
H_PIXELS : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(1024, 12);  -- horizontal display width in pixels
H_FP     : UNSIGNED(9 DOWNTO 0)  := TO_UNSIGNED(24, 10);    -- horizontal front porch width in pixels
H_PULSE  : UNSIGNED(9 DOWNTO 0)  := TO_UNSIGNED(136, 10);    -- horizontal sync pulse width in pixels
H_BP     : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(160, 12);   -- horizontal back porch width in pixels
H_POL    : STD_LOGIC := '1'; -- horizontal sync pulse polarity (1 = positive, 0 = negative)
V_PIXELS : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(768, 12); -- vertical display width in rows
V_FP     : UNSIGNED(5 DOWNTO 0)  := TO_UNSIGNED(3, 6);     -- vertical front porch width in rows
V_PULSE  : UNSIGNED(5 DOWNTO 0)  := TO_UNSIGNED(6, 6);     -- vertical sync pulse width in rows
V_BP     : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(29, 12);   -- vertical back porch width in rows
V_POL    : STD_LOGIC := '1'  -- vertical sync pulse polarity (1 = positive, 0 = negative)
```

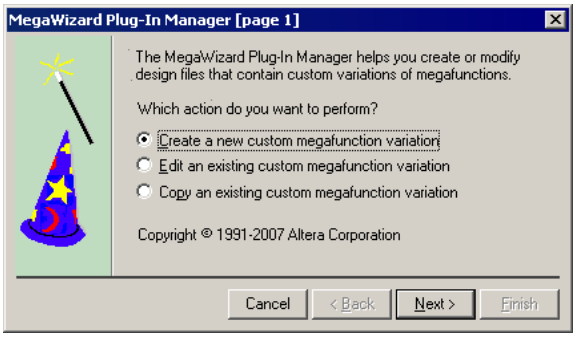
Barevná paleta pro 16 barev:

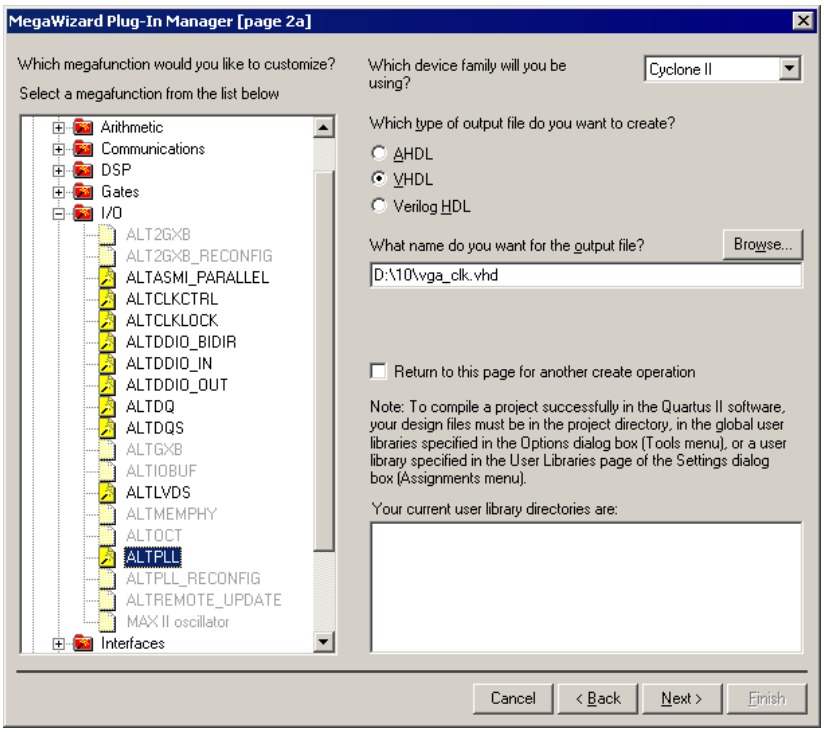
Color	Name	30-bit Binary RGB Vector			Palette Index
		R 10bit	G 10bit	B 10bit	Bin
	Black	"0000000000_0000000000_0000000000"			"0000"
	White	"1111111111_1111111111_1111111111"			"1111"
	Red	"1111111111_0000000000_0000000000"			"1100"
	Lime	"0000000000_1111111111_0000000000"			"1010"
	Blue	"0000000000_0000000000_1111111111"			"1001"
	Yellow	"1111111111_1111111111_0000000000"			"1110"
	Cyan / Aqua	"0000000000_1111111111_1111111111"			"1011"
	Magenta / Fuchsia	"1111111111_0000000000_1111111111"			"1101"
	Silver	"1100000000_1100000000_1100000000"			"1000"
	Gray	"1000000000_1000000000_1000000000"			"0111"
	Maroon	"1000000000_0000000000_0000000000"			"0100"
	Olive	"1000000000_1000000000_0000000000"			"0110"
	Green	"0000000000_1000000000_0000000000"			"0010"
	Purple	"1000000000_0000000000_1000000000"			"0101"
	Teal	"0000000000_1000000000_1000000000"			"0011"
	Navy	"0000000000_0000000000_1000000000"			"0001"

Příloha – použití fázového závěsu v FPGA Altera Cyclone II

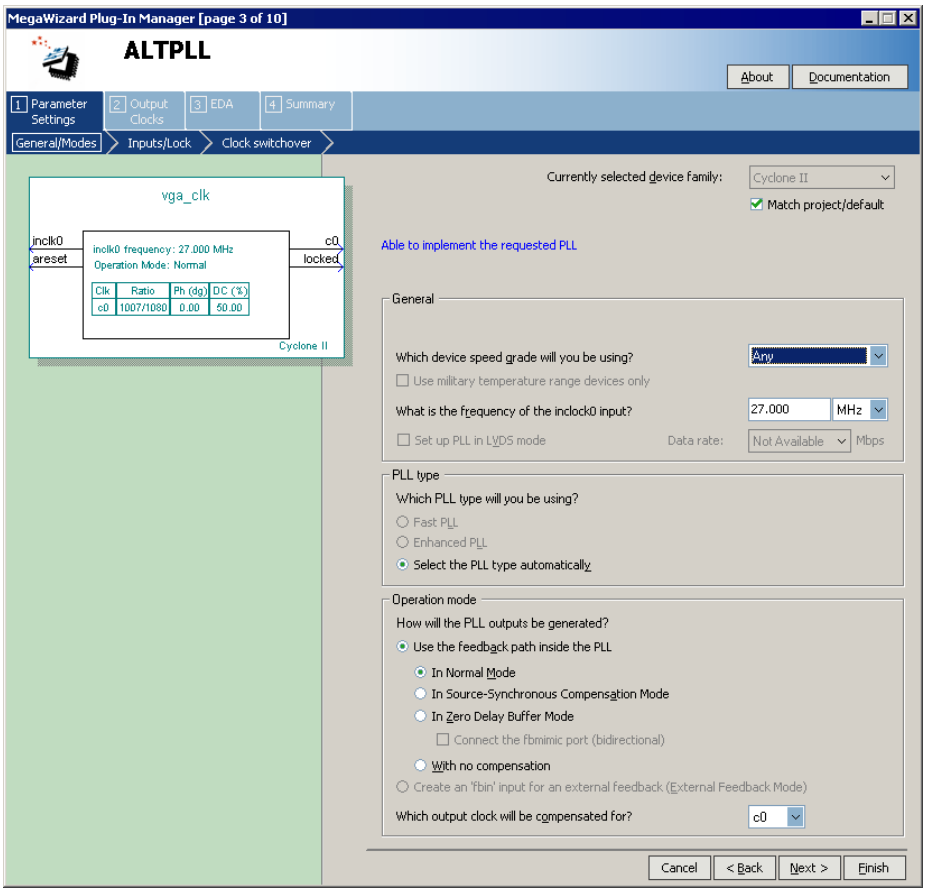
Zvolit menu: Tools | MegaWizard Plug-In Manager …

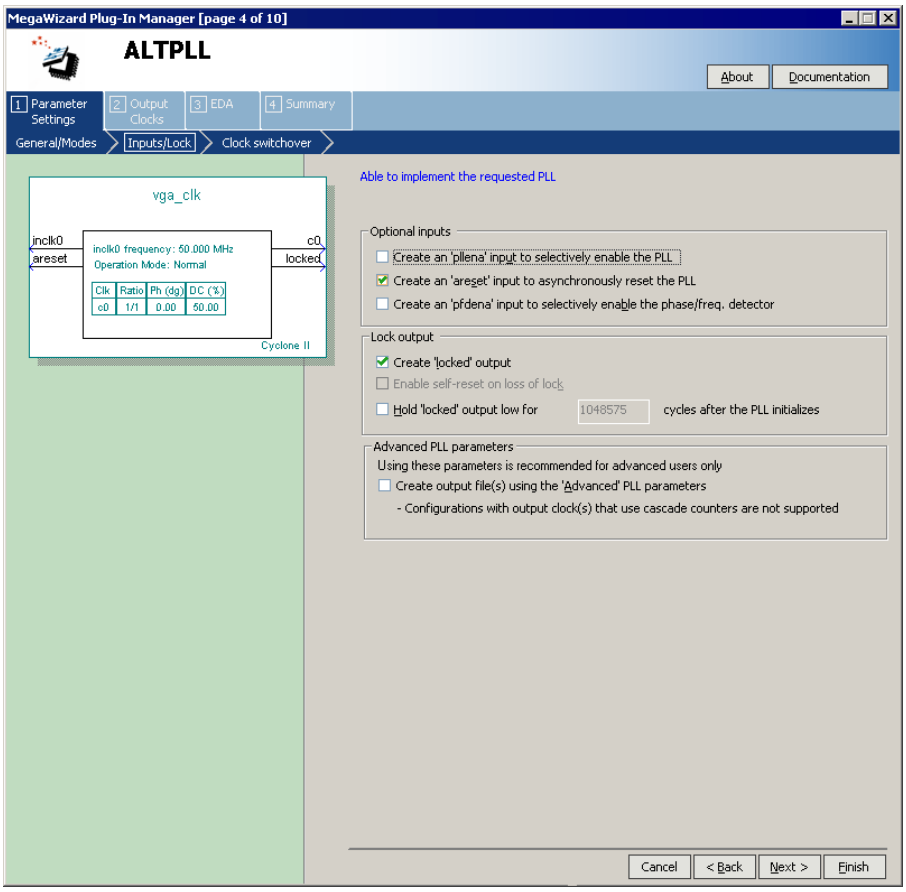


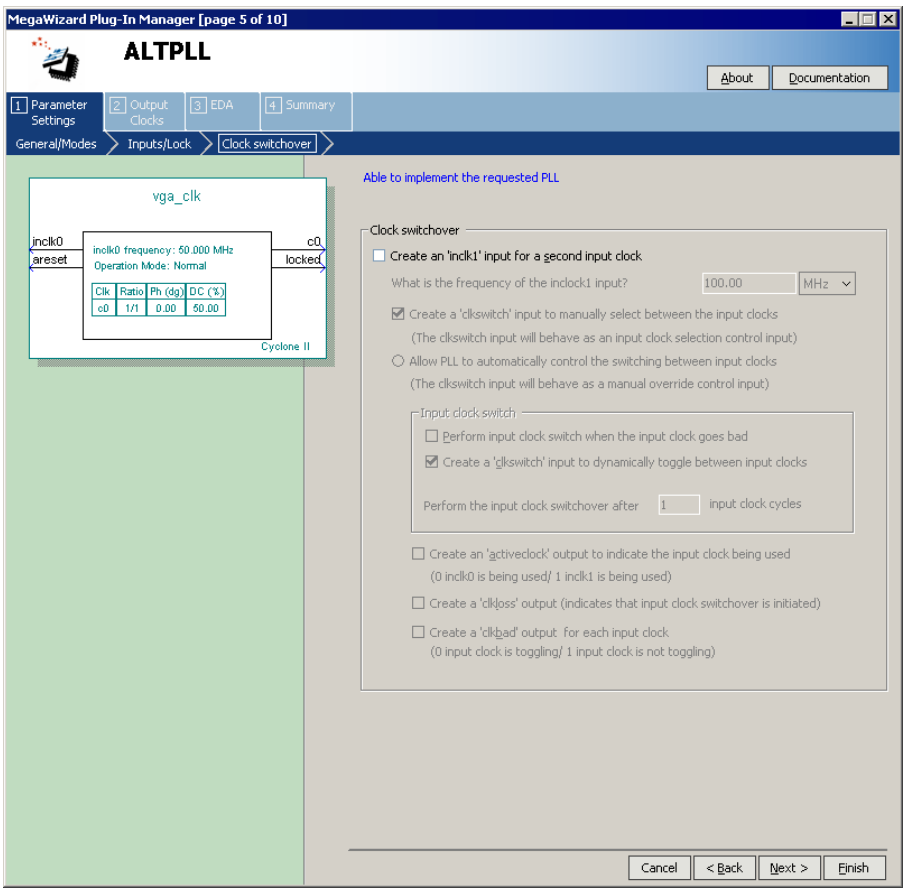




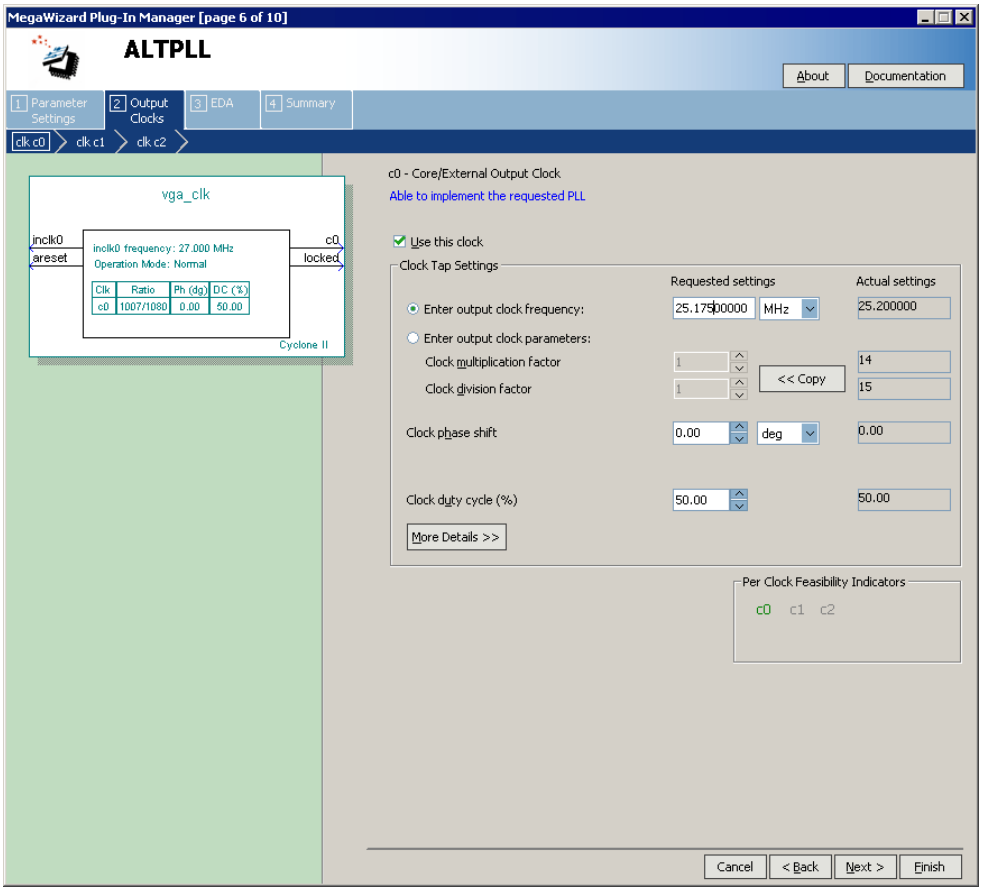
Nastavte vstupní frekvenci fázového závěsu (např. z XTAL oscilátortu 27MHz)

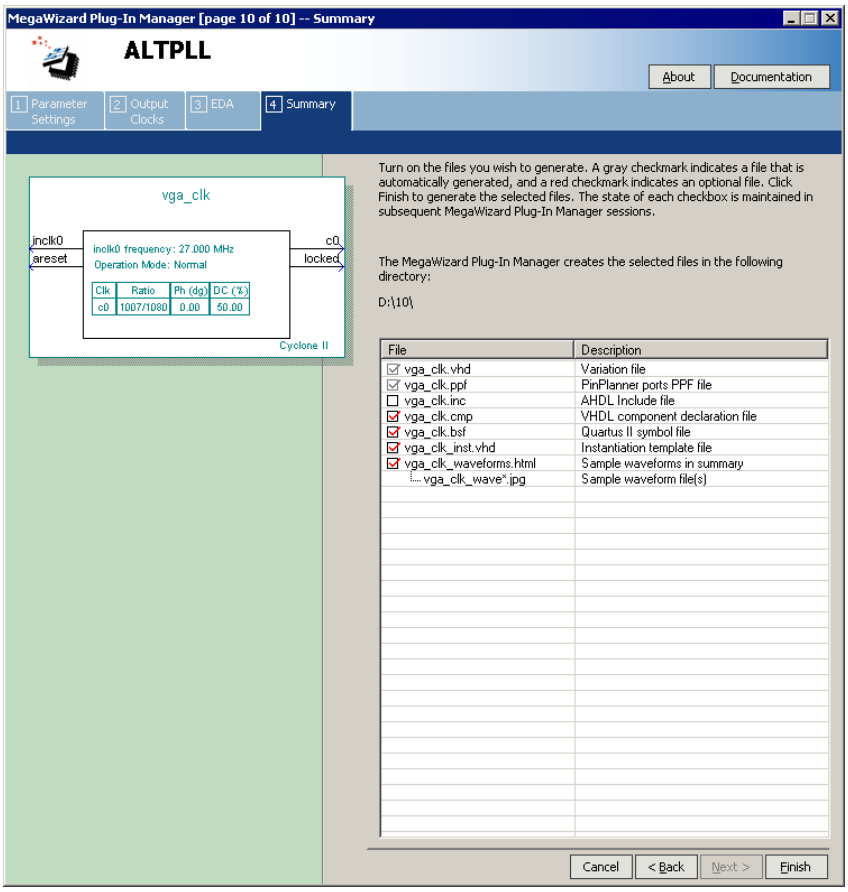



MegaWizard Plug-In Manager [page 4 of 10]
ALTPLL
About
Documentation
1 Parameter Settings
2 Output Clocks
3 EDA
4 Summary
General/Modes
Inputs/Lock
Clock switchover
vga_clk
inclk0
areset
c0
locked
inclk0 frequency: 50.000 MHz
Operation Mode: Normal
Clk
Ratio
Ph (dg)
DC (%)
c0
1/1
0.00
50.00
Cyclone II
Able to implement the requested PLL
Optional inputs
Create an 'pllena' input to selectively enable the PLL
Create an 'areset' input to asynchronously reset the PLL
Create an 'pfdena' input to selectively enable the phase/freq. detector
Lock output
Create 'locked' output
Enable self-reset on loss of lock
Hold 'locked' output low for
1048575
cycles after the PLL initializes
Advanced PLL parameters
Using these parameters is recommended for advanced users only
Create output file(s) using the 'Advanced' PLL parameters
- Configurations with output clock(s) that use cascade counters are not supported
Cancel
< Back
Next >
Finish


MegaWizard Plug-In Manager [page 5 of 10]
ALTPLL
About
Documentation
1 Parameter Settings
2 Output Clocks
3 EDA
4 Summary
General/Modes
Inputs/Lock
Clock switchover
vga_clk
inclk0
areset
c0
locked
inclk0 frequency: 50.000 MHz
Operation Mode: Normal
Clk
Ratio
Ph (dg)
DC (%)
c0
1/1
0.00
50.00
Cyclone II
Able to implement the requested PLL
Clock switchover
Create an 'inclk1' input for a second input clock
What is the frequency of the inclock1 input?
100.00
MHz
Create a 'clkswitch' input to manually select between the input clocks
(The clkswitch input will behave as an input clock selection control input)
Allow PLL to automatically control the switching between input clocks
(The clkswitch input will behave as a manual override control input)
Input clock switch
Perform input clock switch when the input clock goes bad
Create a 'clkswitch' input to dynamically toggle between input clocks
Perform the input clock switchover after
1
input clock cycles
Create an 'activeclock' output to indicate the input clock being used
(0 inclk0 is being used/ 1 inclk1 is being used)
Create a 'clkloss' output (indicates that input clock switchover is initiated)
Create a 'clkbad' output for each input clock
(0 input clock is toggling/ 1 input clock is not toggling)
Cancel
< Back
Next >
Finish

Nastavte požadovanou výstupní frekvenci fázového závěsu (pixel clock)

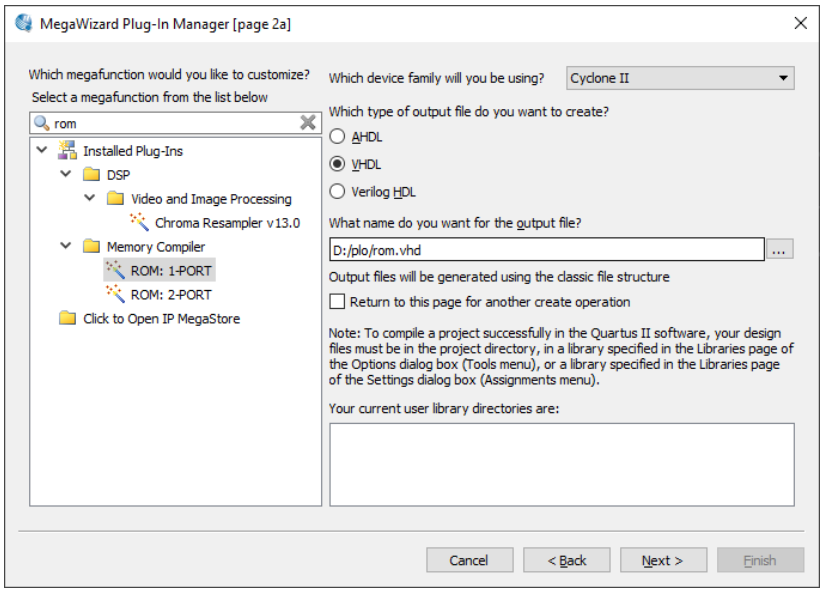


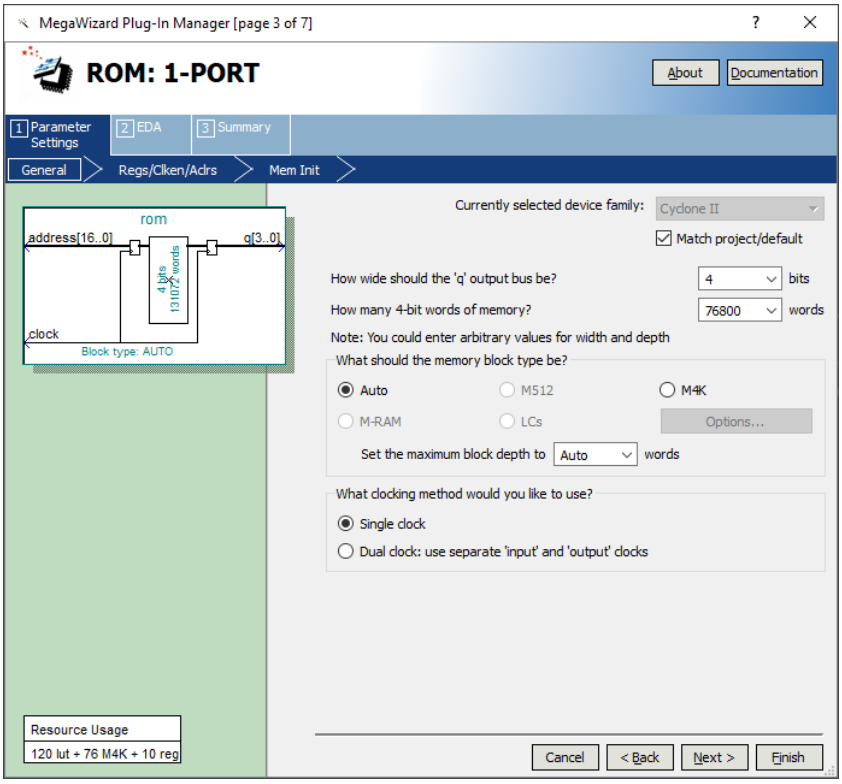


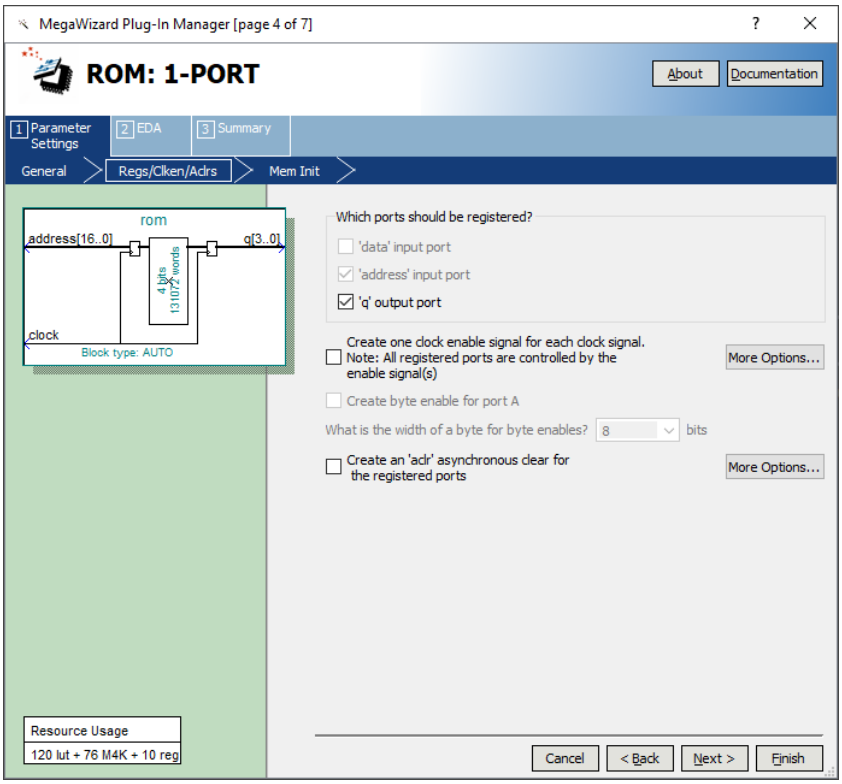
Příloha – použití vložených bloků pamětí ROM v FPGA

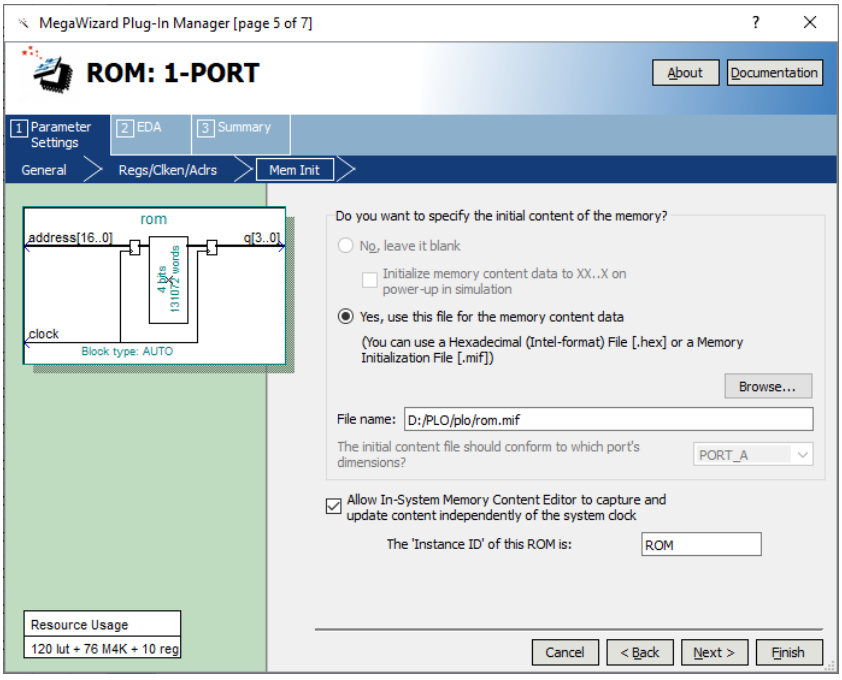
Zvolit menu: Tools | MegaWizard Plug-In Manager …



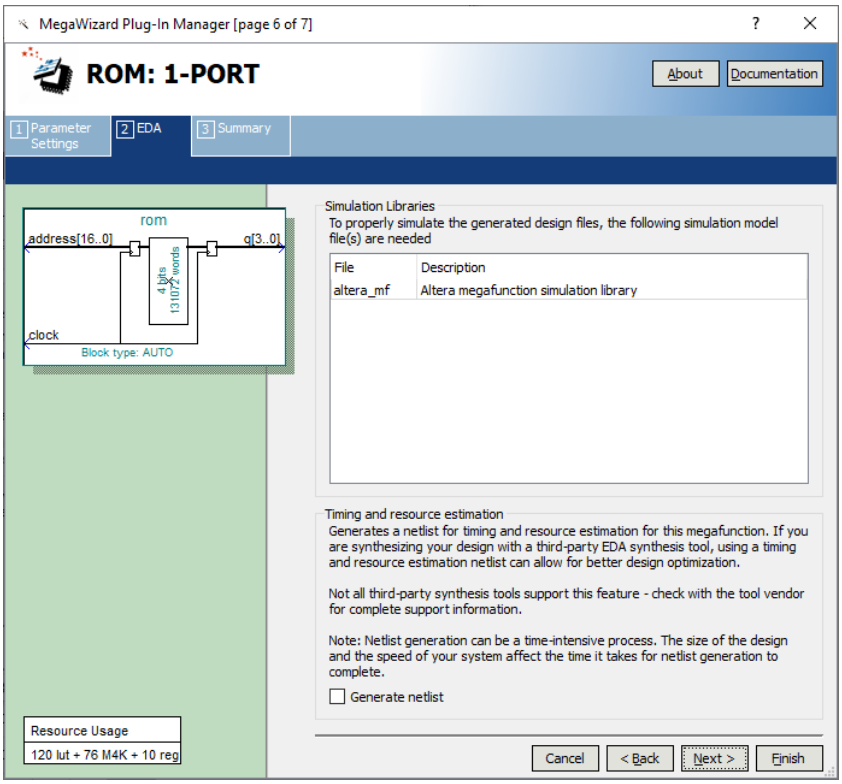

MegaWizard Plug-In Manager [page 2a]
Which megafunction would you like to customize?
Select a megafunction from the list below
rom
Installed Plug-Ins
DSP
Video and Image Processing
Chroma Resampler v13.0
Memory Compiler
ROM: 1-PORT
ROM: 2-PORT
Click to Open IP MegaStore
Which device family will you be using?
Cyclone II
Which type of output file do you want to create?
AHDL
VHDL
Verilog HDL
What name do you want for the output file?
D:/plo/rom.vhd
Output files will be generated using the classic file structure
Return to this page for another create operation
Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in a library specified in the Libraries page of the Options dialog box (Tools menu), or a library specified in the Libraries page of the Settings dialog box (Assignments menu).
Your current user library directories are:
Cancel
< Back
Next >
Finish

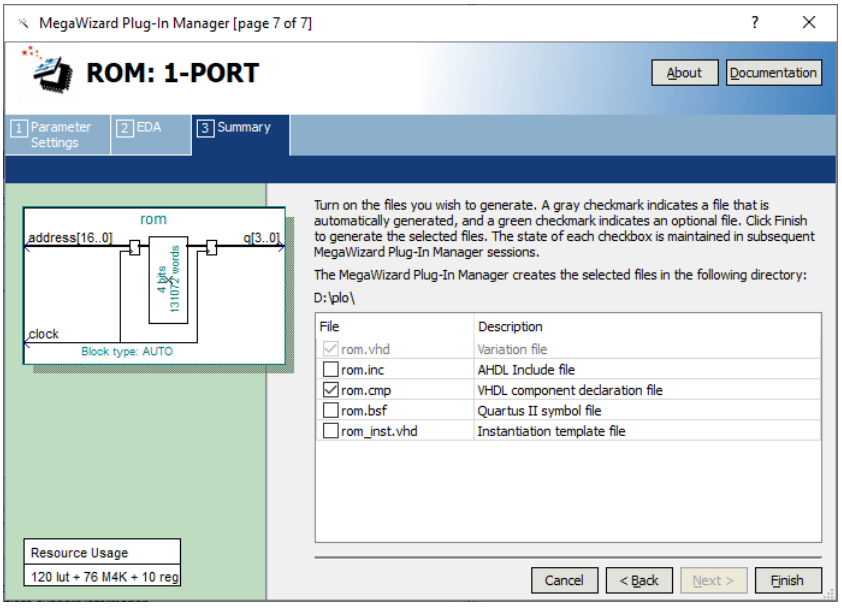

MegaWizard Plug-In Manager [page 3 of 7]
ROM: 1-PORT
About
Documentation
1 Parameter Settings
2 EDA
3 Summary
General
Regs/Clken/Adrs
Mem Init
rom
address[16..0]
q[3..0]
4 bits
131072 words
clock
Block type: AUTO
Currently selected device family:
Cyclone II
Match project/default
How wide should the 'q' output bus be?
4
bits
How many 4-bit words of memory?
76800
words
Note: You could enter arbitrary values for width and depth
What should the memory block type be?
Auto
M512
M4K
M-RAM
LCs
Options...
Set the maximum block depth to
Auto
words
What clocking method would you like to use?
Single clock
Dual clock: use separate 'input' and 'output' clocks
Resource Usage
120 lut + 76 M4K + 10 reg
Cancel
< Back
Next >
Finish





Poznámka: Inicializační data pro ROM paměť použijte ze souboru rom.zip (CourseWare)


MegaWizard Plug-In Manager [page 6 of 7]
ROM: 1-PORT
About
Documentation
1 Parameter Settings
2 EDA
3 Summary
rom
address[16..0]
q[3..0]
4 bits
131072 words
clock
Block type: AUTO
Simulation Libraries
To properly simulate the generated design files, the following simulation model file(s) are needed
File
Description
altera_mf
Altera megafunction simulation library
Timing and resource estimation
Generates a netlist for timing and resource estimation for this megafunction. If you are synthesizing your design with a third-party EDA synthesis tool, using a timing and resource estimation netlist can allow for better design optimization.
Not all third-party synthesis tools support this feature - check with the tool vendor for complete support information.
Note: Netlist generation can be a time-intensive process. The size of the design and the speed of your system affect the time it takes for netlist generation to complete.
Generate netlist
Resource Usage
120 lut + 76 M4K + 10 reg
Cancel
< Back
Next >
Finish


MegaWizard Plug-In Manager [page 7 of 7]
ROM: 1-PORT
About
Documentation
1 Parameter Settings
2 EDA
3 Summary
rom
address[16..0]
q[3..0]
4 bits
131072 words
clock
Block type: AUTO
Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a green checkmark indicates an optional file. Click Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.
The MegaWizard Plug-In Manager creates the selected files in the following directory:
D:\plo\
File
Description
rom.vhd
Variation file
rom.inc
AHDL Include file
rom.cmp
VHDL component declaration file
rom.bsf
Quartus II symbol file
rom_inst.vhd
Instantiation template file
Resource Usage
120 lut + 76 M4K + 10 reg
Cancel
< Back
Next >
Finish