

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 640x480 obrazových bodů (časování viz příloha) zobrazující svíslé 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). Barva okraje je dána vstupem *border_color* (dle *palette index*).

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 „štabní 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 640x480 @ 60 Hz, pixel clk 25.175 MHz
    PIX_FREQ : INTEGER := 25175000;
    H_PIXELS : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(640, 12); -- horizontal display width in pixels
    H_FP      : UNSIGNED(9 DOWNTO 0)  := TO_UNSIGNED(8, 10);   -- horizontal front porch width in pixels
    H_PULSE   : UNSIGNED(9 DOWNTO 0)  := TO_UNSIGNED(96, 10);  -- horizontal sync pulse width in pixels
    H_BP      : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(40, 12);  -- horizontal back porch width in pixels
    H_POL     : STD_LOGIC := '0'; -- horizontal sync pulse polarity (1 = positive, 0 = negative)
    V_PIXELS  : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(480, 12); -- vertical display width in rows
    V_FP      : UNSIGNED(5 DOWNTO 0)  := TO_UNSIGNED(2, 6);    -- vertical front porch width in rows
    V_PULSE   : UNSIGNED(5 DOWNTO 0)  := TO_UNSIGNED(2, 6);    -- vertical sync pulse width in rows
    V_BP      : UNSIGNED(11 DOWNTO 0) := TO_UNSIGNED(25, 12);  -- vertical back porch width in rows
    V_POL     : STD_LOGIC := '0' -- 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; -- vertikální a horizontální 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;
```

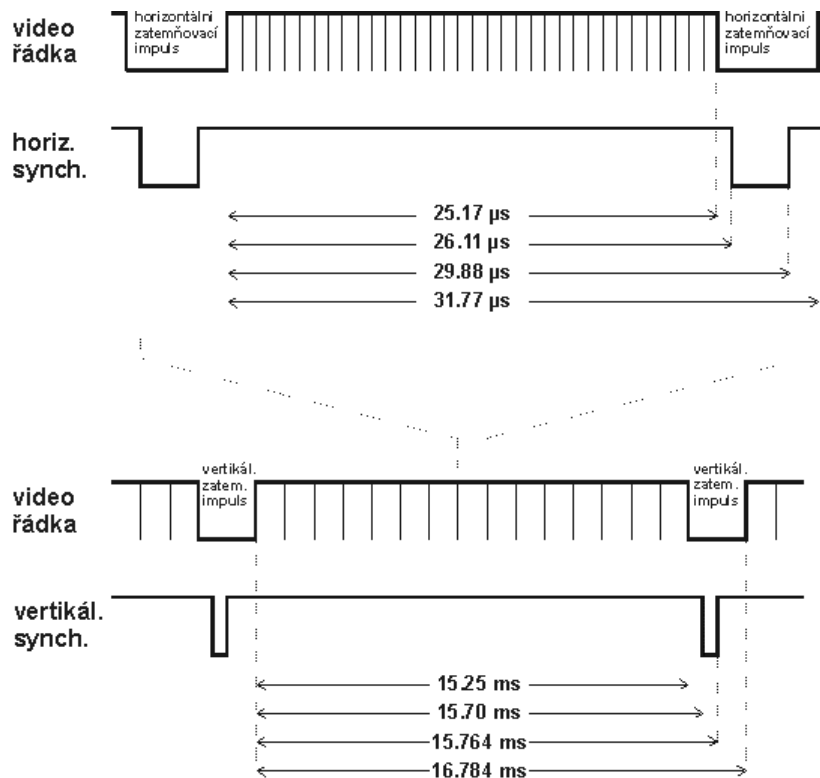
Pokud budete potřebovat, aby top level entita měla další signály, přidejte jejich deklaraci za posledním signálem (tj. za signál *vga_blank*).

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 dalších (vyšších rozlišení 800x600, 1024x768, ...)
- na obhajobu práce si připravte krátkou prezentaci (3-5 minut) 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 VGA je typicky složen např. ze 480 řádek a každá řádka mívá 640 bodů. K vykreslení obrazu obsahuje monitor vychylovací obvody. Tyto obvody umožňují směřovat 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). Časování synchronizačních signálů VGA ukazuje následující obrázek.



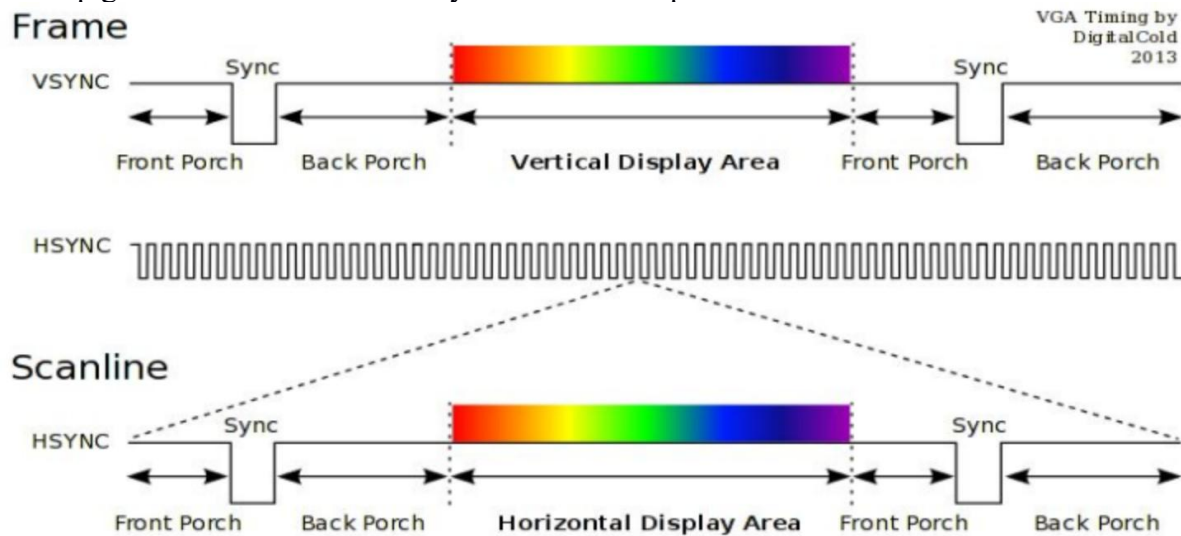
Negativní pulsy na signálu horizontální synchronizace označují začátek a konec řádky složené z bodů. Určují tak, že monitor zobrazuje body mezi levým a pravým okrajem viditelné plochy obrazovky. Body zobrazované řádky jsou vykreslovány během časového intervalu trvajícího 25,17 μs . Signál horizontální synchronizace přejde do úrovně „L“ minimálně za 0,94 μs po posledním zobrazeném bodu a zůstane v této úrovni po dobu 3,77 μs . Nová řádka bodů může začít minimálně za 1,89 μs po ukončení horizontálního synchronizačního pulsu. Takže jedna řádka bodů je vykreslována pouze po dobu 25,17 μs z celkového intervalu 31,77 μs . Zbývajících 6,6 μs probíhá tzv. řádkový zatemňovací interval. Po tuto dobu je na obrazovku zobrazována černá barva.

Podobně negativní pulsy na signálu vertikální synchronizace označují začátek a konec obrazu složeného z řádek. Tím je zajištěno, že monitor zobrazí řádky mezi horním a dolním okrajem viditelné plochy obrazovky. Tyto řádky jsou zobrazovány na monitoru během časového intervalu 15,25 ms. Signál vertikální synchronizace přejde do úrovně „L“ minimálně za 0,45 ms po poslední zobrazené řádce a zůstane v této úrovni po dobu 64 μs . Nová řádka bodů dalšího obrázku může začít minimálně za 1,02 ms po ukončení vertikálního synchronizačního pulsu. Jeden obrázek je tedy vykreslován pouze po dobu 15,25 ms z celkového intervalu 16,784 ms. Zbývajících 1,534 ms probíhá tzv. vertikální zatemňovací interval. Po tuto dobu je na obrazovku opět zobrazována černá barva.

Návod na realizaci generátoru VGA signálu 640x480:

Při obrazových hodinách 25 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 0 až 783 (640+8+96+40=784 bodů v řádce), rozsah čítače řádek pak 0 až 508 (480+2+2+25=509 řádek). Oba dva čítače tedy musí být minimálně desetibitové, (pro vyšší rozlišení budou muset mít ještě více bitů). Pro získání obrazového hodinového signálu využijte nejlépe fázový závěs a generujte hodinový signál 25,0 MHz nebo 25,2 MHz 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. -60°).

















Princip generování obrazu včetně synchronizačních pulzů:



```
-- Video Timing 800x600 @ 60Hz, pixel clock 40 MHz
PIX_FREQ : INTEGER := 40000000;
H_PIXELS : UNSIGNED(11 DOWNT0 0) := TO_UNSIGNED(800, 12); -- horizontal display width in pixels
H_FP     : UNSIGNED(9 DOWNT0 0)  := TO_UNSIGNED(40, 10);  -- horizontal front porch width in pixels
H_PULSE  : UNSIGNED(9 DOWNT0 0)  := TO_UNSIGNED(128, 10); -- horizontal sync pulse width in pixels
H_BP     : UNSIGNED(11 DOWNT0 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 DOWNT0 0) := TO_UNSIGNED(600, 12); -- vertical display width in rows
V_FP     : UNSIGNED(5 DOWNT0 0)  := TO_UNSIGNED(1, 6);   -- vertical front porch width in rows
V_PULSE  : UNSIGNED(5 DOWNT0 0)  := TO_UNSIGNED(4, 6);   -- vertical sync pulse width in rows
V_BP     : UNSIGNED(11 DOWNT0 0) := TO_UNSIGNED(23, 12); -- vertical back porch width in rows
V_POL    : STD_LOGIC := '1' -- vertical sync pulse polarity (1 = positive, 0 = negative)

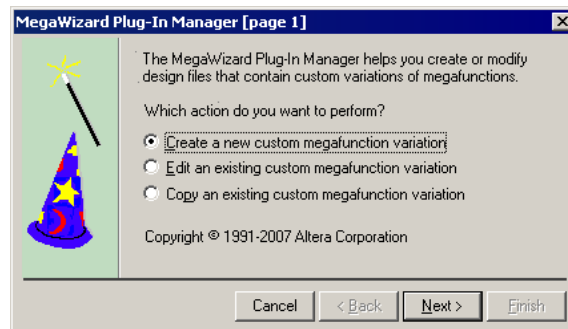
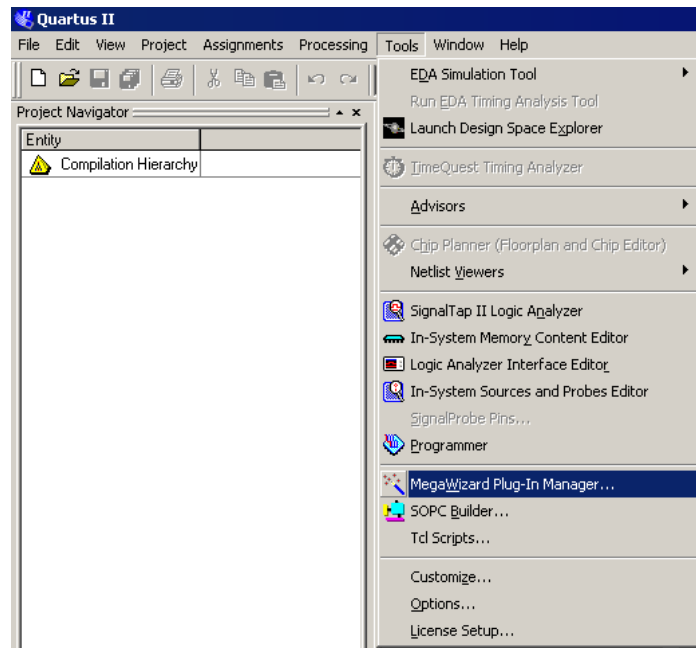
-- Video Timing 1024x768 @ 60Hz, pixel clock 65 MHz
PIX_FREQ : INTEGER := 65000000;
H_PIXELS : UNSIGNED(11 DOWNT0 0) := TO_UNSIGNED(1024, 12); -- horizontal display width in pixels
H_FP     : UNSIGNED(9 DOWNT0 0)  := TO_UNSIGNED(24, 10);  -- horizontal front porch width in pixels
H_PULSE  : UNSIGNED(9 DOWNT0 0)  := TO_UNSIGNED(136, 10); -- horizontal sync pulse width in pixels
H_BP     : UNSIGNED(11 DOWNT0 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 DOWNT0 0) := TO_UNSIGNED(768, 12); -- vertical display width in rows
V_FP     : UNSIGNED(5 DOWNT0 0)  := TO_UNSIGNED(3, 6);   -- vertical front porch width in rows
V_PULSE  : UNSIGNED(5 DOWNT0 0)  := TO_UNSIGNED(6, 6);   -- vertical sync pulse width in rows
V_BP     : UNSIGNED(11 DOWNT0 0) := TO_UNSIGNED(29, 12); -- vertical back porch width in rows
V_POL    : STD_LOGIC := '1' -- vertical sync pulse polarity (1 = positive, 0 = negative)
```

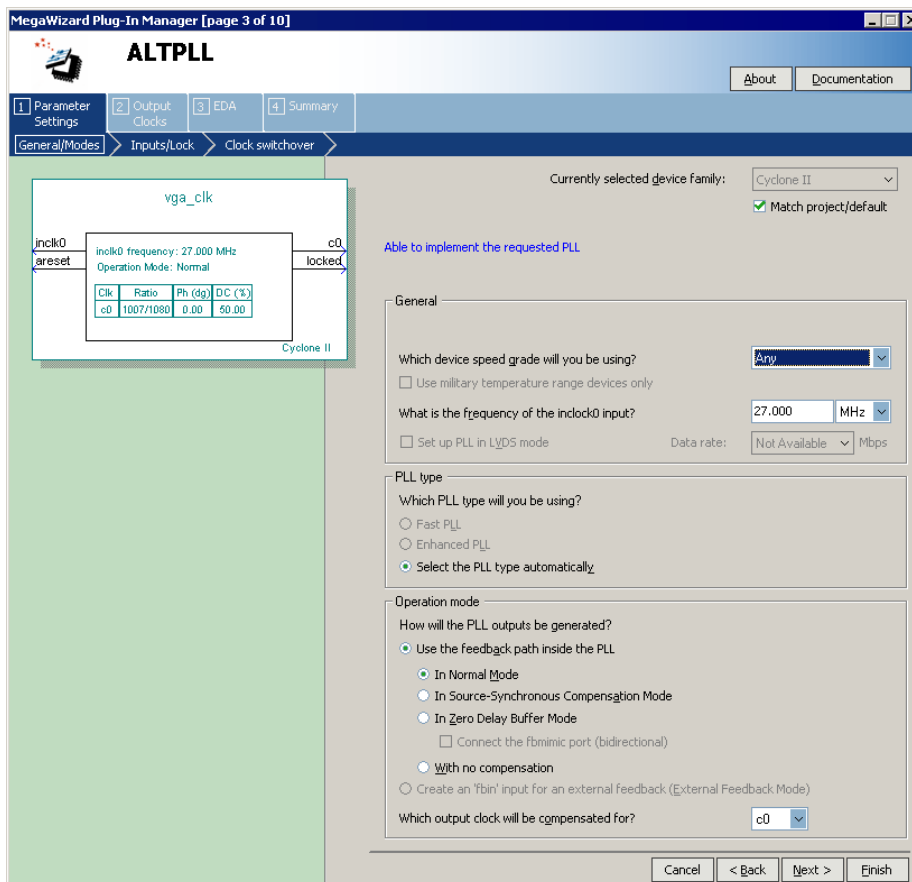
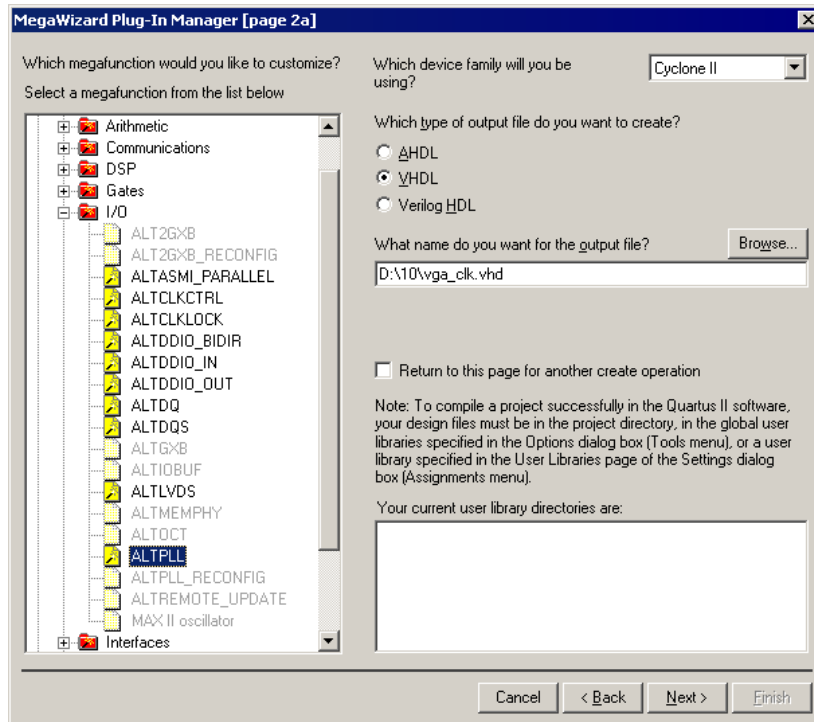
Barevná paleta pro 16 barev:

Color Name	30-bit Binary RGB Vector			Palette Index
	R 10bit	G 10bit	B 10bit	Bin
 Black	"000000000_000000000_000000000"			"0000"
 White	"111111111_111111111_111111111"			"1111"
 Red	"111111111_000000000_000000000"			"1100"
 Lime	"000000000_111111111_000000000"			"1010"
 Blue	"000000000_000000000_111111111"			"1001"
 Yellow	"111111111_111111111_000000000"			"1110"
 Cyan / Aqua	"000000000_111111111_111111111"			"1011"
 Magenta / Fuchsia	"111111111_000000000_111111111"			"1101"
 Silver	"110000000_110000000_110000000"			"1000"
 Gray	"100000000_100000000_100000000"			"0111"
 Maroon	"100000000_000000000_000000000"			"0100"
 Olive	"100000000_100000000_000000000"			"0110"
 Green	"000000000_100000000_000000000"			"0010"
 Purple	"100000000_000000000_100000000"			"0101"
 Teal	"000000000_100000000_100000000"			"0011"
 Navy	"000000000_000000000_100000000"			"0001"

Příloha – použití fázového závěsu v FPGA

Zvolit menu: Tools | MegaWizard Plug-In Manager ...





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

Cyclone II

Able to implement the requested PLL

Optional inputs

- Create an 'pllena' input to selectively enable the PLL
- Create an 'areget' 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 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

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 inclk1 input? 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 input clock cycles

- Create an 'activedlock' 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

MegaWizard Plug-In Manager [page 6 of 10]

ALTPLL

About Documentation

1 Parameter Settings 2 Output Clocks 3 EDA 4 Summary

clk c0 > clk c1 > clk c2 >

vga_clk

inclk0
areset

inclk0 frequency: 27.000 MHz
Operation Mode: Normal

Clk	Ratio	Ph (dg)	DC (%)
c0	1007/1080	0.00	50.00

c0
locked

Cyclone II

c0 - Core/External Output Clock
Able to implement the requested PLL

Use this clock

Clock Tap Settings

	Requested settings	Actual settings
<input checked="" type="radio"/> Enter output clock frequency:	25.1750000 MHz	25.200000
<input type="radio"/> Enter output clock parameters:		
Clock multiplication factor	14	14
Clock division factor	15	15
Clock phase shift	0.00 deg	0.00
Clock duty cycle (%)	50.00	50.00

More Details >>

Per Clock Feasibility Indicators

c0 c1 c2

Cancel < Back Next > Finish

MegaWizard Plug-In Manager [page 10 of 10] -- Summary

ALTPLL

About Documentation

1 Parameter Settings 2 Output Clocks 3 EDA 4 Summary

vga_clk

inclk0
areset

inclk0 frequency: 27.000 MHz
Operation Mode: Normal

Clk	Ratio	Ph (dg)	DC (%)
c0	1007/1080	0.00	50.00

c0
locked

Cyclone II

Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a red 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:

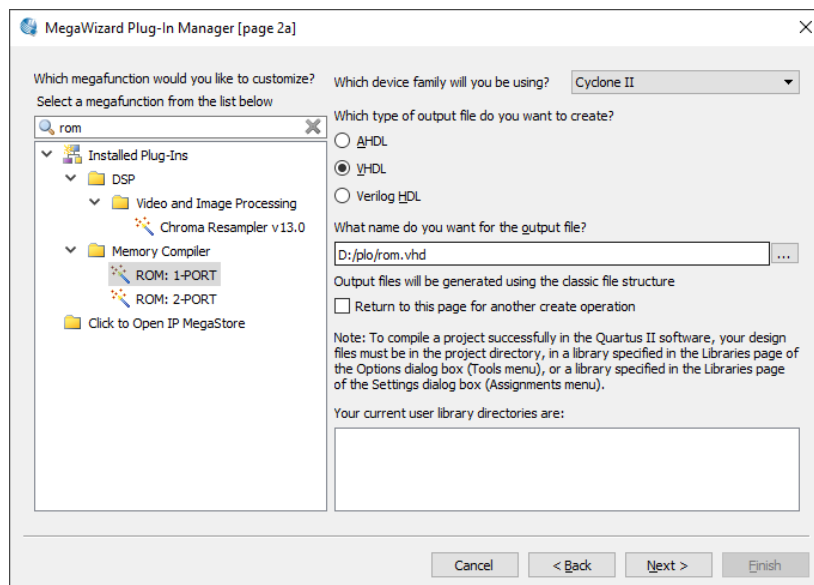
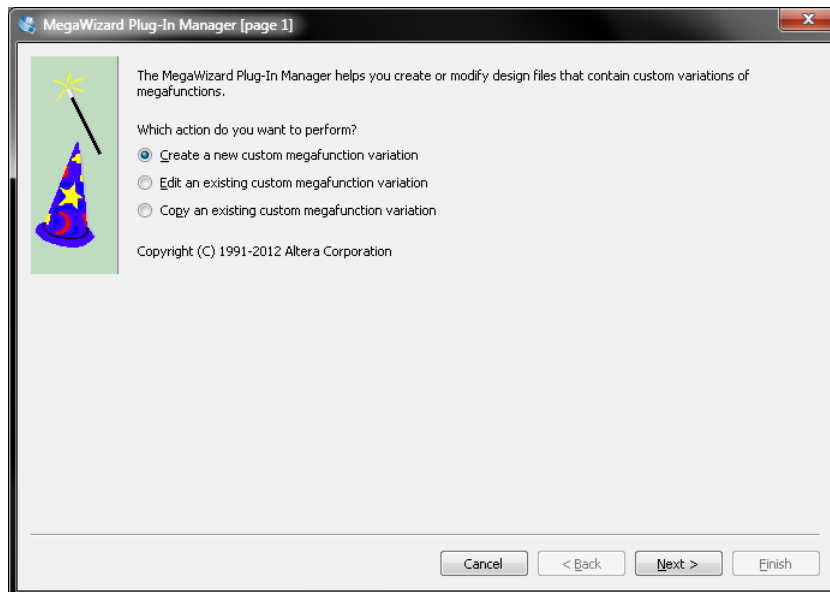
D:\10\

File	Description
<input checked="" type="checkbox"/> vga_clk.vhd	Variation file
<input checked="" type="checkbox"/> vga_clk.ppf	PinPlanner ports PPF file
<input type="checkbox"/> vga_clk.inc	AHDL Include file
<input checked="" type="checkbox"/> vga_clk.cmp	VHDL component declaration file
<input checked="" type="checkbox"/> vga_clk.bsf	Quartus II symbol file
<input checked="" type="checkbox"/> vga_clk_inst.vhd	Instantiation template file
<input checked="" type="checkbox"/> vga_clk_waveforms.html	Sample waveforms in summary
... vga_clk_wave*.jpg	Sample waveform file(s)

Cancel < Back Next > Finish

Příloha – použití vložených bloků paměti ROM v FPGA

Zvolit menu: Tools | MegaWizard Plug-In Manager ...



MegaWizard Plug-In Manager [page 3 of 7]

ROM: 1-PORT

About Documentation

1 Parameter Settings 2 EDA 3 Summary

General > Regs/Cken/Acfs > Mem Init >

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

MegaWizard Plug-In Manager [page 4 of 7]

ROM: 1-PORT

About Documentation

1 Parameter Settings 2 EDA 3 Summary

General > Regs/Cken/Acfs > Mem Init >

Which ports should be registered?

'data' input port
 'address' input port
 'q' output port

Create one clock enable signal for each clock signal.
 Note: All registered ports are controlled by the enable signal(s) More Options...

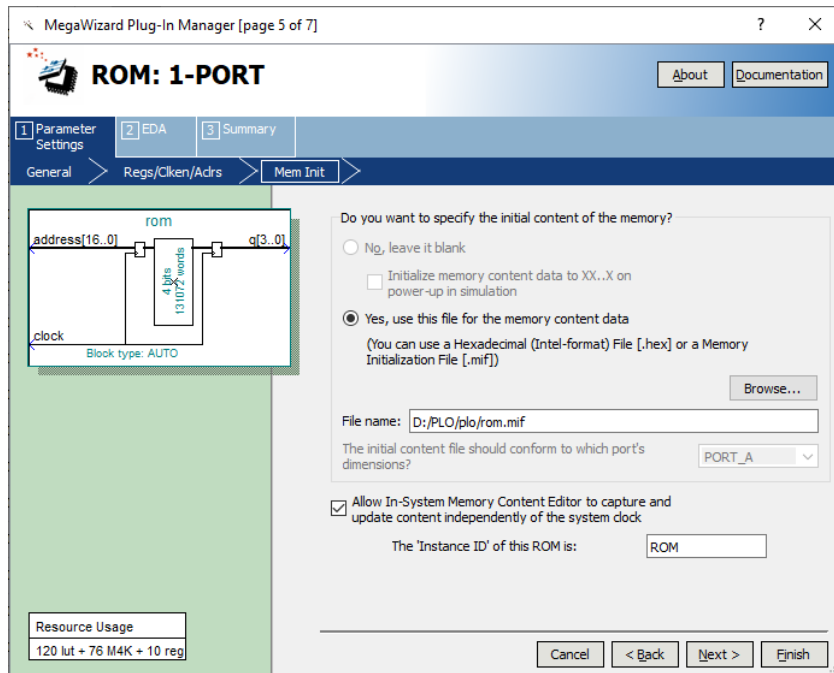
Create byte enable for port A

What is the width of a byte for byte enables? 8 bits

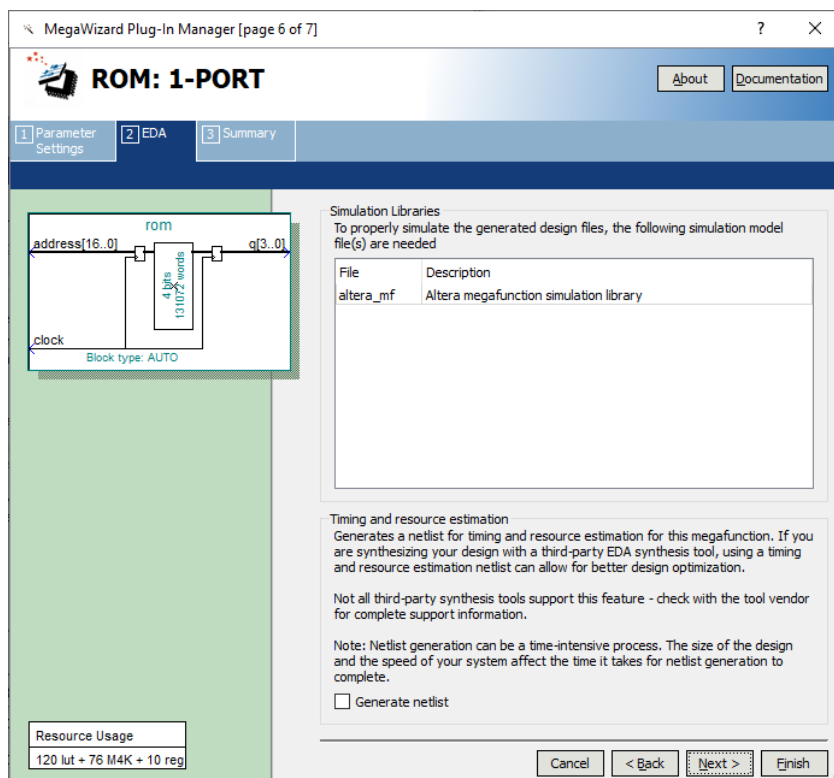
Create an 'aclr' asynchronous clear for the registered ports More Options...

Resource Usage
120 lut + 76 M4K + 10 reg

Cancel < Back Next > Finish



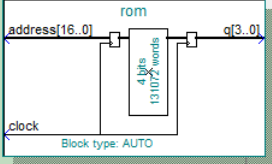
Poznámka: Inicializační data pro ROM paměť použijte ze souboru [rom.zip](#) (CourseWare)



MegaWizard Plug-In Manager [page 7 of 7] ? X

ROM: 1-PORT About Documentation

1 Parameter Settings | 2 EDA | 3 Summary



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
<input checked="" type="checkbox"/> rom.vhd	Variation file
<input type="checkbox"/> rom.inc	AHDL Include file
<input checked="" type="checkbox"/> rom.cmp	VHDL component declaration file
<input type="checkbox"/> rom.bsf	Quartus II symbol file
<input type="checkbox"/> rom_inst.vhd	Instantiation template file

Resource Usage
120 lut + 76 M4K + 10 reg

Cancel < Back Next > Finish