Rostock University
Faculty of Computer Science and Electronic Engineering
Electronic Circuits and Appliances Division (Institute GS)

VHDL-Lab 2: Controlling a Monitor in XGA-Mode

(Version for Spartan 6 on Scarab miniSpartan6+ board)

Task:

One part of a video control circuit is the creation of signals for the synchronization of the monitor and the DAC (digital to analog converter). Dependent on the resolution and the picture rate per second it has to create a distinct pattern for the signals:

HSYNC - horizontal synchronization signal (row)
VSYNC - vertical synchronization signal (picture) und
BLANK - blanking of the beam (horizontal and vertical), signal to the DAC

This part of a Video controller will be designed and tested in this lab.

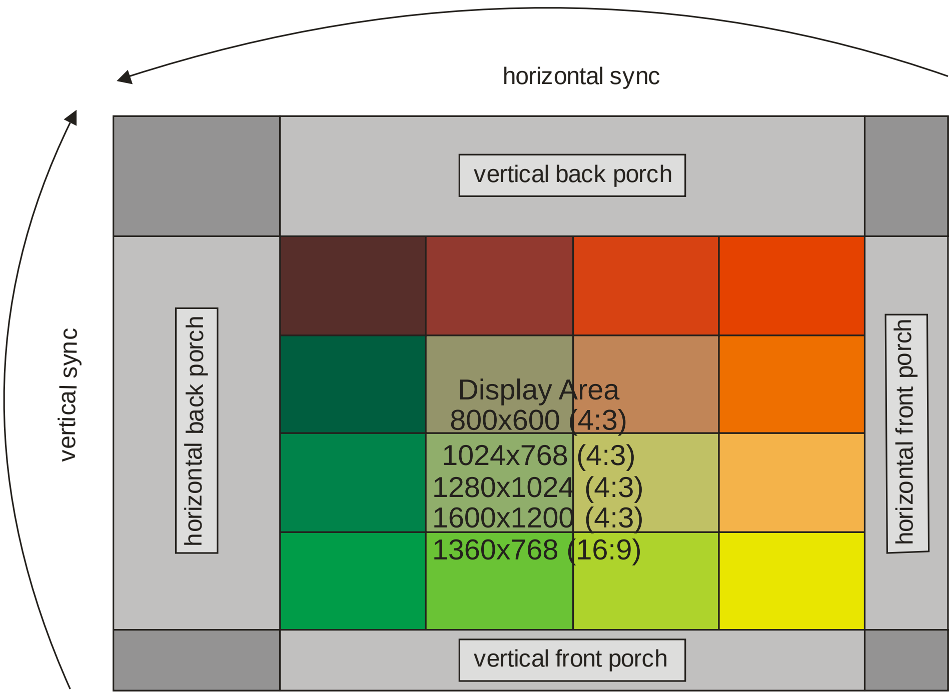
All signals are derived from the VIDEOCLOCK f_p:

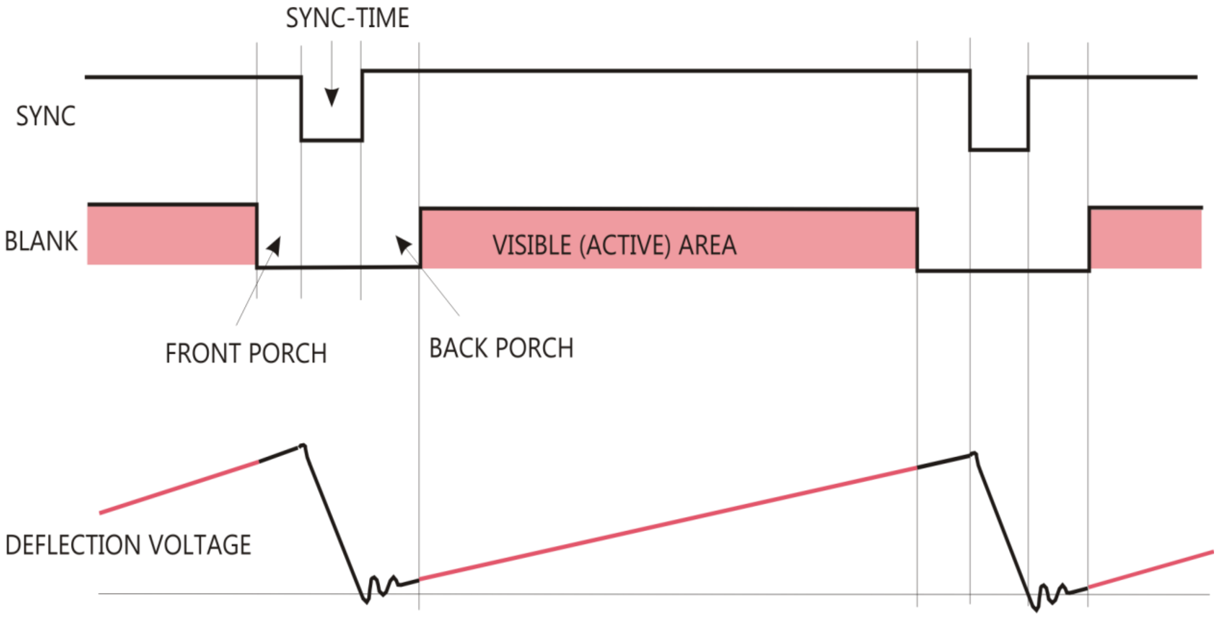
f_p = picture rate * number of rows * number of pixels per row

The numbers of rows and pixels per row **include** the time (expressed in numbers of rows/pixels) for blanking outside the border of the visible picture and for the horizontal and vertical retrace of the beam (SYNC-Pulses).

Resolution Refresh	Clock M/D	H-Active	H-Front-Porch	H-SYNC	H-Back-Porch	V-Active	V-Front-Porch	V-SYNC	V-Back-Porch
640x480 @60Hz	25.175 2/8	640	16	96 (low)	48	480	10	2 (low)	33
640x480 @75Hz	31.5 5/16	640	16	64 (low)	120	480	1	3 (low)	16
800x600 @72Hz	50 2/4	800	56	120 (high)	64	600	37	6 (high)	23
1024x768 @60Hz	65 13/20	1024	24	136 (low)	80	768	3	6 (low)	29
1024x768 @75Hz	78,75 11/14	1024	16	96 (high)	176	768	1	3 (high)	28
1280x1024 @60Hz	**108 27/25**	**1280**	**48**	**112 (low)**	**248**	**1024**	**1**	**3 (low)**	**38**

Table 1: Timing for video modes





We use a Spartan 6 on Scarab miniSpartan6+ board 50 MHz system clock. The Spartan-6-devices have a special clock management cell, based on a PLL, the Digital Clock Manager (DCM).

There is a frequency synthesis mode (DFS) available for Digital Clock Modules (DCM). Using this mode we can derive a clock of $f = \frac{f_{in} \cdot MULT}{DIV}$ with MULT an integer in the range of 2…256 and DIV an integer in the range of 1…256.
In VHDL based design simulation the MULT and DIV values are set as generics into the DCM model. For synthesis we use entries in the XILINX constraint file (ucf-file).
One can use the XILINX core generator wizard to generate the clock source (Project -> New Source -> Coregen -> FPGA Features and Design -> Clocking -> Clocking Wizard).

For 50 MHz input and 108 MHz output MULT/DIV = 28/13 (107.69 MHz] or 13/6 (108.33 MHZ) are practical values. The wizard will calculate them from inputs for input and output frequency.

The card uses a 240 MHz true colour digital to analogous converter (ADV7125KST240). Digital colour inputs are binary encoded with 8 bits per colour.

Logic zero at BLANK input of the DAC drives the analog outputs to blanking level (black). The DAC can encode the synchronization information on the GREEN output. We will not use this option and set the CSYNC output to '1' permanently.
VGA-part of demonstration board with xc6xlx9-3ftg256 FPGA

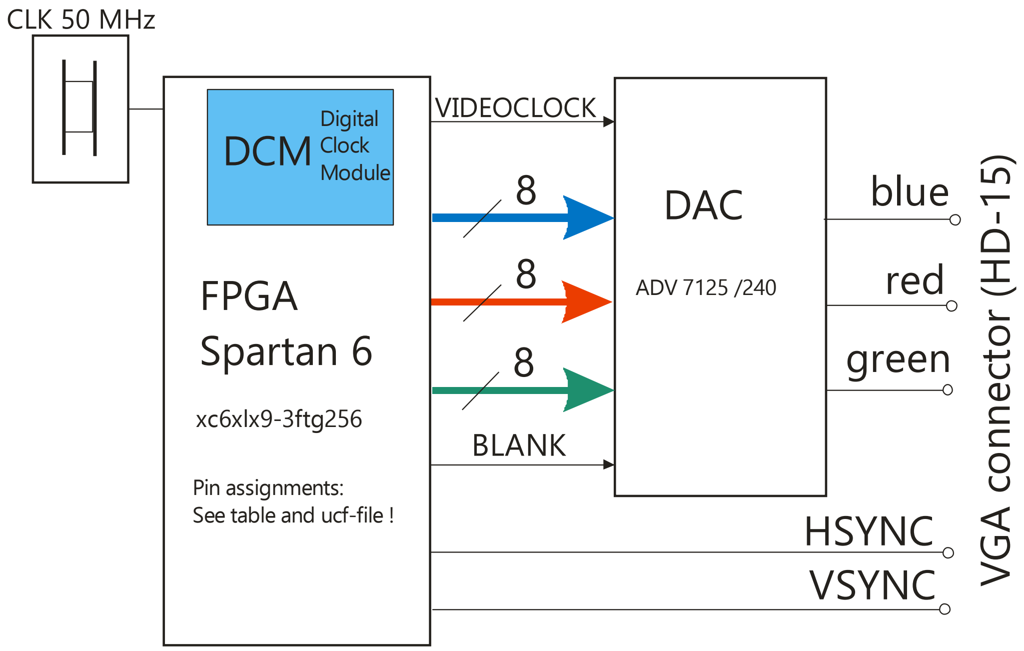


Table 1: Pin assignment for demonstration board with xc6xlx9-3ftg256:

VGA-Signal	ADV725	Adapter-Verbinder	Scarab-Port	Spartan 6
CLOCK	24	D5	PORTD2	E6
HSYNC	-	D3	PORTD0	D6
VSYNC	-	D4	PORTD1	C6
BLANK	11	D6	PORTD3	C5
RED0	41	A1	PORTA0	E7
RED1	42	A2	PORTA1	C8
RED2	43	A3	PORTA2	D8
RED3	44	A4	PORTA3	E8
RED4	45	A5	PORTA4	D9
RED5	46	A6	PORTA5	A10
RED6	47	A9	PORTA6	B10
RED7	48	A10	PORTA7	C10
GREEN0	3	A11	PORTA8	E10
GREEN1	4	A12	PORTA9	F9
GREEN2	5	A13	PORTA10	F10
GREEN3	6	A14	PORTA11	D11
GREEN4	7	B1	PORTB0	E11

GREEN5	8	B2	PORTB1	D14
GREEN6	9	B3	PORTB2	D12
GREEN7	10	B4	PORTB3	E12
BLUE0	16	B5	PORTB4	E13
BLUE 1	17	B6	PORTB5	F13
BLUE 2	18	B9	PORTB6	F12
BLUE 3	19	B10	PORTB7	F14
BLUE 4	20	B11	PORTB8	G12
BLUE 5	21	B12	PORTB9	G14
BLUE 6	22	B13	PORTB10	H14
BLUE 7	23	B14	PORTB11	J14

Tasks:

1. Create a design for the 1280x1024@60Hz resolution with CLK_50MHz as input clock and the signals VIDEOCLK, HSYNC, VSYNC and BLANK as outputs. In addition it shall produce the color information for a picture on 3*8 bit. The picture should be a predefined pattern. Divide the design into the parts:
   - Generation and control of clocks
   - Horizontal counter
   - Vertical counter
   - Creation of RGB signals

   .
2. Make a functional (VHDL-) simulation in Modelsim or XILINX ISim!
3. Synthesize the design using XSE from XILINX!
4. Implement the design in a xc6xlx9-3ftg256 FPGA from XILINX. Can your design work with the given pixel clock?
5. Carry out the post-layout simulation in Modelsim or Xilinx ISim!
6. Open the design inside fpga_editor and try to recognize your description in the placed and routed hardware.
7. Download the design to a testboard and show that it works.
8. Measure the signals by use of an oscilloscope and/or logic analyzer.

Some Useful Hints:

1. Use integer types with limited ranges for your counters.
2. Use the HSYNC-Output from the horizontal counter as a clock for the vertical counter. Be aware that an additional clock driver will be necessary.
3. You can construct color patterns using the row or column position.
4. Use the prepared template vga.vhd for your design and the testbench vga_tb.vhd for simulation.
5. Try to implement the design flow in scripts instead of using GUI. It helps to understand the details of the design flow.

Additional Tasks:

A good idea to create a background would be showing boxes of 32x32 pixels with (pseudo-) randomly generated colors.
Another idea is the display of a (moving) pattern like a circle or some other figure. You can use the core generator from XILINX to implement memory (ROM or RAM-cells). Use such cells for the storage of patterns. Show these patterns on the screen.

If you put the pattern shown in the picture into a ROM of 256x1 bits you can show a circle on the display. Therefore it is necessary to read the cells in the right order.
For this purpose a counter is connected with the address inputs of the memory and depending of the content of the memory ('0', '1') a preselected color information is displayed. For more than one color or pattern a memory of 256 x n bits must be implemented.

Modules generated by core generator are part of the FPGA-library like clock buffers, oscillators and must be inserted as components into the VHDL description.

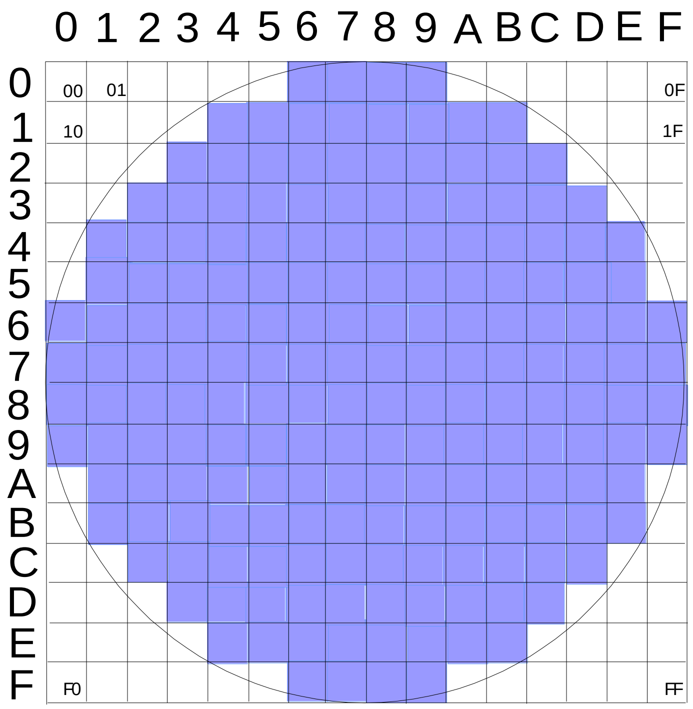
Core generator generates a VHDL-file for simulation and a piece of hardware that the XILINX development software can insert into the design. The synthesis software handles this module as a black box.
For the component instantiation you need the names and port declarations of the components.

A usable clock buffer is:

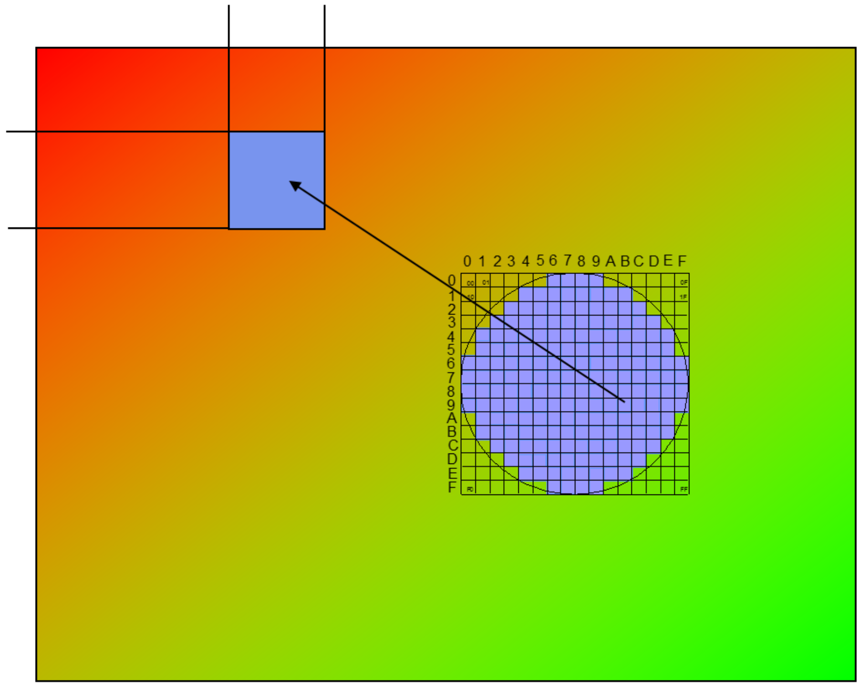
```
component BUFGS
      Port (       I : In     std_logic;
                   O : Out   std_logic );
   end component;
```

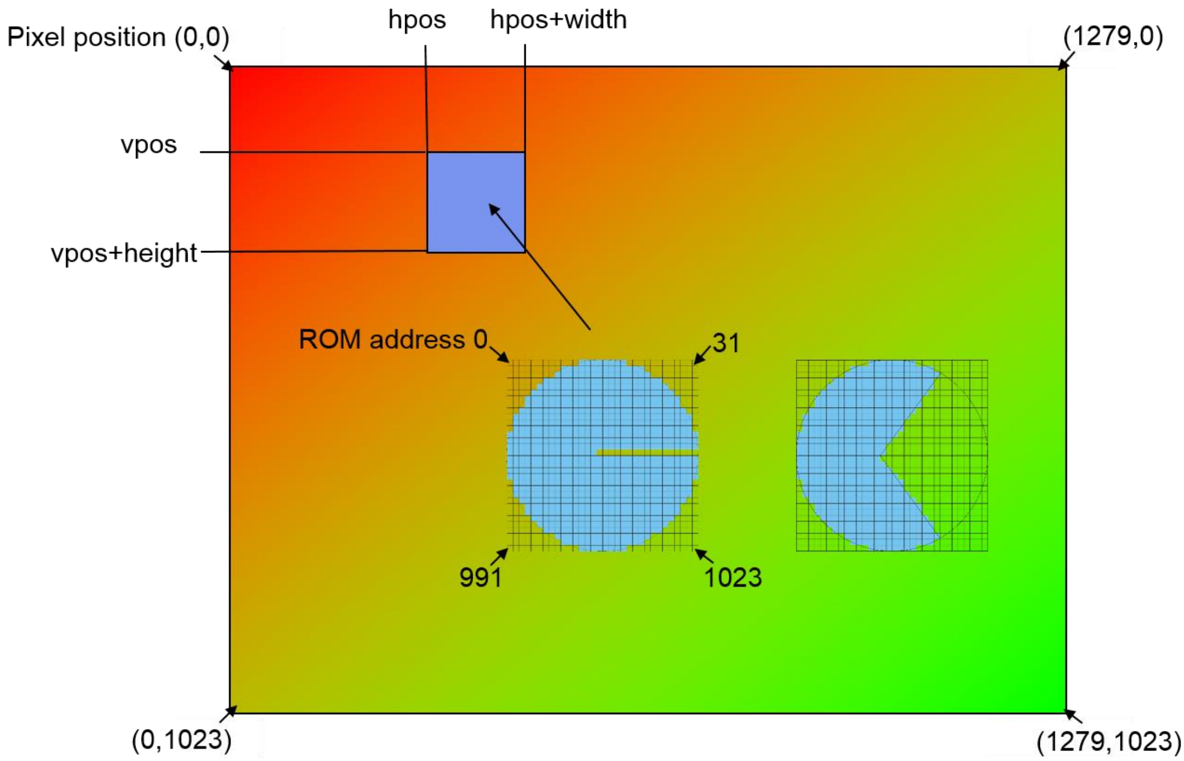
Here is an example of a ROM of 256x2 Bit. In your design you might have selected another name and other ranges for address and data. You can find the exact port declaration in the generated VHDL-File from the core generator.
The contents of the memory cells is given in a file with the extension .coe.

```
component rom1 is
  PORT(
    A:      IN std_logic_vector(7 DOWNTO 0);
    DO: OUT std_logic_vector(1 DOWNTO 0));
end component;
```



This figure can help to create a pattern file for a (moving) object.


0 1 2 3 4 5 6 7 8 9 A B C D E F
0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F


Pixel position (0,0)
hpos
hpos+width
(1279,0)
vpos
vpos+height
ROM address 0
31
991
1023
(0,1023)
(1279,1023)

50,00	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
2	100,0	50,0	33,3	25,0	20,0	16,7	14,3	12,5	11,1	10,0	9,1	8,3	7,7	7,1	6,7	6,3	5,9	5,6	5,3	5,0	4,8
3	150,0	75,0	50,0	37,5	30,0	25,0	21,4	18,8	16,7	15,0	13,6	12,5	11,5	10,7	10,0	9,4	8,8	8,3	7,9	7,5	7,1
4	200,0	100,0	66,7	50,0	40,0	33,3	28,6	25,0	22,2	20,0	18,2	16,7	15,4	14,3	13,3	12,5	11,8	11,1	10,5	10,0	9,5
5	250,0	125,0	83,3	62,5	50,0	41,7	35,7	31,3	27,8	25,0	22,7	20,8	19,2	17,9	16,7	15,6	14,7	13,9	13,2	12,5	11,9
6	>fmax	150,0	100,0	75,0	60,0	50,0	42,9	37,5	33,3	30,0	27,3	25,0	23,1	21,4	20,0	18,8	17,6	16,7	15,8	15,0	14,3
7	>fmax	175,0	116,7	87,5	70,0	58,3	50,0	43,8	38,9	35,0	31,8	29,2	26,9	25,0	23,3	21,9	20,6	19,4	18,4	17,5	16,7
8	>fmax	200,0	133,3	100,0	80,0	66,7	57,1	50,0	44,4	40,0	36,4	33,3	30,8	28,6	26,7	25,0	23,5	22,2	21,1	20,0	19,0
9	>fmax	225,0	150,0	112,5	90,0	75,0	64,3	56,3	50,0	45,0	40,9	37,5	34,6	32,1	30,0	28,1	26,5	25,0	23,7	22,5	21,4
10	>fmax	250,0	166,7	125,0	100,0	83,3	71,4	62,5	55,6	50,0	45,5	41,7	38,5	35,7	33,3	31,3	29,4	27,8	26,3	25,0	23,8
11	>fmax	>fmax	183,3	137,5	110,0	91,7	78,6	68,8	61,1	55,0	50,0	45,8	42,3	39,3	36,7	34,4	32,4	30,6	28,9	27,5	26,2
12	>fmax	>fmax	200,0	150,0	120,0	100,0	85,7	75,0	66,7	60,0	54,5	50,0	46,2	42,9	40,0	37,5	35,3	33,3	31,6	30,0	28,6
13	>fmax	>fmax	216,7	162,5	130,0	108,3	92,9	81,3	72,2	65,0	59,1	54,2	50,0	46,4	43,3	40,6	38,2	36,1	34,2	32,5	31,0
14	>fmax	>fmax	233,3	175,0	140,0	116,7	100,0	87,5	77,8	70,0	63,6	58,3	53,8	50,0	46,7	43,8	41,2	38,9	36,8	35,0	33,3
15	>fmax	>fmax	250,0	187,5	150,0	125,0	107,1	93,8	83,3	75,0	68,2	62,5	57,7	53,6	50,0	46,9	44,1	41,7	39,5	37,5	35,7
16	>fmax	>fmax	>fmax	200,0	160,0	133,3	114,3	100,0	88,9	80,0	72,7	66,7	61,5	57,1	53,3	50,0	47,1	44,4	42,1	40,0	38,1
17	>fmax	>fmax	>fmax	212,5	170,0	141,7	121,4	106,3	94,4	85,0	77,3	70,8	65,4	60,7	56,7	53,1	50,0	47,2	44,7	42,5	40,5
18	>fmax	>fmax	>fmax	225,0	180,0	150,0	128,6	112,5	100,0	90,0	81,8	75,0	69,2	64,3	60,0	56,3	52,9	50,0	47,4	45,0	42,9
19	>fmax	>fmax	>fmax	237,5	190,0	158,3	135,7	118,8	105,6	95,0	86,4	79,2	73,1	67,9	63,3	59,4	55,9	52,8	50,0	47,5	45,2
20	>fmax	>fmax	>fmax	250,0	200,0	166,7	142,9	125,0	111,1	100,0	90,9	83,3	76,9	71,4	66,7	62,5	58,8	55,6	52,6	50,0	47,6
21	>fmax	>fmax	>fmax	>fmax	210,0	175,0	150,0	131,3	116,7	105,0	95,5	87,5	80,8	75,0	70,0	65,6	61,8	58,3	55,3	52,5	50,0
22	>fmax	>fmax	>fmax	>fmax	220,0	183,3	157,1	137,5	122,2	110,0	100,0	91,7	84,6	78,6	73,3	68,8	64,7	61,1	57,9	55,0	52,4
23	>fmax	>fmax	>fmax	>fmax	230,0	191,7	164,3	143,8	127,8	115,0	104,5	95,8	88,5	82,1	76,7	71,9	67,6	63,9	60,5	57,5	54,8
24	>fmax	>fmax	>fmax	>fmax	240,0	200,0	171,4	150,0	133,3	120,0	109,1	100,0	92,3	85,7	80,0	75,0	70,6	66,7	63,2	60,0	57,1
25	>fmax	>fmax	>fmax	>fmax	250,0	208,3	178,6	156,3	138,9	125,0	113,6	104,2	96,2	89,3	83,3	78,1	73,5	69,4	65,8	62,5	59,5
26	>fmax	>fmax	>fmax	>fmax	>fmax	216,7	185,7	162,5	144,4	130,0	118,2	108,3	100,0	92,9	86,7	81,3	76,5	72,2	68,4	65,0	61,9
27	>fmax	>fmax	>fmax	>fmax	>fmax	225,0	192,9	168,8	150,0	135,0	122,7	112,5	103,8	96,4	90,0	84,4	79,4	75,0	71,1	67,5	64,3
28	>fmax	>fmax	>fmax	>fmax	>fmax	233,3	200,0	175,0	155,6	140,0	127,3	116,7	107,7	100,0	93,3	87,5	82,4	77,8	73,7	70,0	66,7
29	>fmax	>fmax	>fmax	>fmax	>fmax	241,7	207,1	181,3	161,1	145,0	131,8	120,8	111,5	103,6	96,7	90,6	85,3	80,6	76,3	72,5	69,0
31	>fmax	>fmax	>fmax	>fmax	>fmax	>fmax	221,4	193,8	172,2	155,0	140,9	129,2	119,2	110,7	103,3	96,9	91,2	86,1	81,6	77,5	73,8
32	>fmax	>fmax	>fmax	>fmax	>fmax	>fmax	228,6	200,0	177,8	160,0	145,5	133,3	123,1	114,3	106,7	100,0	94,1	88,9	84,2	80,0	76,2

Table 2: Multiplier MULT (yellow) and divisor DIV (blue) for input clock of 50 MHz and output clocks up to 250 MHz.
Found 108 MHz output clock for MULT/DIV=13/6 (108.33 MHz) or MULT/ DIV=28/13 (107.69 MHz).