Due Date: Sunday October 6, 11:59pm– Submit this editable **Word** document NOT pdf

0 / 32

EEDG/CE 6370
Design and Analysis or Reconfigurable Systems
Homework 7
VGA-FPGA Interface

Name: Prathamesh Sanjay Gadad

Email: psg220003@utdallas.edu

Part I – VGA implementation with DCM IP to set clock

a.) Follow the instructions in the lab sheet and implement the design which draws two squares of different color to the screen using Altera's IP catalogue to create a clock of different frequency (based on screen size and resolution). **Create a short video demonstrating the working design**.

Marks
4

Part I: https://youtu.be/YXev3qN_tr4

b.) Report the number of ALMs, ALUTs and critical path of the design from the synthesis report. Compute the maximum frequency. Include the screenshots generated by Quartus.

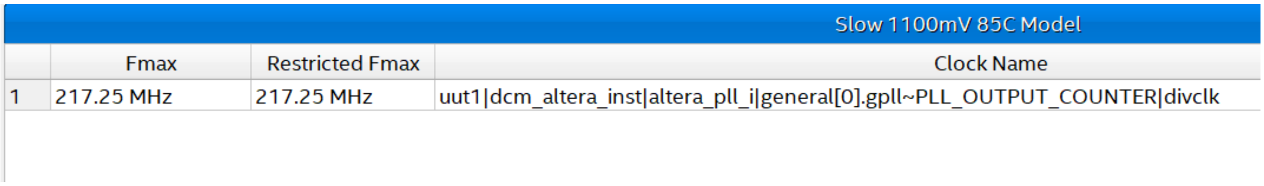
Marks
2

ALMs	ALUTs	Critical Path
43	14	Vga_controller:uut2|row[3] to blue[1]

Maximum Frequency: 120MHz

Reference:

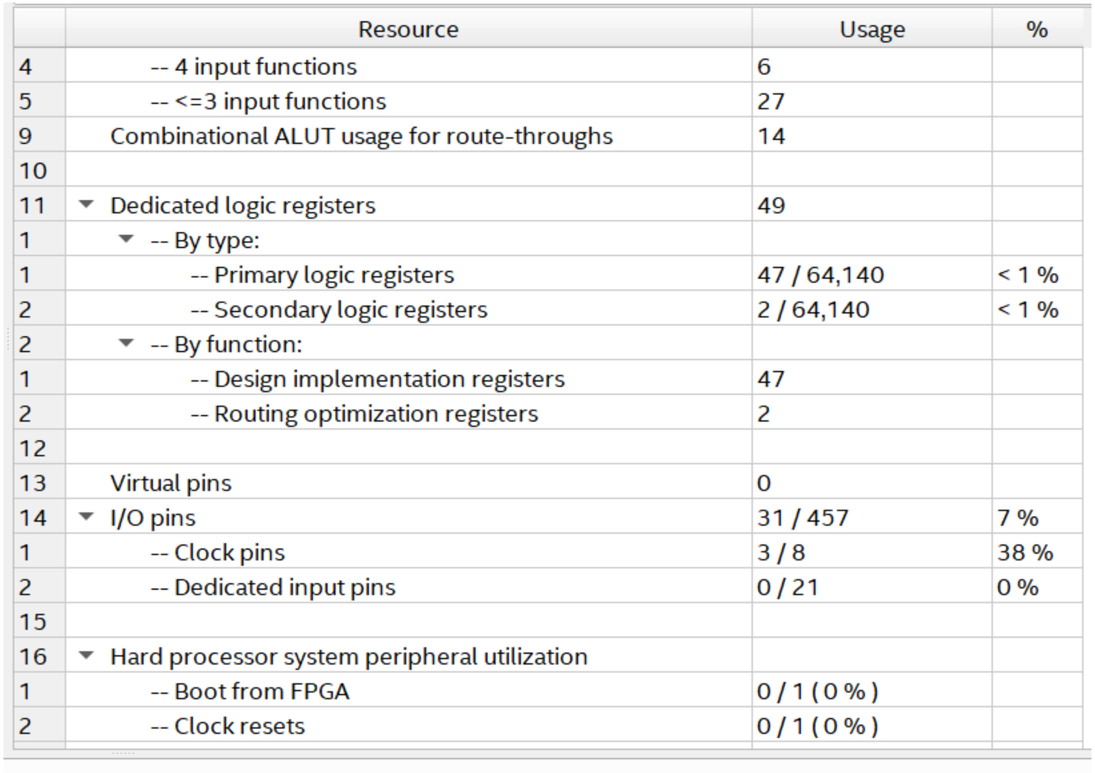
FMax



Slow 1100mV 85C Model

	Fmax	Restricted Fmax	Clock Name
1	217.25 MHz	217.25 MHz	uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk

ALUTs



	Resource	Usage	%
4	-- 4 input functions	6	
5	-- <=3 input functions	27	
9	Combinational ALUT usage for route-throughs	14	
10			
11	Dedicated logic registers	49	
1	-- By type:		
1	-- Primary logic registers	47 / 64,140	< 1 %
2	-- Secondary logic registers	2 / 64,140	< 1 %
2	-- By function:		
1	-- Design implementation registers	47	
2	-- Routing optimization registers	2	
12			
13	Virtual pins	0	
14	I/O pins	31 / 457	7 %
1	-- Clock pins	3 / 8	38 %
2	-- Dedicated input pins	0 / 21	0 %
15			
16	Hard processor system peripheral utilization		
1	-- Boot from FPGA	0 / 1 (0 %)	
2	-- Clock resets	0 / 1 (0 %)	

ALMs

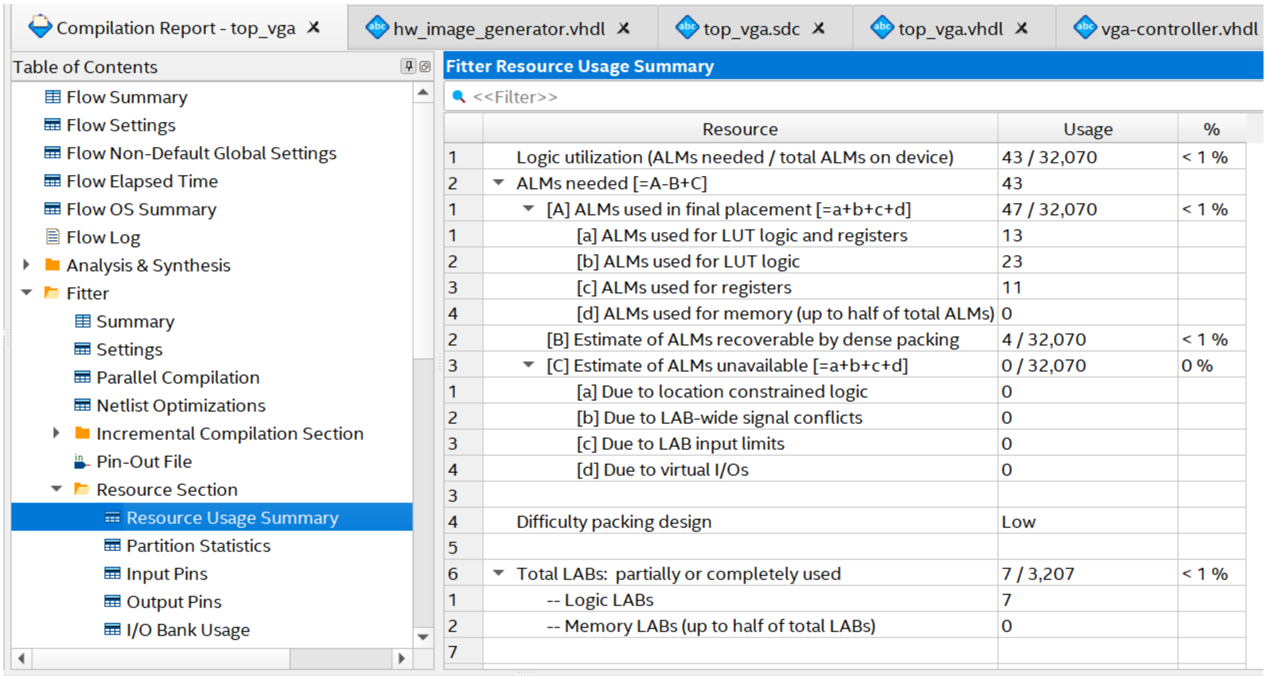


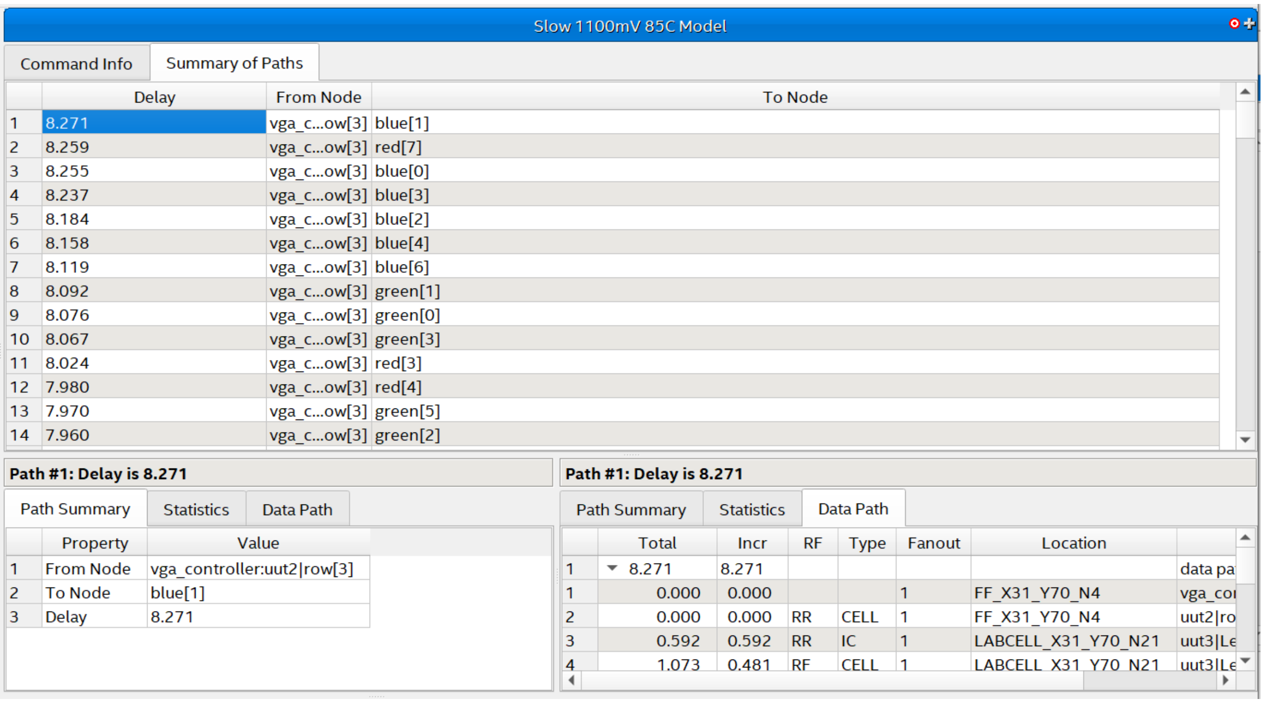
Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
  - Summary
  - Settings
  - Parallel Compilation
  - Netlist Optimizations
  - Incremental Compilation Section
  - Pin-Out File
  - Resource Section
    - Resource Usage Summary
    - Partition Statistics
    - Input Pins
    - Output Pins
    - I/O Bank Usage

Fitter Resource Usage Summary

	Resource	Usage	%
1	Logic utilization (ALMs needed / total ALMs on device)	43 / 32,070	< 1 %
2	ALMs needed [=A-B+C]	43	
1	[A] ALMs used in final placement [=a+b+c+d]	47 / 32,070	< 1 %
1	[a] ALMs used for LUT logic and registers	13	
2	[b] ALMs used for LUT logic	23	
3	[c] ALMs used for registers	11	
4	[d] ALMs used for memory (up to half of total ALMs)	0	
2	[B] Estimate of ALMs recoverable by dense packing	4 / 32,070	< 1 %
3	[C] Estimate of ALMs unavailable [=a+b+c+d]	0 / 32,070	0 %
1	[a] Due to location constrained logic	0	
2	[b] Due to LAB-wide signal conflicts	0	
3	[c] Due to LAB input limits	0	
4	[d] Due to virtual I/Os	0	
3			
4	Difficulty packing design	Low	
5			
6	Total LABs: partially or completely used	7 / 3,207	< 1 %
1	-- Logic LABs	7	
2	-- Memory LABs (up to half of total LABs)	0	
7			

Critical Path & Delay



Slow 1100mV 85C Model

	Delay	From Node	To Node
1	8.271	vga_c...ow[3]	blue[1]
2	8.259	vga_c...ow[3]	red[7]
3	8.255	vga_c...ow[3]	blue[0]
4	8.237	vga_c...ow[3]	blue[3]
5	8.184	vga_c...ow[3]	blue[2]
6	8.158	vga_c...ow[3]	blue[4]
7	8.119	vga_c...ow[3]	blue[6]
8	8.092	vga_c...ow[3]	green[1]
9	8.076	vga_c...ow[3]	green[0]
10	8.067	vga_c...ow[3]	green[3]
11	8.024	vga_c...ow[3]	red[3]
12	7.980	vga_c...ow[3]	red[4]
13	7.970	vga_c...ow[3]	green[5]
14	7.960	vga_c...ow[3]	green[2]

Path #1: Delay is 8.271

	Property	Value
1	From Node	vga_controller:uut2|row[3]
2	To Node	blue[1]
3	Delay	8.271

Path #1: Delay is 8.271

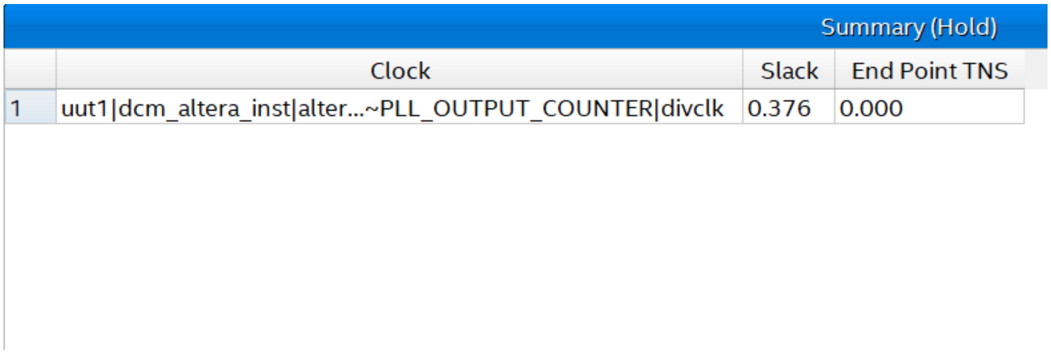
	Total	Incr	RF	Type	Fanout	Location	
1	8.271	8.271					data pa
1	0.000	0.000			1	FF_X31_Y70_N4	vga_co
2	0.000	0.000	RR	CELL	1	FF_X31_Y70_N4	uut2|ro
3	0.592	0.592	RR	IC	1	LABCELL_X31_Y70_N21	uut3|Le
4	1.073	0.481	RF	CELL	1	LABCELL_X31_Y70_N21	uut3|Le

Part II – Timing analysis

c.) Open TimeQuest. 1) Netlist → Create Timing netlist. 2.) Read .sdc file from VGA project (Constraints → read sdc file). 3.) Update timing netlist (Netlist → Update Timing netlist) Report the slack of the circuit. Is it positive or negative? Explain what that means (Reports → slack → Report setup summary and Report hold summary.

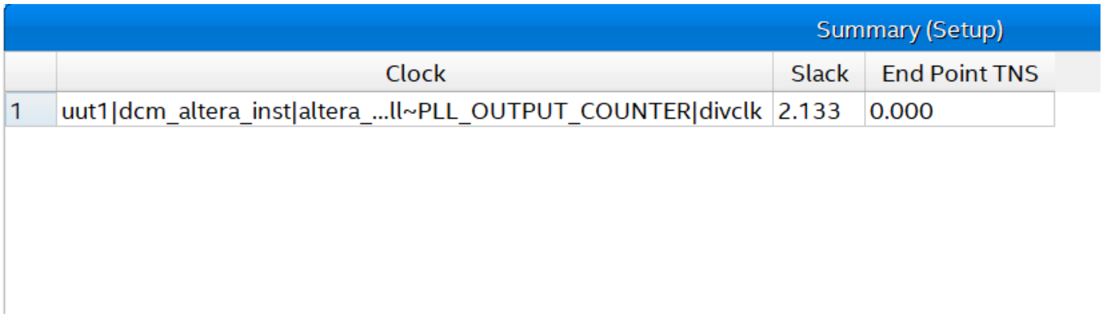
Marks
2

In the setup and hold summary the slack is positive, which suggests that the circuit meets the timing requirements, which is a good outcome.



Summary (Hold)

	Clock	Slack	End Point TNS
1	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk	0.376	0.000

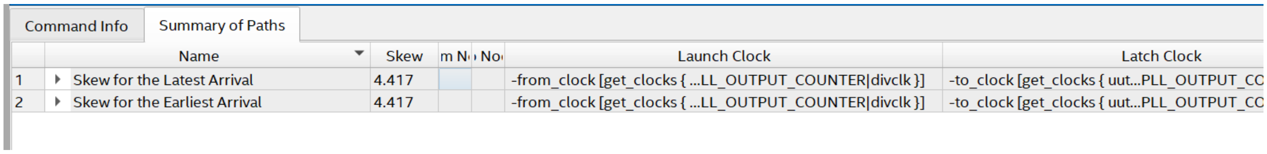


Summary (Setup)

	Clock	Slack	End Point TNS
1	uut1|dcm_altera_inst|altera_...ll~PLL_OUTPUT_COUNTER|divclk	2.133	0.000

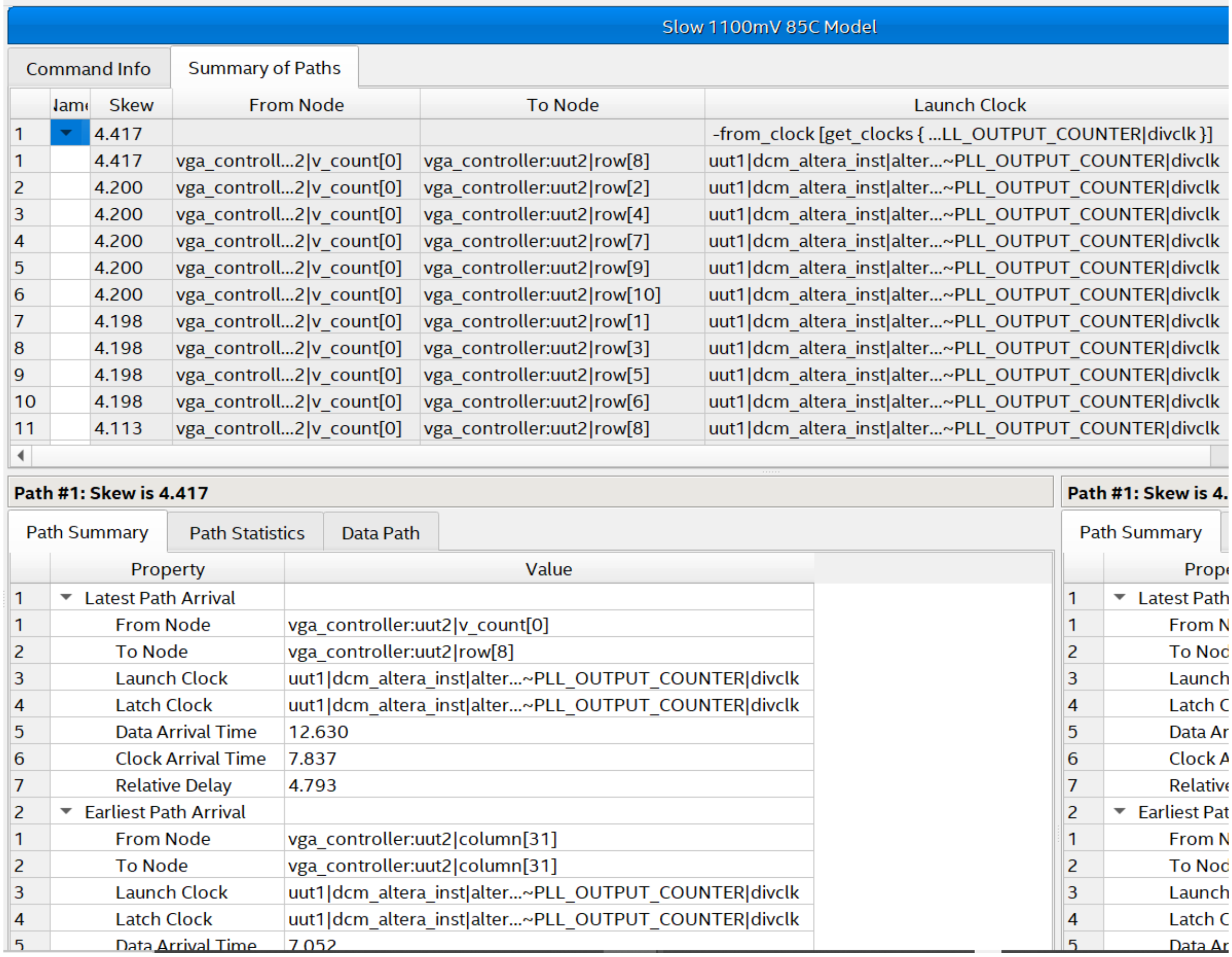
e.) Reporting the skew. Task pane → Custom reports → report skew. Edit “From clock” and “To clock” combo boxes. Which clock should you choose? Explain why.

Marks
2



Command Info | Summary of Paths

	Name	Skew	m N	No	Launch Clock	Latch Clock
1	Skew for the Latest Arrival	4.417			-from_clock [get_clocks { ...LL_OUTPUT_COUNTER|divclk }]	-to_clock [get_clocks { uut...PLL_OUTPUT_CO
2	Skew for the Earliest Arrival	4.417			-from_clock [get_clocks { ...LL_OUTPUT_COUNTER|divclk }]	-to_clock [get_clocks { uut...PLL_OUTPUT_CO



Slow 1100mV 85C Model

Command Info | Summary of Paths

	lame	Skew	From Node	To Node	Launch Clock
1		4.417			-from_clock [get_clocks { ...LL_OUTPUT_COUNTER|divclk }]
1		4.417	vga_controll...2|v_count[0]	vga_controller:uut2|row[8]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
2		4.200	vga_controll...2|v_count[0]	vga_controller:uut2|row[2]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
3		4.200	vga_controll...2|v_count[0]	vga_controller:uut2|row[4]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
4		4.200	vga_controll...2|v_count[0]	vga_controller:uut2|row[7]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
5		4.200	vga_controll...2|v_count[0]	vga_controller:uut2|row[9]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
6		4.200	vga_controll...2|v_count[0]	vga_controller:uut2|row[10]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
7		4.198	vga_controll...2|v_count[0]	vga_controller:uut2|row[1]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
8		4.198	vga_controll...2|v_count[0]	vga_controller:uut2|row[3]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
9		4.198	vga_controll...2|v_count[0]	vga_controller:uut2|row[5]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
10		4.198	vga_controll...2|v_count[0]	vga_controller:uut2|row[6]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
11		4.113	vga_controll...2|v_count[0]	vga_controller:uut2|row[8]	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk

Path #1: Skew is 4.417

Path Summary | Path Statistics | Data Path

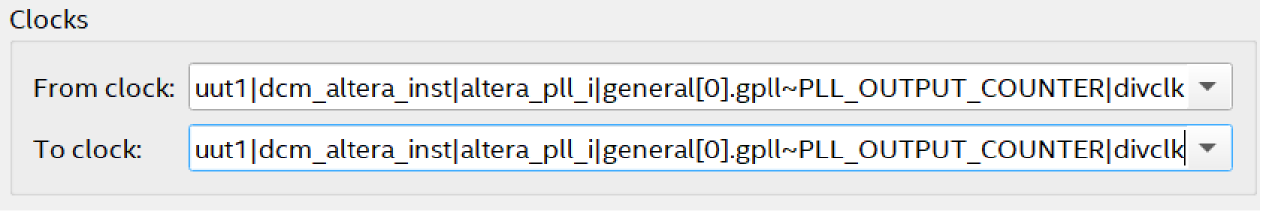
	Property	Value
1	Latest Path Arrival	
1	From Node	vga_controller:uut2|v_count[0]
2	To Node	vga_controller:uut2|row[8]
3	Launch Clock	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
4	Latch Clock	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
5	Data Arrival Time	12.630
6	Clock Arrival Time	7.837
7	Relative Delay	4.793
2	Earliest Path Arrival	
1	From Node	vga_controller:uut2|column[31]
2	To Node	vga_controller:uut2|column[31]
3	Launch Clock	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
4	Latch Clock	uut1|dcm_altera_inst|alter...~PLL_OUTPUT_COUNTER|divclk
5	Data Arrival Time	7.052

Path #1: Skew is 4.

Path Summary

	Prop
1	Latest Path
1	From N
2	To Noc
3	Launch
4	Latch C
5	Data Ar
6	Clock A
7	Relative
2	Earliest Pat
1	From N
2	To Noc
3	Launch
4	Latch C
5	Data Ar

Clocks Selected



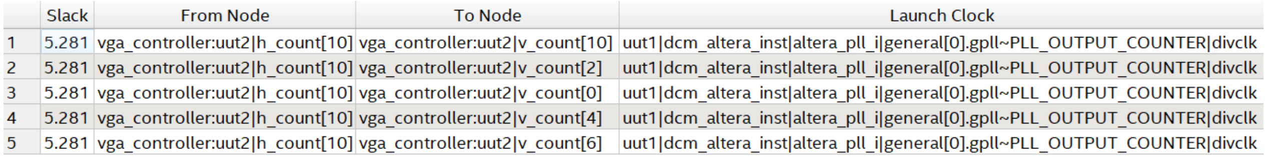
Clocks

From clock:	uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk
To clock:	uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk

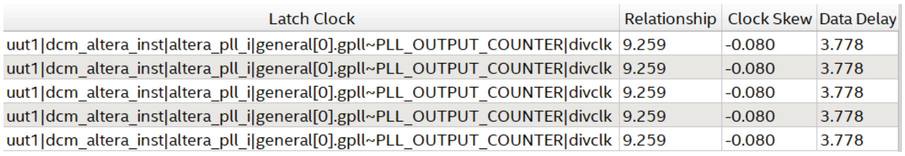
We selected the above clock because the output signals are driven by the PLL-generated clocks. This can be verified by running a Timing Report, which shows that the paths are derived from the PLL clock. The PLL, in this case, is a clock generated using the IP.

f.) Is the slack positive or negative? What does the slack result obtained suggests?

Marks
2



	Slack	From Node	To Node	Launch Clock
1	5.281	vga_controller:uut2|h_count[10]	vga_controller:uut2|v_count[10]	uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk
2	5.281	vga_controller:uut2|h_count[10]	vga_controller:uut2|v_count[2]	uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk
3	5.281	vga_controller:uut2|h_count[10]	vga_controller:uut2|v_count[0]	uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk
4	5.281	vga_controller:uut2|h_count[10]	vga_controller:uut2|v_count[4]	uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk
5	5.281	vga_controller:uut2|h_count[10]	vga_controller:uut2|v_count[6]	uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk



Latch Clock	Relationship	Clock Skew	Data Delay
uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk	9.259	-0.080	3.778
uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk	9.259	-0.080	3.778
uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk	9.259	-0.080	3.778
uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk	9.259	-0.080	3.778
uut1|dcm_altera_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk	9.259	-0.080	3.778

The Slack is 'Positive,' indicating that the design meets the timing requirements.

Part III – Power Estimation

a.) Use the PowerPlay Excel sheet to estimate the total power of the design. Enter manually the resources used. Let the toggle rate at its default value. Annotate the estimated dynamic and static power, with the following values. Discuss the results.

Marks
6

Configs	Frequency [MHz]	Temperature [ºC]	Toggle rate [%]	Static power (Quiescent) [W]	Dynamic Power [W]	Total Power [W]
Config 1	50	50	12.5			
Config 2	60	50	12.5			
Config 3	70	50	12.5			
Config 4	50	55	12.5			
Config 5	50	60	12.5			

Config 6	50	65	12.5			
Config 7	50	50	15			
Config 8	50	50	20			
Config 9	50	50	25			

Compare the results obtained from Power Play Early Power Estimator (Excel spreadsheet) with the results obtained from Power Play Power Analyzer (Quartus II power estimation tool). Do the results match? Explain why yes/no.

Marks
4

Power Play Early Power Estimator,

Configs	Frequency [MHz]	Temperature [ºC]	Toggle rate [%]	Static power (Quiescent) [W]	Dynamic Power [W]	Total Power [W]
Config 1	50	50	12.5	0.319	0.017	0.336
Config 2	60	50	12.5	0.319	0.017	0.336
Config 3	70	50	12.5	0.319	0.017	0.336
Config 4	50	55	12.5	0.328	0.016	0.344
Config 5	50	60	12.5	0.337	0.016	0.353
Config 6	50	65	12.5	0.347	0.016	0.363
Config 7	50	50	13.75	0.319	0.019	0.338
Config 8	50	50	15	0.319	0.021	0.340
Config 9	50	50	16.25	0.319	0.023	0.342

Power Play Analyzer,

Configs	Frequency [MHz]	Temperature [ºC]	Toggle rate [%]	Static power (Quiescent) [W]	Dynamic Power [W]	Total Power [W]
Config 1	50	50	12.5	0.456	0.011	0.480
Config 2	60	50	12.5	0.457	0.011	0.481
Config 3	70	50	12.5	0.457	0.013	0.484
Config 4	50	55	12.5	0.468	0.011	0.491
Config 5	50	60	12.5	0.479	0.011	0.503

Config 6	50	65	12.5	0.492	0.011	0.516
Config 7	50	50	13.75	0.457	0.011	0.483
Config 8	50	50	15	0.457	0.011	0.485
Config 9	50	50	16.25	0.457	0.011	0.487

The outcomes vary because the Early Power Estimator relies only on the provided inputs and lacks detailed design aspects like configuration, routing, and other factors. On the other hand, the Power Analyzer delivers more precise results as it considers a full range of details, including design configurations, routing, and resource usage.

Part IV – Re-doing VGA Output

Re-do the design and display one of the following messages (choose one) on the center of the screen in UTD green or orange:

- I love FPGAs
- UT Dallas
- HW > SW

The screen background should be white background (full display). Write down the RTL code that generates this image from the 'hw_image_generator' module. Only the code snippet. Adjust the monitor size parameters (porch values and horizontal and vertical pixels, etc..). Include the parameters settings here. Finally create a YouTube video showing that this works

Marks
10

Part II: https://youtu.be/45BhbDg42f4

Code Snippet:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY hw_image_generator IS
  GENERIC(
    pixels_y :  INTEGER := 478;   -- row that first color will persist until
    pixels_x :  INTEGER := 600     -- column that first color will persist
until
  );
  PORT(
    disp_ena :  IN   STD_LOGIC;  -- display enable ('1' = display time, '0' =
blanking time)
    row      :  IN   INTEGER;    -- row pixel coordinate
    column   :  IN   INTEGER;    -- column pixel coordinate
    red      :  OUT  STD_LOGIC_VECTOR(7 DOWNTO 0) := (OTHERS => '0');  -- red
magnitude output to DAC
    green    :  OUT  STD_LOGIC_VECTOR(7 DOWNTO 0) := (OTHERS => '0');  --
green magnitude output to DAC
    blue     :  OUT  STD_LOGIC_VECTOR(7 DOWNTO 0) := (OTHERS => '0')   -- blue
magnitude output to DAC
  );
END hw_image_generator;

ARCHITECTURE behavior OF hw_image_generator IS

  -- Character 'S' bitmap in 20x100 format, each bit represents a 20x20 block
  TYPE bitmap_array IS ARRAY (0 TO 19) OF STD_LOGIC_VECTOR(99 DOWNTO 0);
  CONSTANT char_S : bitmap_array := (

"00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000",

"00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000",

"00010000000010001111111110000011111111100111111111001000000000100000000001011
11111100111111110000000",

"00010000000010000000100000000010000000100100000001001000000000100000000001000
00000100100000000000000",

"00010000000010000000100000000010000000100100000001001000000000100000000001000
00000100100000000000000",

"00010000000010000000100000000010000000100100000001001000000000100000000001000
00000100100000000000000",

"00010000000010000000100000000010000000100111111111001000000000100000000001011
11111100100000000000000",

"00010000000010000000100000000010000000100100000001001000000000100000000001000
00000100111111110000000",

"00010000000010000000100000000010000000100100000001001000000000100000000001000
00000100000000010000000",

"00010000000010000000100000000010000000100100000001001000000000100000000001000
00000100000000010000000",
"00010000000010000000100000000010000000100100000001001000000000100000000001000
00000100000000010000000",

"00011111111110000000100000000011111111100100000001001111111100111111111001000
00000100111111110000000",

"00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000",

"00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000",

"00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000",

"00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000",

"00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000",

"00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000",

"00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000",

"00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000"
  );

BEGIN
  PROCESS(disp_ena, row, column)
  BEGIN

    IF (disp_ena = '1') THEN  -- display time
      -- Set the background to white
      red <= (OTHERS => '1');
      green <= (OTHERS => '1');
      blue <= (OTHERS => '1');

-- Assuming char_S is a 20x100 array of std_logic representing the character
bitmap
-- This example assumes char_S is declared as follows:
-- type char_array is array (0 to 19) of std_logic_vector(99 downto 0);
-- signal char_S : char_array;

-- Loop through the character bitmap for "S"
    FOR i IN 0 TO 19 LOOP  -- 20 rows (flipped)
    IF (row >= (450 + i * 10) AND row < (470 + i * 10)) THEN  -- Scale rows by
10
        FOR j IN 0 TO 99 LOOP  -- 100 columns
```

```
            IF (column >= (550 + j * 10) AND column < (570 + j * 10)) THEN  --
Scale columns by 10
                IF (char_S(i)(99 - j) = '1') THEN  -- Flip the bitmap
horizontally
                    -- Set pixel color to black for "S"
                    red <= X"FF";   -- Assign all bits to '0'
                    green <= X"A5"; -- Assign all bits to '0'
                    blue <=  X"00";  -- Assign all bits to '0'
                END IF;
            END IF;
        END LOOP;
    END IF;
END LOOP;




    ELSE  -- blanking time
      red <= (OTHERS => '0');
      green <= (OTHERS => '0');
      blue <= (OTHERS => '0');
    END IF;

  END PROCESS;
END behavior;
```

Parameter Settings for the Monitor:

```
  h_pulse  :  INTEGER   := 44;   --horiztonal sync pulse width in pixels
    h_bp     :  INTEGER   := 148;   --horiztonal back porch width in pixels
    h_pixels :  INTEGER   := 1920;  --horiztonal display width in pixels
    h_fp     :  INTEGER   := 88;   --horiztonal front porch width in pixels
    h_pol    :  STD_LOGIC := '0';   --horizontal sync pulse polarity (1 =
positive, 0 = negative)
    v_pulse  :  INTEGER   := 5;     --vertical sync pulse width in rows
    v_bp     :  INTEGER   := 36;    --vertical back porch width in rows
    v_pixels :  INTEGER   := 1080;  --vertical display width in rows
    v_fp     :  INTEGER   := 4;     --vertical front porch width in rows
    v_pol    :  STD_LOGIC := '1');  --vertical sync pulse polarity (1 =
positive, 0 = negative)
```