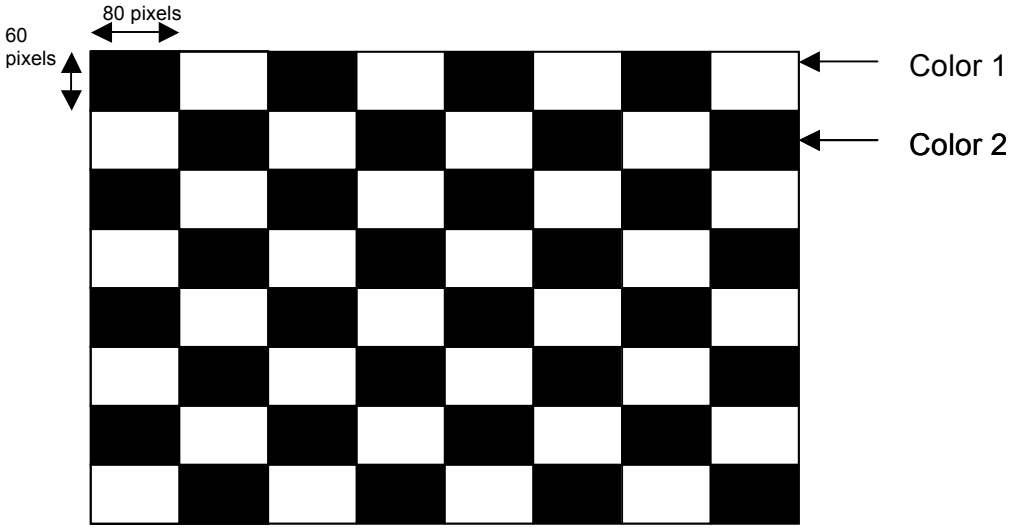
Experiment 6

Design and implement a digital circuit capable of displaying predefined patterns on the screen of a VGA monitor, and provide the basic components for a chess game. Your circuit must generate all control and data signals driving the VGA output of the BASYS2 board. Please refer to the introductory lab slides for information about the meaning and timing of VGA control signals. Use the clock frequency of 25 MHz, and the resolution as close as possible to 640x480 pixels. This Lab can be done in teams or individually, but teams are required to finish more tasks.

Task 1 (for individuals: required, 2 points; for teams: required, 1.5 points)

Design a circuit to display an 8x8 chess board with two alternating colors. Each box should be 80 pixels wide and 60 pixels high. Pressing button 0 will cycle through 16 different colors for one set of boxes. Button 1 will cycle through 16 different colors for the other set of boxes. Bellow is a diagram of the chess board:

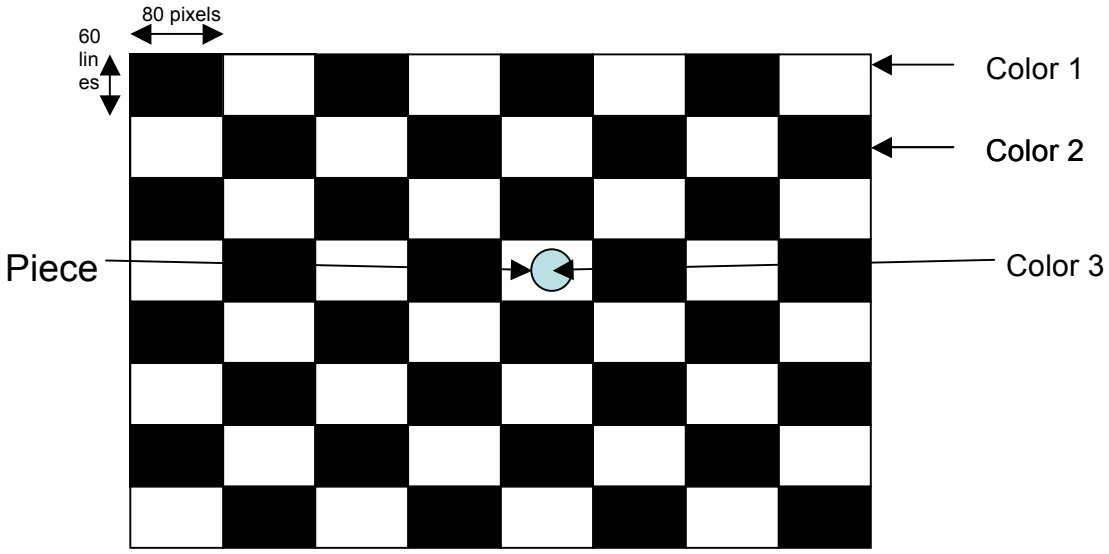


Switch Meanings				
	7	6-2	1	0
	'0'	Don't Care	'0'	'0'

Button 0 : cycle through 16 colors for color 1
Button 1 : cycle through 16 colors for color 2

Task 2 (for individuals: required, 2 points; for teams: required, 1.5 points)

Expand task 1 by putting a chess piece in the center of one of the boxes. If switch 3 is up (1) the piece is a rook represented by a square. If switch 3 is down (0) then the piece is a bishop represented by a circle. During this task all switches should be off except switch 0 (should be on) and switch 3 (which is used to determine the type of piece). Also buttons 0 and 1 should have the same functionality as in task 1. In addition pressing button 2 should cause the color of the piece to cycle through 16 different colors.



Switch Meanings					
	7	6-3	2	1	0
	'0'	"0000"	if 0 piece is a bishop IF 1 piece is a castle	'0'	'1'

Button 0 : cycle through 16 colors for color 1
Button 1 : cycle through 16 colors for color 2
Button 2 : cycle through 16 colors for color 3

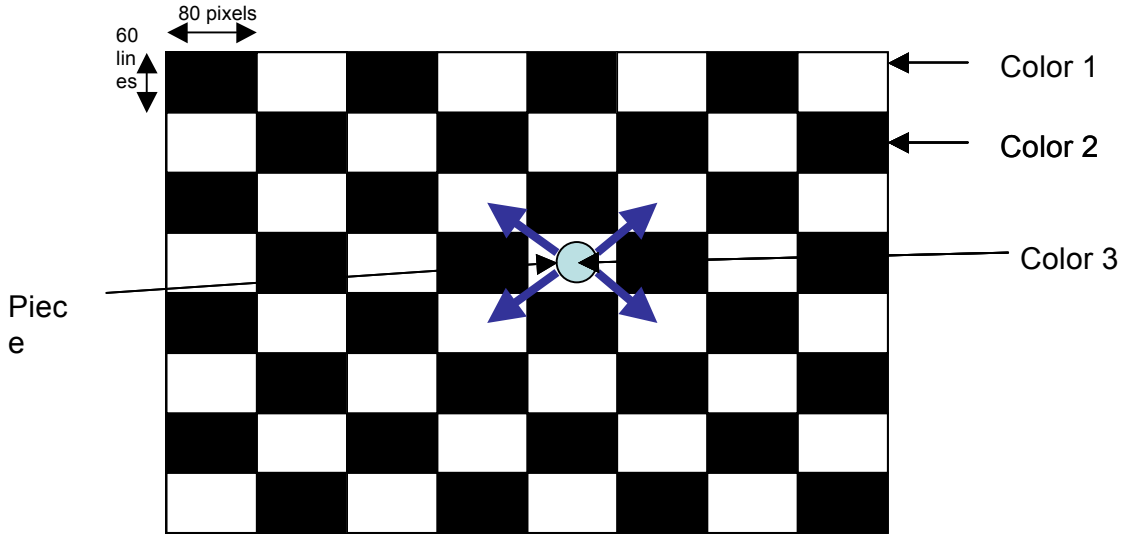
Task 3 (for individuals: required, 2 points; for teams: required, 1.5 points)

Extend Task 2 to allow the piece to move one square in any direction that is legal for that piece to move. i.e the bishop can only move on the diagonals and the rook can only move the vertical or horizontal directions. Switch 7 will act as the trigger to change the buttons behavior from changing colors to moving in a direction. If the switch is off then the buttons have their old behavior of changing colors, but if switch 7 is on then the buttons have the following behavior depending on whether a bishop or a rook is selected:

If piece is a **rook**	If piece is a **bishop**
Button 0 : Move piece one box up	Button 0 : Move piece diagonally to the upper left
Button 1 : Move piece one box down	Button 1 : Move piece diagonally to the upper right

Button 2 : Move piece one box left	Button 2 : Move piece diagonally to the lower left
Button 3 : Move piece one box right	Button 3 : Move piece diagonally to the lower right

All switches should be off except switch 7, 2 (which determine type of piece and color/movement mode) and 1 (which should be on to indicate that this is task 3) Bellow is a diagram for the case when the bishop is selected.



Switch Meanings				
7	6-3	2	1	0
Move '1' or change color '0'	"0000"	if 0 piece is a circle IF 1 piece is a squre	'1'	0'

Task 4 (for individuals: BONUS, 2 points; for teams: required, 1.5 points)

Extend task 3 to allow the piece to move n number of spaces not just one. The number of spaces to move in a given direction will be provided in binary on switches 6-4. Switches 0 and 1 should be set to on ('1') indicating that this is task 4, and switch 3 should be off. If a movement cannot be legally done (e.g., the move would cause the piece to go off the board) LED 7 should light up and the piece should not move from its previous location.

Task 5 (for individuals: BONUS, 2 points; for teams: BONUS, 1.5 points)

Add a Knight to the possible pieces if switch (3) is on. Use the buttons however you see fit to allow the knight to move in the 8

legal ways. All other functions of the switches should be the same. (e.x. if switch 7 is off the button should cycle through the colors.)

For the all tasks, determine the following properties:

1. Number of CLB slices
2. Minimum clock period after synthesis [ns]
3. Maximum clock frequency after synthesis [MHz]
4. Minimum clock period after implementation [ns]
5. Maximum clock frequency after implementation [MHz]
6. Minimum latency after implementation [ns]
7. Maximum throughput after implementation [bits/s]

As a part of the design process:

1. Draw an optimized block diagram of your color and sync generator circuits.
2. Translate your block diagram (and the rest of the circuits) into RTL VHDL.
3. Write a separate testbench capable of verifying the functionality of the sync and the color generator
4. Synthesize, implement, and load your design on the Basys board to verify functionality.

In the lab report include:

1. Optimized block diagrams (hand-drawn hardcopy submitted in class and used during an exit quiz, and an electronic copy submitted using Blackboard in the pdf or MS Word format (preferably, the scanned version of the hand-drawn hardcopy).
2. VHDL source codes for:
    a. Clock management unit
    b. VGA Sync generator
    c. Color Generator
    d. Top Level Unit
3. User Constraint File (UCF)
4. Testbench and waveforms from the functional simulation (electronic versions submitted using Blackboard) for both filters.

Important Dates

	Monday section	Tuesday section	Wednesday section	Thursday section
Hands-on Session and Introduction to the Experiment	03/29/2010	03/30/2010	03/31/2010	04/01/2010
Demonstration and Deliverables Due	**04/12/2010**	**04/13/2010**	**04/14/2010**	**04/15/2010**