

Lab 09
Flip-Flops and Counters

ECE 380-002
University of Alabama

Yichen Huang
Thomas Dillman

2019/11/11

Introduction

In this lab, we mainly focusing on the design two types of flip-flops: one is edge-triggered D flip-flop and another one is JK flip-flop. Two create two components through body diagram files and VHDL code. We also created a BDC counter in two ways those are through VHDL code and another one is use D flip-flop. Finally, we upload these circuits to the DE1 board to test the circuit.

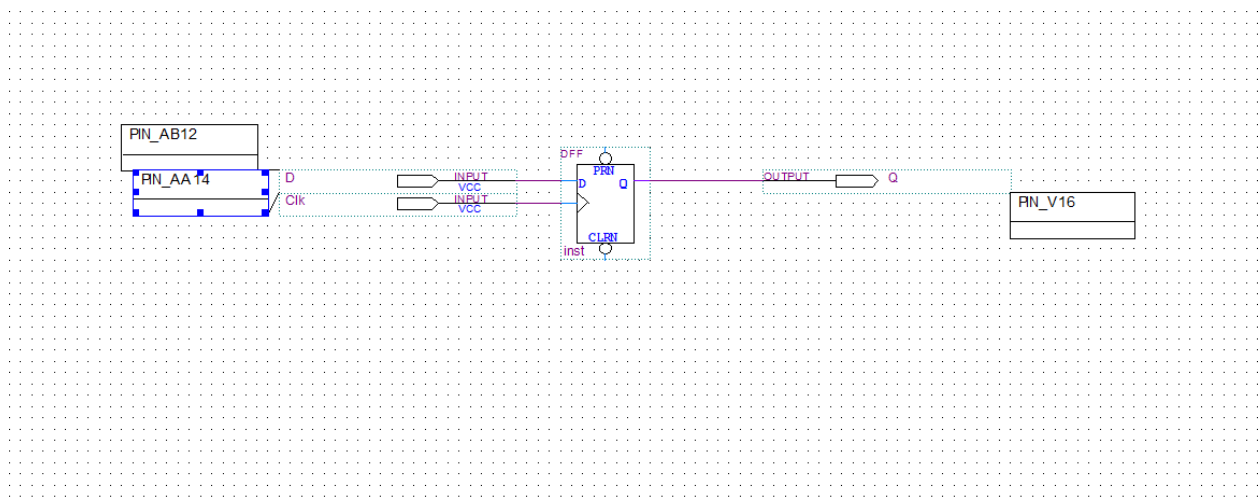
Procedure

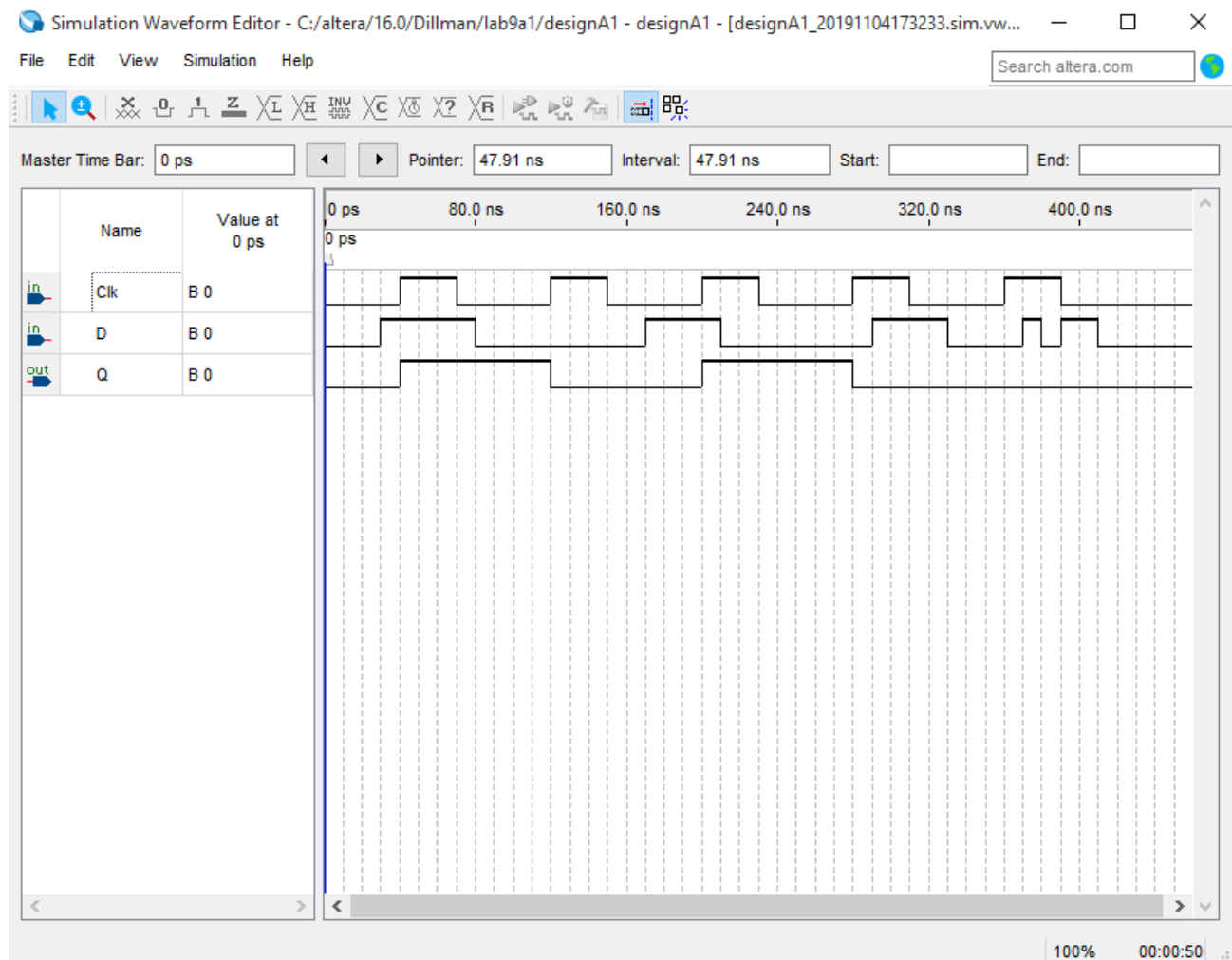
A. Prelab

a. Design A

(1) BDF File

We create a BDF file and implement a D flip-flop; after we compiled, we run the test through waveform file.





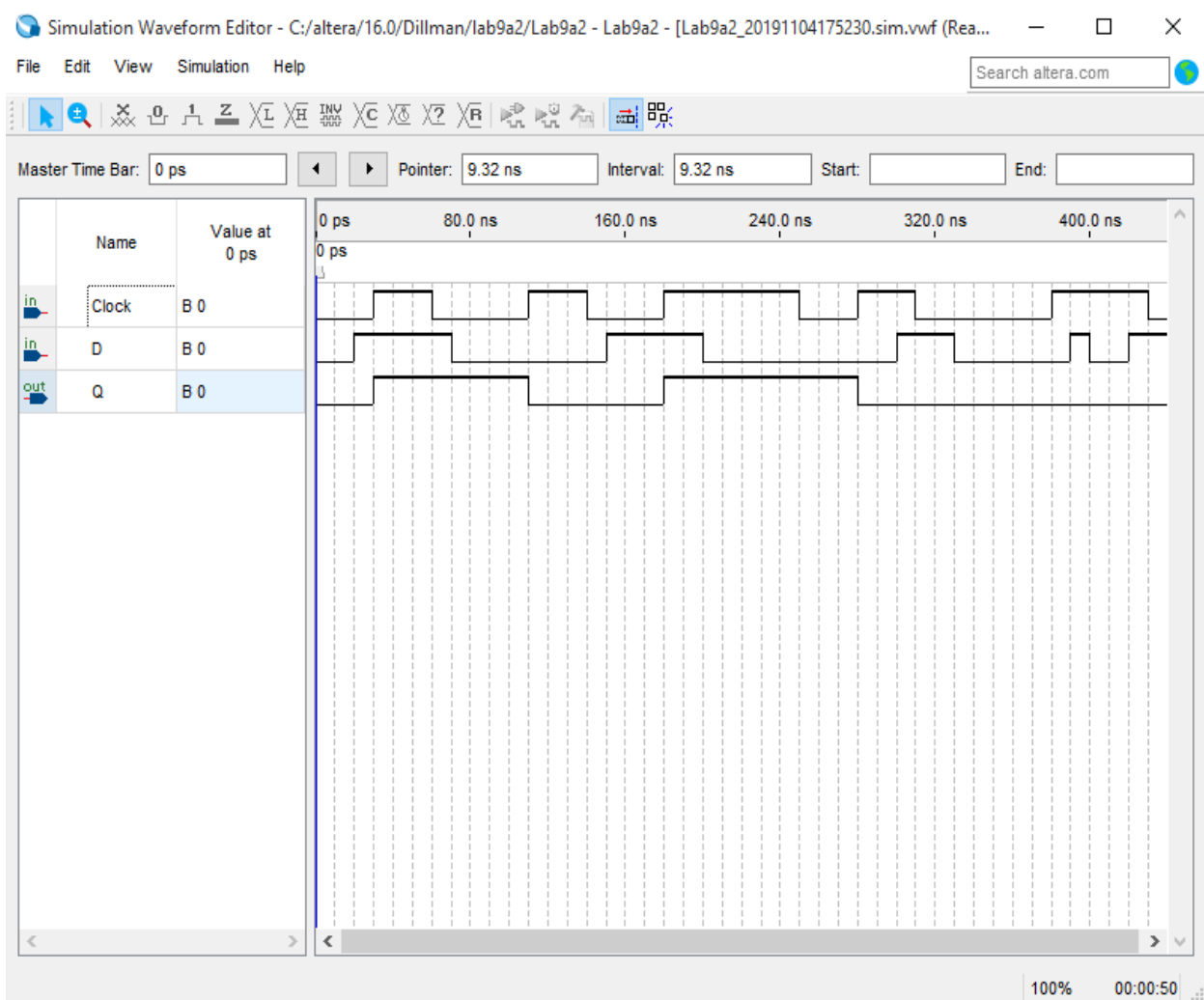
(2) VHDL File

We create D flip-flop through VHDL code. After we compile the file, we run the waveform file to test our design.

```

1  LIBRARY ieee ;
2  USE ieee.std_logic_1164.all ;
3  ENTITY Lab9a2 IS
4  PORT ( D, clock : IN STD_LOGIC ;
5        Q : OUT STD_LOGIC) ;
6  END Lab9a2 ;
7  ARCHITECTURE Behavior OF Lab9a2 IS
8  BEGIN
9  PROCESS ( clock )
10 BEGIN
11 IF clock'EVENT AND clock = '1' THEN
12   Q <= D ;
13 END IF ;
14 END PROCESS ;
15 END Behavior ;

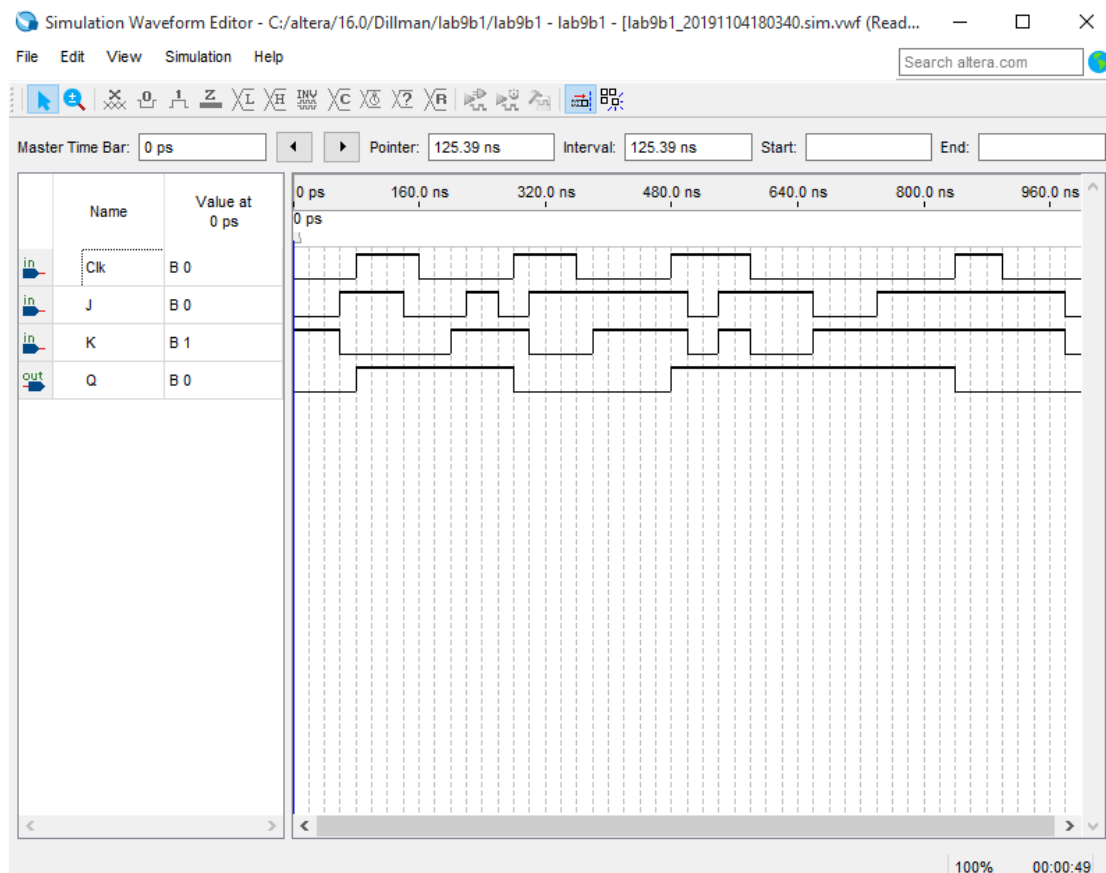
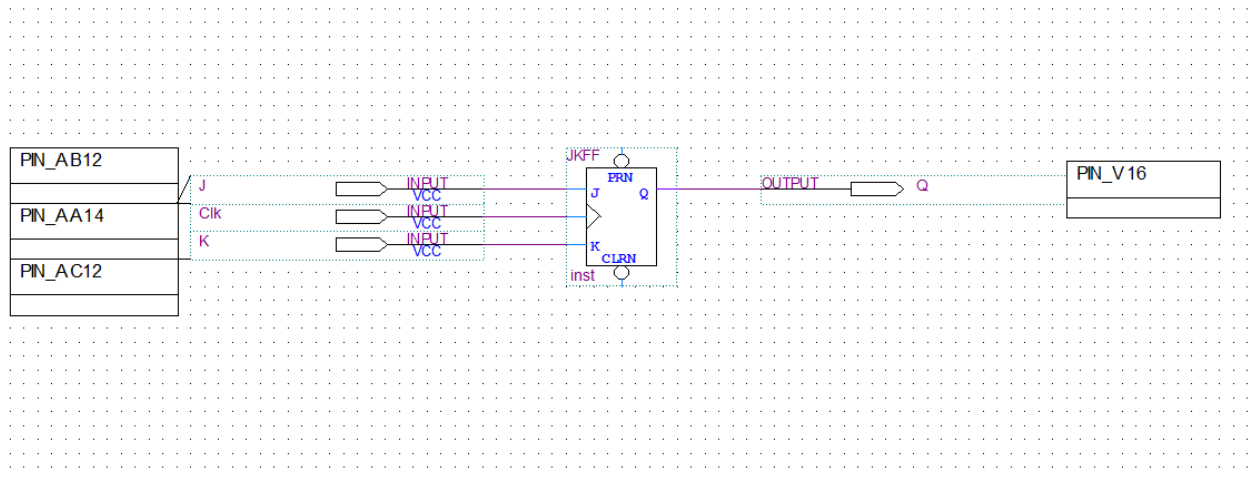
```



b. Design B

(1) BDF File

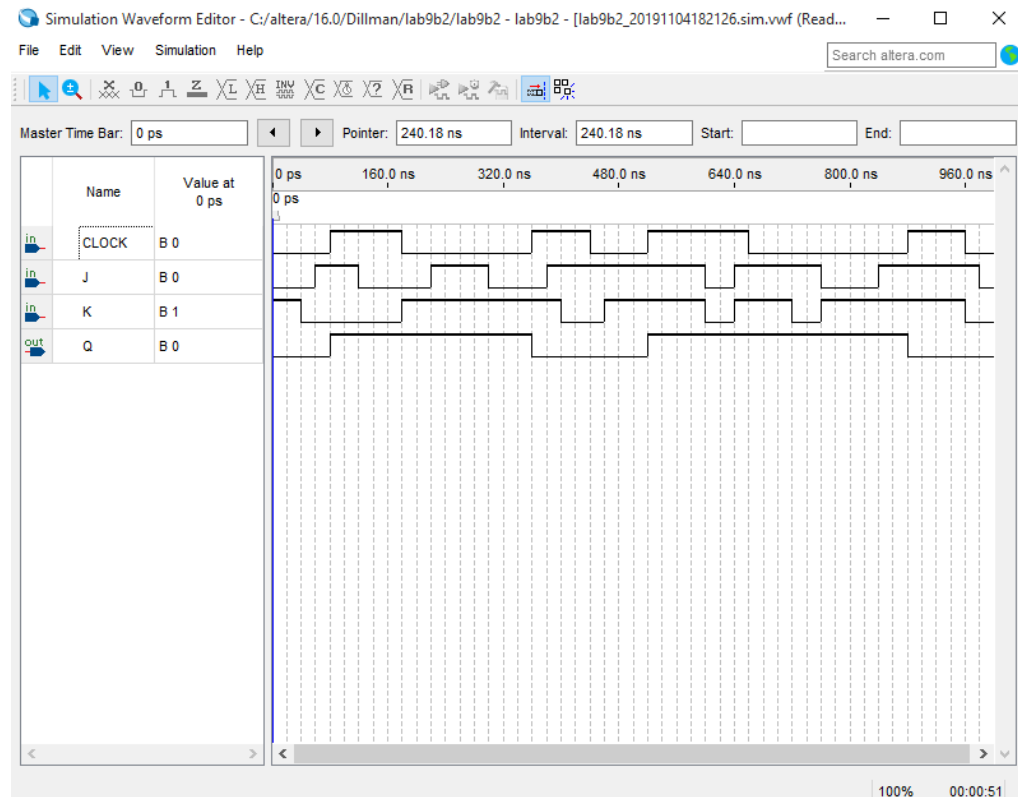
We create the JK flip-flop through BDF file. After we compile the file, we run the waveform file to test it.



(2) VHDL File

We create the JK flip-flop through VHDL code and then we test the file through the waveform file.

```
1  library ieee;
2  use ieee. std_logic_1164.all;
3  use ieee. std_logic_arith.all;
4  use ieee. std_logic_unsigned.all;
5
6  entity lab9b2 is
7  PORT( J,K,CLOCK: in std_logic;
8  Q : out std_logic);
9  end lab9b2;
10
11  Architecture behavioral of lab9b2 is
12  begin
13  PROCESS(CLOCK)
14  | variable TMP: std_logic;
15  | begin
16  | if(CLOCK='1' and CLOCK'EVENT) then
17  | if(J='0' and K='0')then
18  | TMP:=TMP;
19  | elsif(J='1' and K='1')then
20  | TMP:= not TMP;
21  | elsif(J='0' and K='1')then
22  | TMP:='0';
23  | else
24  | TMP:='1';
25  | end if;
26  | end if;
27  | Q<=TMP;
28  | end PROCESS;
29  end behavioral;
```



c. Design C

(1) VHDL

In this design, we implement a BCD counter through VHDL code.

In this design, we have three input: Enable, Reset and CLK and 4-bits output Q.

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_arith.all;
4  use ieee.std_logic_unsigned.all;
5  entity lab9c1 is
6  port(Clk: in std_logic; clear: in std_logic; Enable : in std_logic;
7       Q : out std_logic_vector(3 downto 0));
8  end lab9c1;
9  architecture RTL of lab9c1 is
10     signal Count: std_logic_vector(3 downto 0);
11     begin
12     lab9c1: process(Clk,clear)
13     begin
14         if (clear = '0')then
15             Count <= (others => '0');
16         elsif (Clk'event and Clk = '1')then
17             if (Enable = '1')then
18                 if (Count = "1001")then
19                     Count <= (others => '0');
20                 else
21                     Count <= Count + 1;
22                 end if;
23             end if;
24         end if;
25     end process lab9c1;
26     Q <= Count;
27 end RTL;

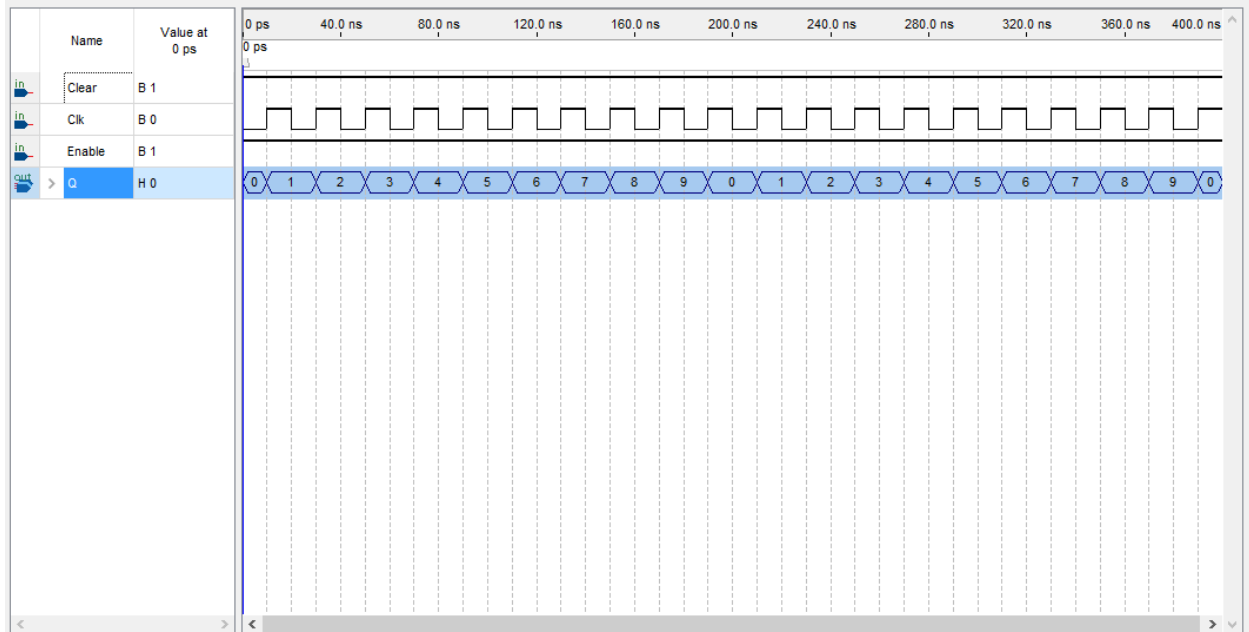
```

File Edit View Simulation Help

Search altera.com



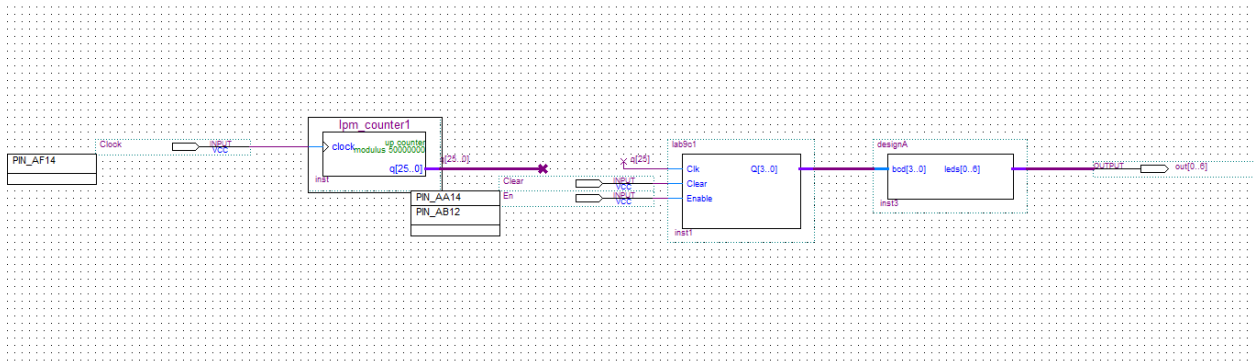
Master Time Bar: 0 ps Pointer: 318.72 ns Interval: 318.72 ns Start: 0 ps End: 400.0 ns



0% 00:00:00

(2) BCD Counter with LED display

In this design we create a BEF file and implement the symbols from previous design and lab2 design. We then design a code converter from lab8.



```
1  LIBRARY ieee ;
2  USE ieee.std_logic_1164.all ;
3  ENTITY designA IS
4  PORT ( bcd: IN STD_LOGIC_VECTOR(3 DOWNTO 0) ;
5        leds: OUT STD_LOGIC_VECTOR(0 TO 6) ) ;
6  END designA ;
7  BEGIN
8  ARCHITECTURE Behavior OF designA IS
9  PROCESS ( bcd )
10 BEGIN
11 CASE bcd IS -- abcdefg
12 WHEN "0000" => leds <= "0000001";
13 WHEN "0001" => leds <= "1001111";
14 WHEN "0010" => leds <= "0010010";
15 WHEN "0011" => leds <= "0000110";
16 WHEN "0100" => leds <= "1001100";
17 WHEN "0101" => leds <= "0100100";
18 WHEN "0110" => leds <= "0100000";
19 WHEN "0111" => leds <= "0001111";
20 WHEN "1000" => leds <= "0000000";
21 WHEN "1001" => leds <= "0001100";
22 WHEN "1010" => leds <= "0100000";
23 WHEN "1011" => leds <= "0100100";
24 WHEN "1100" => leds <= "1001100";
25 WHEN "1101" => leds <= "0000110";
26 WHEN "1110" => leds <= "0010010";
27 WHEN "1111" => leds <= "1001111";
28 WHEN OTHERS => leds <= "-----";
29 END CASE ;
30 END PROCESS ;
31 END Behavior;
```

B. During the lab

a. Design A

(1)Part I

In this part we assign the input and output to the pin planner then we compiled again. Finally, we upload the design to the board, and we do the test on the board.

The screenshot shows the Pin Planner application window. The title bar indicates the path: C:/altera/16.0/Dillman/lab9a1/designA1 - designA1. The menu bar includes File, Edit, View, Processing, Tools, Window, and Help. A search bar is present in the top right corner.

On the left side, there are several panels:

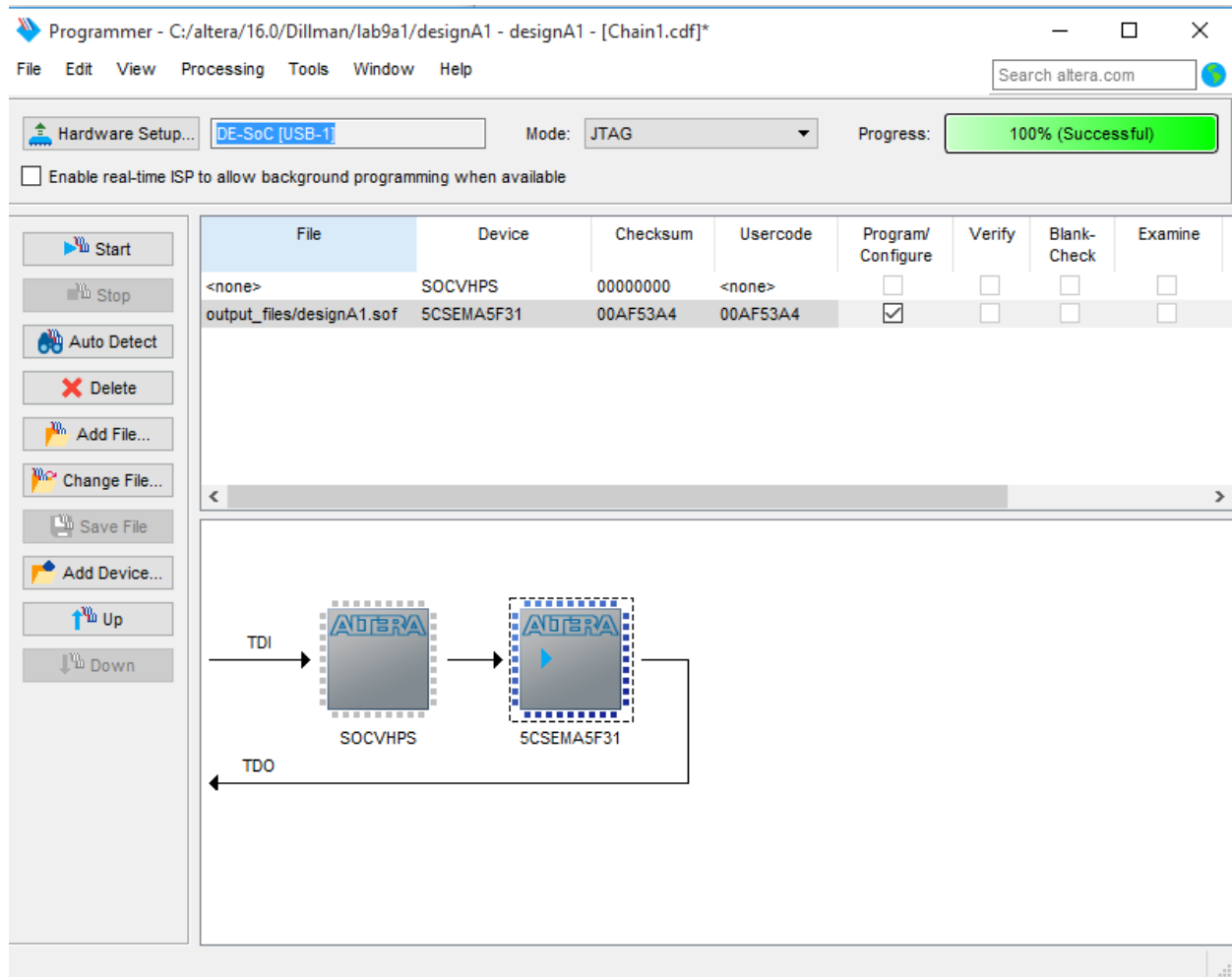
- Report:** Shows "Report not available".
- Groups:** Includes "Report".
- Tasks:** Lists "Early Pin Planning" (with sub-tasks "Early Pin Planning..." and "Run IO Assignment Analysis") and "Run IO Assignment Analysis".
- Pin Legend:** Defines symbols for "User IO" (white circle), "User assigned IO" (red circle), and "Fitter assigned IO" (green circle).

The main area displays a "Top View - Wire Bond" of a "Cyclone V - 5CSEMA5F31C8" chip. The chip's pin grid is visible, with various pins highlighted in different colors (red, green, blue) to indicate their assignment status.

At the bottom, a table lists the assigned pins:

Node Name	Direction	Location	IO Bank	VREF Group	Fitter Location	IO Standard
in Clk	Input	PIN_AA14	3B	B3B_N0	PIN_AA14	2.5 V
in D	Input	PIN_AB12	3A	B3A_N0	PIN_AB12	2.5 V
out Q	Output	PIN_V16	4A	B4A_N0	PIN_V16	2.5 V
<<new node>>						

The bottom status bar shows "100%" zoom and a timer at "00:00:50".



(2)Part II

In this part we repeat the same procedure in the previous part.

Pin Planner - C:/altera/16.0/Dillman/lab9a2/Lab9a2 - Lab9a2

File Edit View Processing Tools Window Help

Search altera.com

Report

Report not available

Groups Report

Tasks

- Early Pin Planning
 - Early Pin Planning...
 - Run I/O Assignment Analysis

Pin Legend

Symbol	Pin Type
○	User I/O
●	User assigned I/O
●	Fitter assigned I/O

Top View - Wire Bond
Cyclone V - 5CSEMA5F31C8

Named: * Edit: Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
Clock	Input	PIN_AA14	3B	B3B_N0	PIN_AA14	2.5 V
D	Input	PIN_AB12	3A	B3A_N0	PIN_AB12	2.5 V
Q	Output	PIN_V16	4A	B4A_N0	PIN_V16	2.5 V
s	Unknown					2.5 V (default)
<<new node>>						

Programmer - C:/altera/16.0/Dillman/lab9a2/Lab9a2 - Lab9a2 - [Chain1.cdf]*

File Edit View Processing Tools Window Help

Search altera.com

Hardware Setup... DE-SoC [USB-1] Mode: JTAG Progress: 100% (Successful)

☐ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine
<none>	SOCVHPS	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
output_files/Lab9a2.sof	5CSEMA5F31	00AF53A4	00AF53A4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Start Stop Auto Detect Delete Add File... Change File... Save File Add Device... Up Down

Diagram showing JTAG connection between SOCVHPS and 5CSEMA5F31 devices.

```

graph LR
    TDI --> SOCVHPS
    SOCVHPS --> 5CSEMA5F31
    5CSEMA5F31 --> TDO
  
```

b. Design B

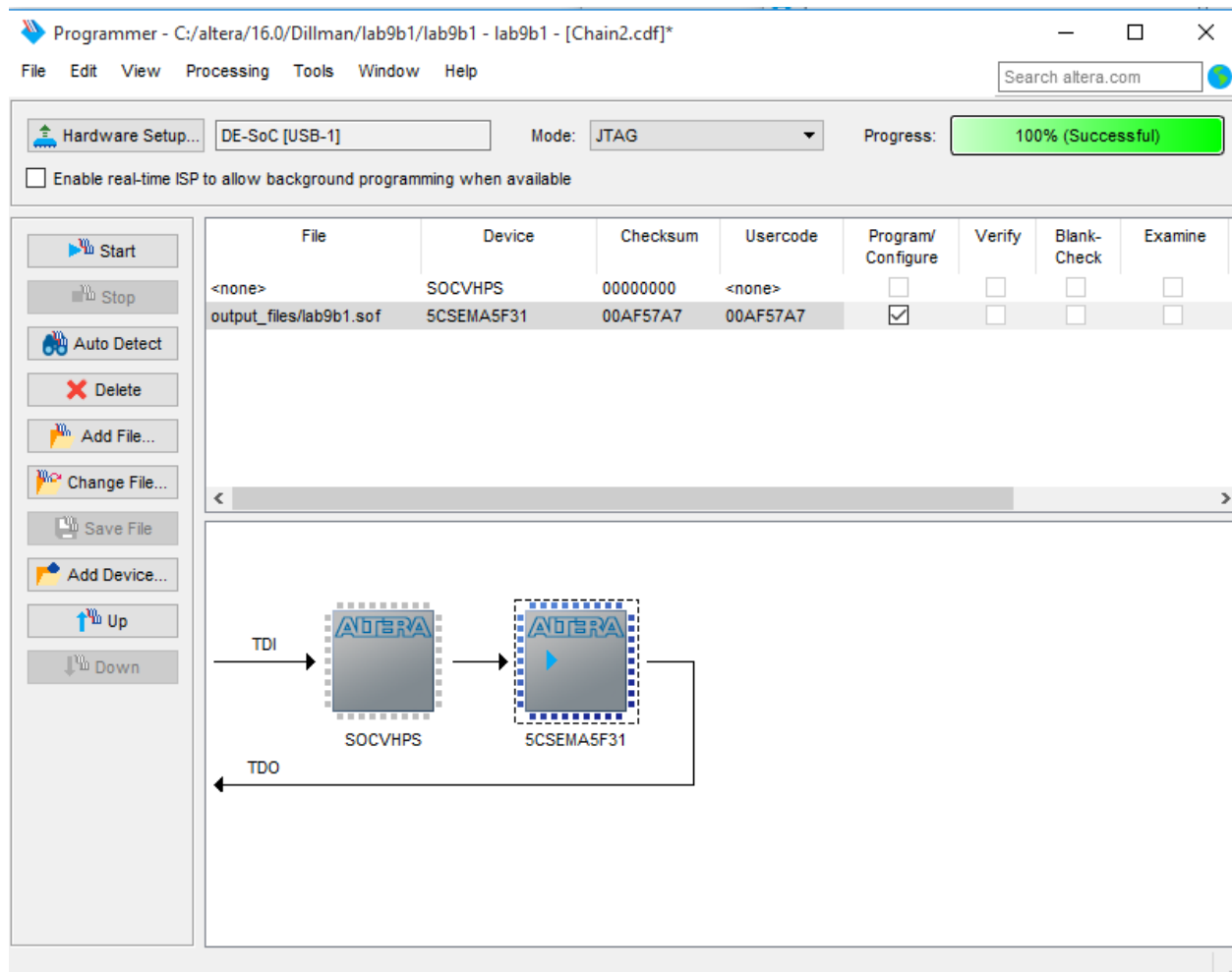
(1)Part I

In this part, we assign input and output of JK flip-flop to the pin. Then we compiled the file again.

Finally, we upload the design to the upload to DE1 board and run the test.

The screenshot shows the Pin Planner interface for a Cyclone V - 5CSEMA5F31C8 device. The top view of the chip is displayed, showing the pin grid and the assigned pins. The pin assignments are listed in the table below:

Node Name	Direction	Location	IO Bank	VREF Group	Fitter Location	IO Standard
in Clk	Input	PIN_AA14	3B	B3B_N0	PIN_AA14	2.5 V
in J	Input	PIN_AB12	3A	B3A_N0	PIN_AB12	2.5 V
in K	Input	PIN_AC12	3A	B3A_N0	PIN_AC12	2.5 V
out Q	Output	PIN_V16	4A	B4A_N0	PIN_V16	2.5 V
<<new node>>						



(2)Part II

We do the same procedure in the previous part.

Pin Planner - C:/altera/16.0/Dillman/lab9b2/lab9b2 - lab9b2

File Edit View Processing Tools Window Help

Search altera.com

Report

Report not available

Groups Report

Tasks

- Early Pin Planning
 - Early Pin Planning...
 - Run I/O Assignment Analysis

Pin Legend

Symbol	Pin Type
	User I/O
	User assigned I/O
	Fitter assigned I/O

Top View - Wire Bond
Cyclone V - 5CSEMA5F31C8

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
CLOCK	Input	PIN_AA14	3B	B3B_N0	PIN_AA14	2.5 V
J	Input	PIN_AB12	3A	B3A_N0	PIN_AB12	2.5 V
K	Input	PIN_AC12	3A	B3A_N0	PIN_AC12	2.5 V
Q	Output	PIN_V16	4A	B4A_N0	PIN_V16	2.5 V

<<new node>>

Programmer - C:/altera/16.0/Dillman/lab9b2/lab9b2 - lab9b2 - [Chain1.cdf]*

File Edit View Processing Tools Window Help

Search altera.com

Hardware Setup... DE-SoC [USB-1] Mode: JTAG Progress: 100% (Successful)

☐ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine
<none>	SOCVHPS	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
output_files/lab9b2.sof	5CSEMA5F31	00AF57A7	00AF57A7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Start Stop Auto Detect Delete Add File... Change File... Save File Add Device... Up Down

```
graph LR; SOCVHPS -- TDI --> 5CSEMA5F31; 5CSEMA5F31 -- TDO --> SOCVHPS;
```


c. Design C

For the design C, we assigned the input and output then compiled the file again. After we upload the design to the board, we start test.

Pin Planner - C:/altera/16.0/Dillman/lab9c2/lab9c2 - lab9c2

File Edit View Processing Tools Window Help

Search altera.com

Report

Report not available

Groups Report

Tasks

- Early Pin Planning
 - Early Pin Planning...
 - Run I/O Assignment Analysis

Pin Legend

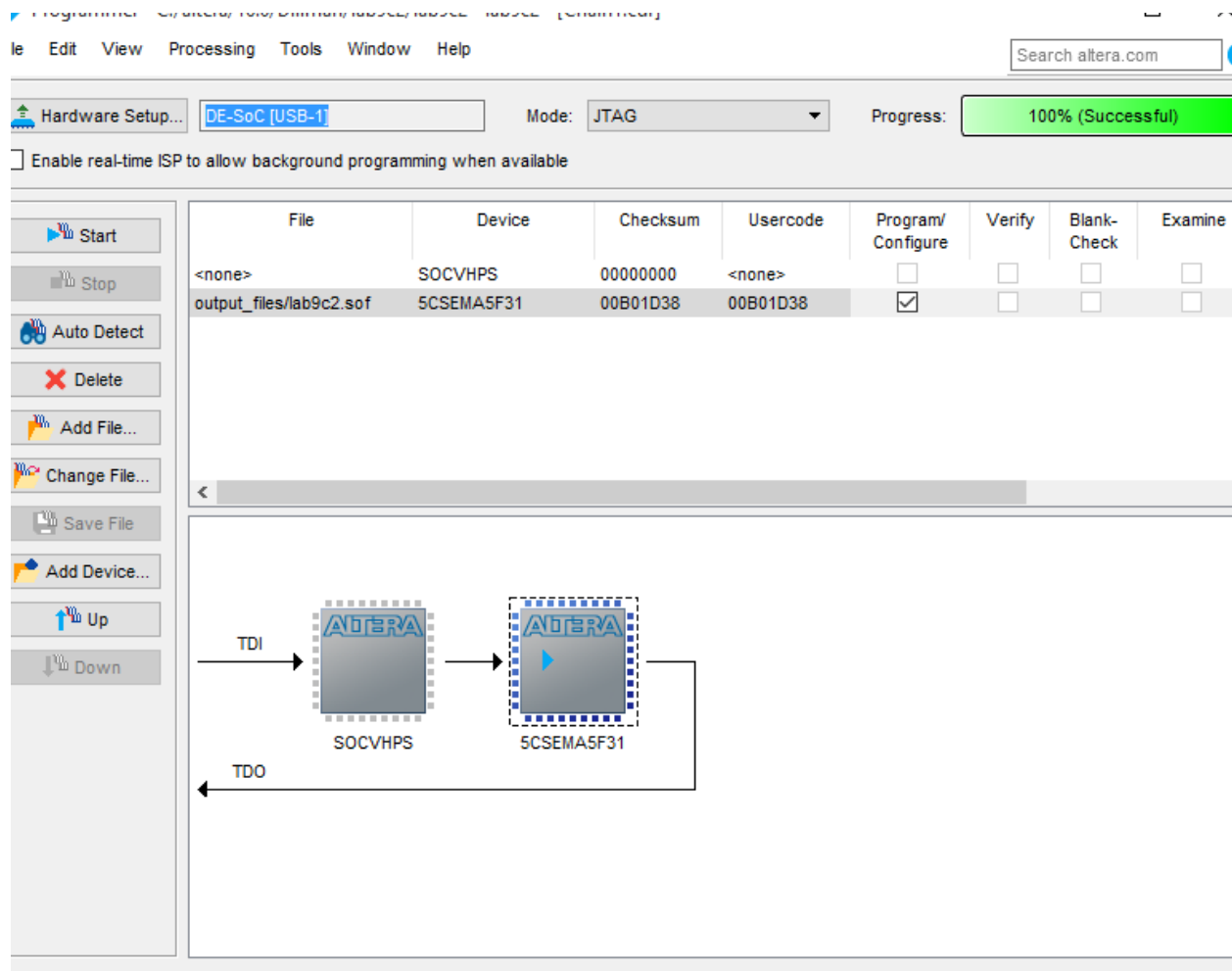
Symbol	Pin Type
○	User I/O
●	User assigned I/O
●	Fitter assigned I/O

Top View - Wire Bond
Cyclone V - 5CSEMA5F31C8

Named: * Edit: Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
in Clear	Input	PIN_AA14	3B	B3B_N0	PIN_AA14	2.5 V
in Clock	Input	PIN_AF14	3B	B3B_N0	PIN_AF14	2.5 V
in En	Input	PIN_AB12	3A	B3A_N0	PIN_AB12	2.5 V
out[0]	Output	PIN_AE26	5A	B5A_N0	PIN_AE26	2.5 V
out[1]	Output	PIN_AE27	5A	B5A_N0	PIN_AE27	2.5 V
out[2]	Output	PIN_AE28	5A	B5A_N0	PIN_AE28	2.5 V
out[3]	Output	PIN_AG27	5A	B5A_N0	PIN_AG27	2.5 V
out[4]	Output	PIN_AF28	5A	B5A_N0	PIN_AF28	2.5 V
out[5]	Output	PIN_AG28	5A	B5A_N0	PIN_AG28	2.5 V

100% 00:00:52



Result

All the designs are tested. The result is same as we expected.

Conclusion

- Connect 1 Hz clock signal with the four-bit counter. And use the seven-segment LED to display the counting numbers, 0 to 9.

<u>Pre-Lab (30%)</u>	<u>Score</u>	<u>TA initial</u>
30% Designs	A1	JP

<u>Report (70%)</u>		
10% Introduction	B1	JP
10% Procedures	B1	JP
20% Results	CA	JP
30% Conclusions	C2	JP

<u>Lab Grade (100%)</u>	
<u>Bonus Points:</u>	