INSTITUTE OF ENGINEERING AND TECHNOLOGY, DAVV
INDORE

ELECTRONICS & INSTUMENTATION ENGINEERING

CIRCUIT DESIGN USING HDL(EIR7E1)

LAB VIVA

Session: 2020-21

SUBMITTED TO:

Dr. Vaibhav Neema

Roll No: 17E7001

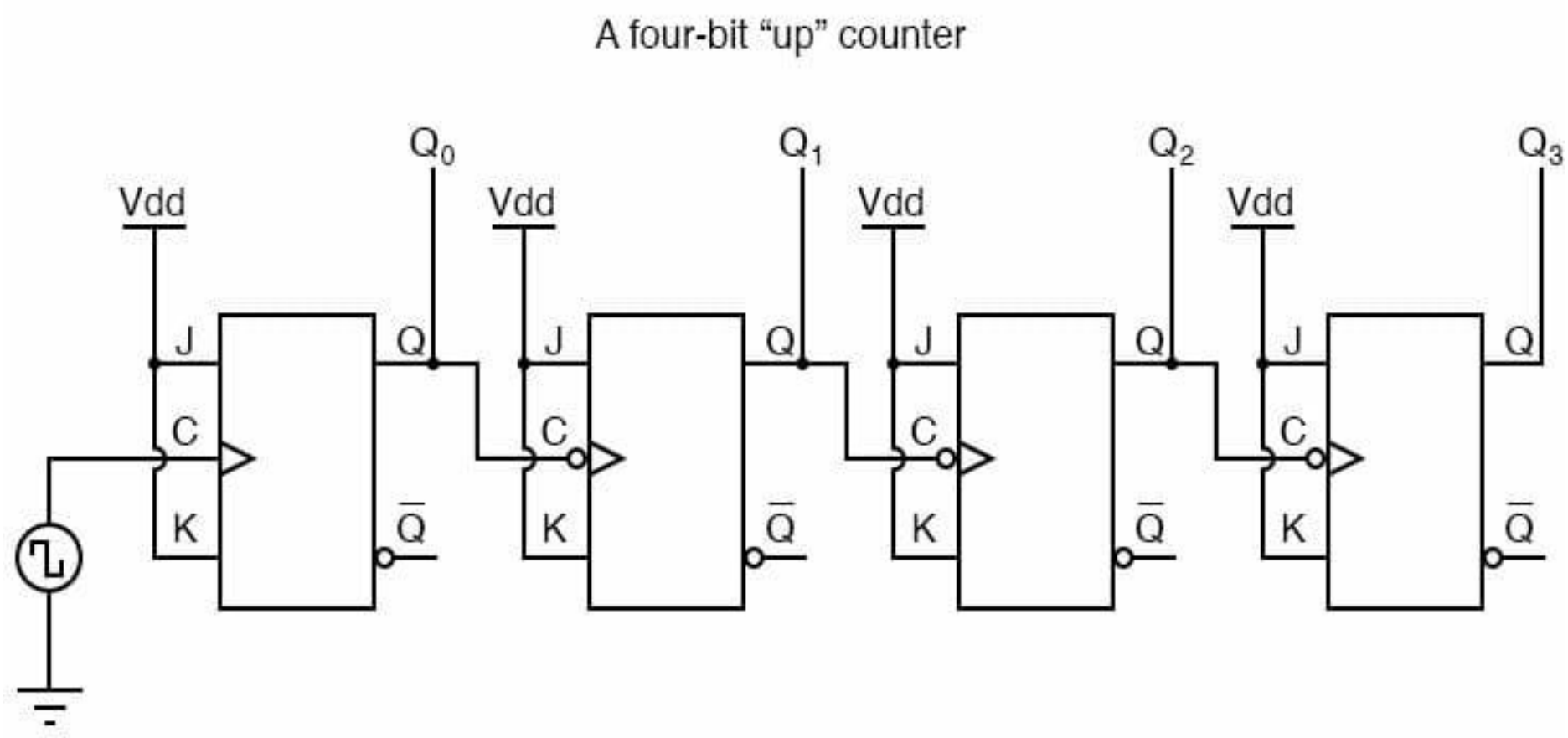
SUBMITTED BY:

Aastha Arora

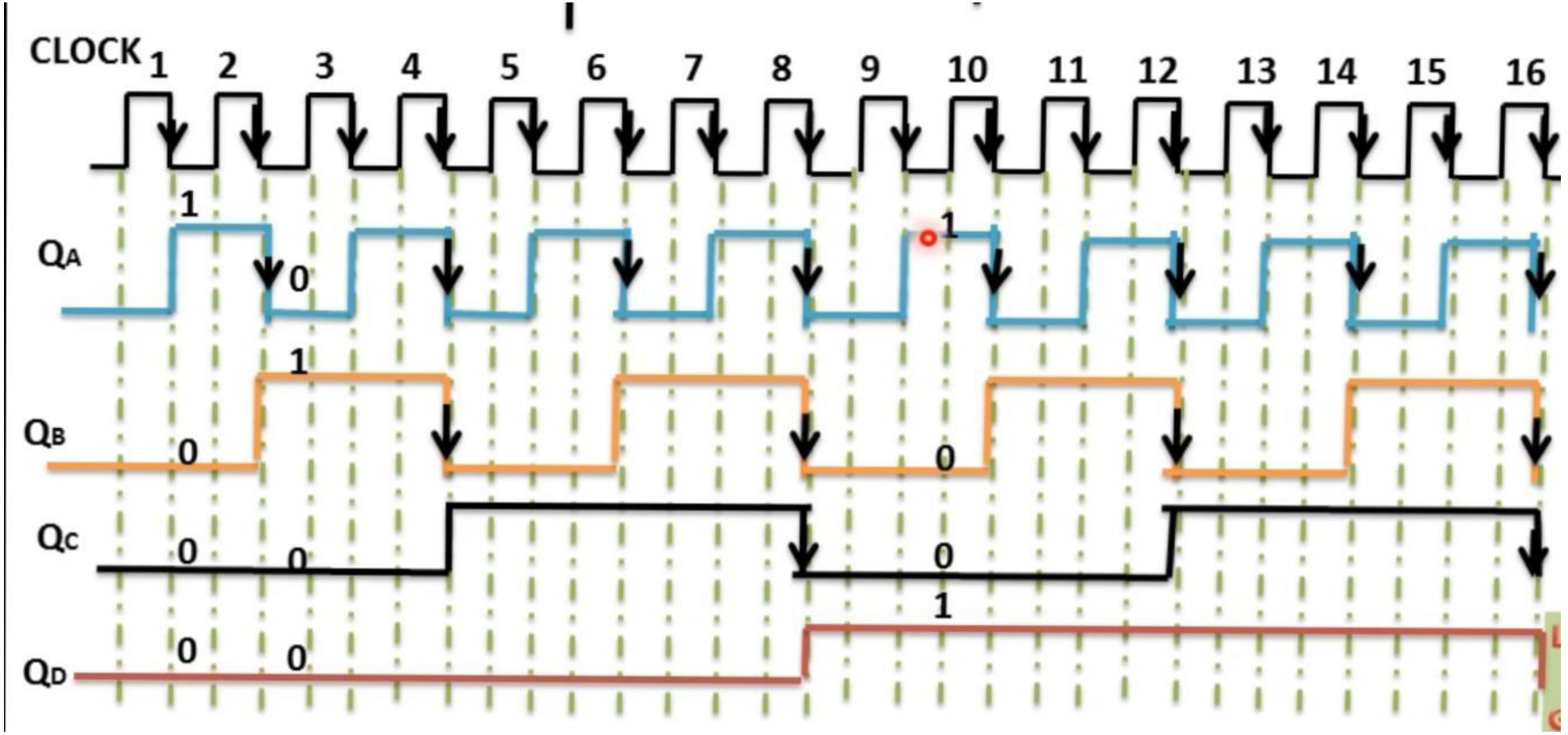
Email: aasthaarora28@gmail.com

Phone: 9754656668

4 Bit asynchronous counter working:



Waveform of asynchronous counter:



Truth table:

clk	Q_D	Q_C	Q_B	Q_A	decimal
Initial	0	0	0	0	0
1st falling edge	0	0	0	1	1
2nd falling edge	0	0	1	0	2
9th falling edge	1	0	0	1	9
14th falling edge	1	1	1	0	14

Driver code:

- library IEEE; **use IEEE.STD_LOGIC_1164.ALL**;
- **use IEEE.STD_LOGIC_UNSIGNED.ALL**;
- -- VHDL project: VHDL code for up counter entity UP_COUNTER **is** Port (clk: **in** std_logic; -- clock input reset: **in** std_logic;
- -- reset input counter: **out** std_logic_vector(**3** downto **0**)
- -- output 4-bit counter);
- **end UP_COUNTER**;
- architecture Behavioral **of** UP_COUNTER **is** signal counter_up: std_logic_vector(**3** downto **0**);
- **begin**
- –- up counter process(clk)
- **begin if**(rising_edge(clk)) **then**
- **if**(reset='1') **then** counteBehavior “0000";
- **else** counter_up <= counter_up + ‘1’;
- **end if**;
- **end if**;
- **end process**;
- counter <= counter_up;
- **end Behavioral**;

Testbench code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
 entity tb_counters is end tb_counters;
 architecture Behavioral of tb_counters is
component UP_COUNTER Port ( clk: in std_logic; -- clock input
                            reset: in std_logic;-- reset input
                           counter: out std_logic_vector(3 downto 0) -- output 4-bit counter );
 end component;
signal reset,clk: std_logic;
signal counter:std_logic_vector(3 downto 0); begin dut: UP_COUNTER port map (clk => clk, reset=>reset, counter => counter);
-- Clock process definitions clock_process :process begin clk <= '0'; wait for 10 ns; clk <= '1';
 wait for 10 ns; end process;
-- Stimulus process stim_proc: process begin -- hold reset state for 100 ns. reset <= '1'; wait for 20 ns;
reset <= '0';
wait;
end process;
end Behavioral;
```