

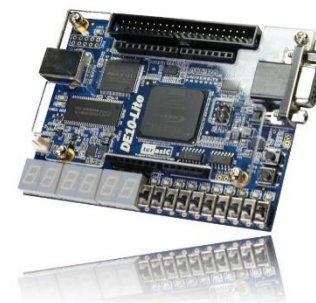
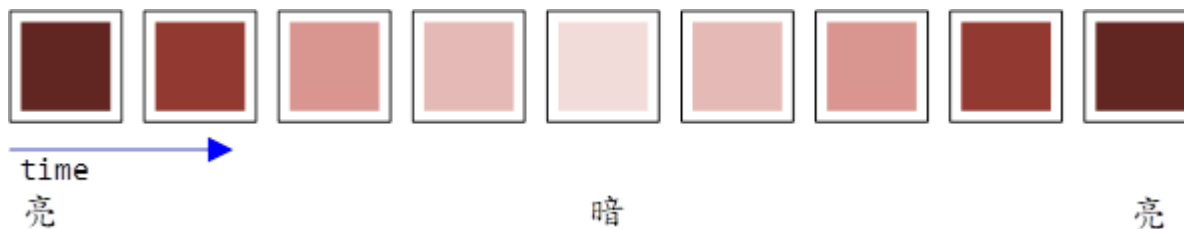
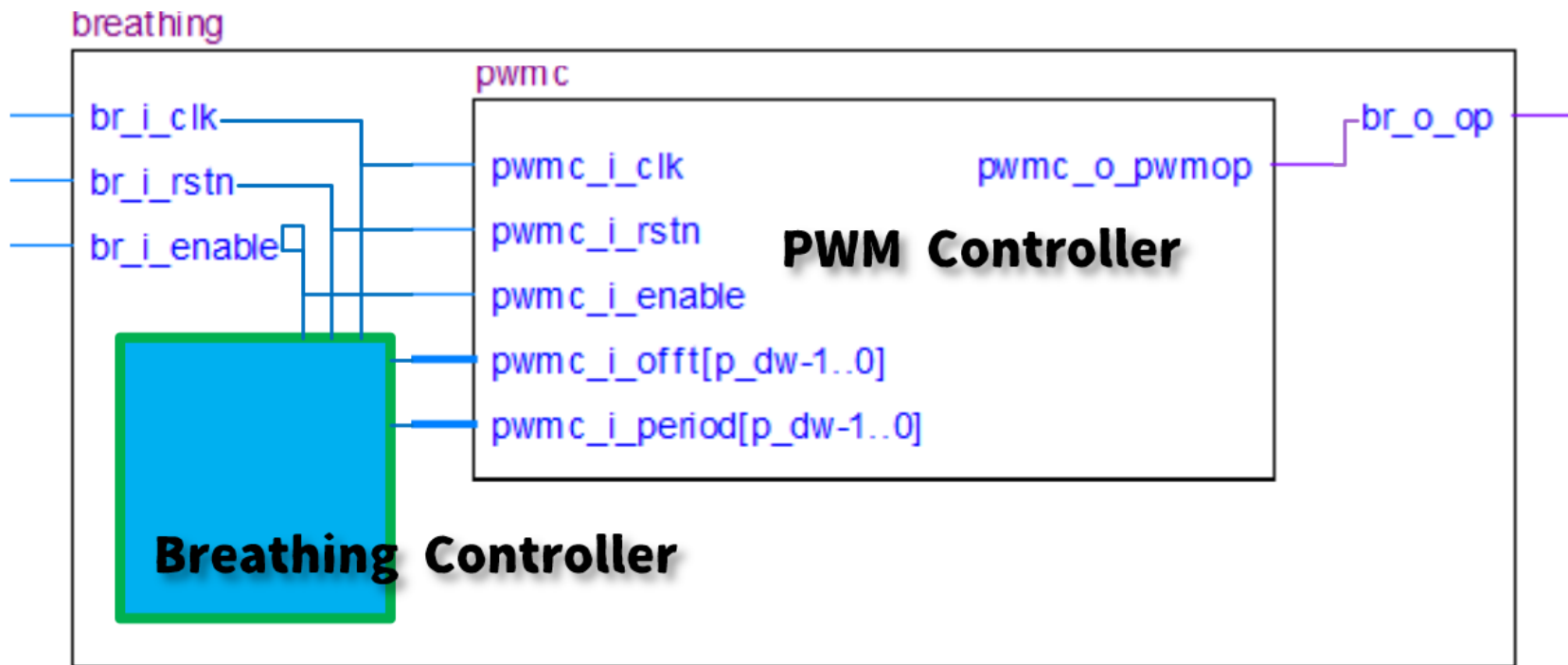
GRAFCET 電路設計-呼吸燈

Outline

- 呼吸燈
- PWM控制器
- 呼吸燈控制器
- 隨堂練習



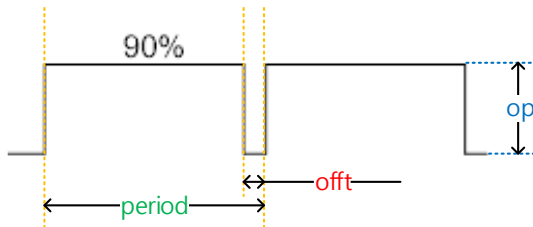
呼吸燈



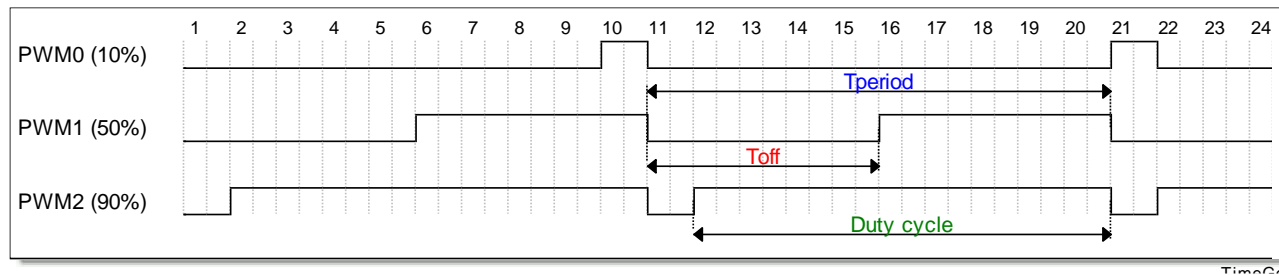
PWM控制器

• Interface

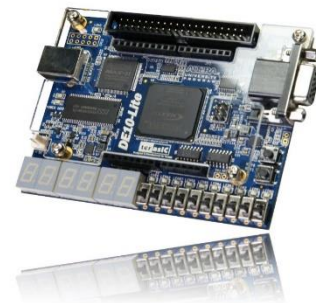
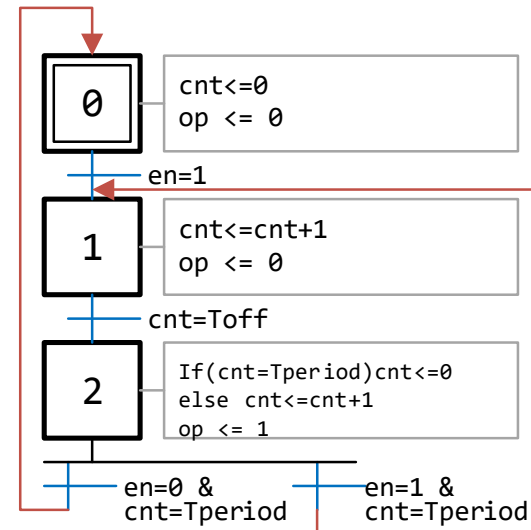
Sig.	Dir.	Bit	Desc.
clk	in	1	時脈輸入
rstn	in	1	重置
en	in	1	致能
offt	in	8 (Default)	如圖
period	in	8 (Default)	如圖
op	out	1	如圖



• Waveform



• Grafect離散事件



PWM控制器

- VHDL Source Code
 - pwmc.vhdl

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity pwmc is
    generic(
        p_dw : integer := 8
    );
    port(
        pwmc_i_clk      : in  std_logic;
        pwmc_i_rstn     : in  std_logic;
        pwmc_i_enable   : in  std_logic;
        pwmc_i_offt     : in  std_logic_vector((p_dw-1) downto 0);
        pwmc_i_period   : in  std_logic_vector((p_dw-1) downto 0);
        pwmc_o_pwmop    : out std_logic
    );
end entity pwmc;

architecture rtl of pwmc is

    type grafect is (X0, X1, X2);
    signal w_next_state : grafect;
    signal r_curr_state : grafect;

    signal w_cntr : std_logic_vector((p_dw-1) downto 0);
    signal r_cntr : std_logic_vector((p_dw-1) downto 0);

    signal w_pwmbuf : std_logic;
    signal r_pwmbuf : std_logic;

begin

    gstate: process(pwmc_i_clk, pwmc_i_rstn)
    begin
        if pwmc_i_rstn = '0' then
            r_curr_state <= X0;
        elsif rising_edge(pwmc_i_clk) then
            r_curr_state <= w_next_state;
        end if;
    end process;
```

```
control_path: process(r_curr_state, pwmc_i_enable, r_cntr, pwmc_i_offt, pwmc_i_period)
begin
    case r_curr_state is
        when X0 =>
            if pwmc_i_enable = '1' then
                w_next_state <= X1;
            else
                w_next_state <= X0;
            end if;
        when X1 =>
            if r_cntr = pwmc_i_offt then
                w_next_state <= X2;
            else
                w_next_state <= X1;
            end if;
        when X2 =>
            if r_cntr = pwmc_i_period then
                if pwmc_i_enable = '1' then
                    w_next_state <= X1;
                else
                    w_next_state <= X0;
                end if;
            else
                w_next_state <= X2;
            end if;
        end case;
    end process;

    data_path: process(r_curr_state, r_cntr, pwmc_i_offt, pwmc_i_period)
    begin
        case r_curr_state is
            when X0 =>
                w_cntr <= (others => '0');
                w_pwmbuf <= '0';
            when X1 =>
                w_cntr <= r_cntr + '1';
                w_pwmbuf <= '0';
            when X2 =>
                if r_cntr = pwmc_i_period then
                    w_cntr <= (others => '0');
                else
                    w_cntr <= r_cntr + '1';
                end if;
                w_pwmbuf <= '1';
            end case;
        end process;

    outbuf_path: process(pwmc_i_clk, pwmc_i_rstn)
    begin
        if pwmc_i_rstn = '0' then
            r_cntr <= (others => '0');
            r_pwmbuf <= '0';
        elsif rising_edge(pwmc_i_clk) then
            r_cntr <= w_cntr;
            r_pwmbuf <= w_pwmbuf;
        end if;
    end process;

    pwmc_o_pwmop <= r_pwmbuf;

end rtl;
```

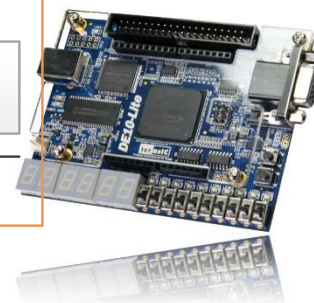
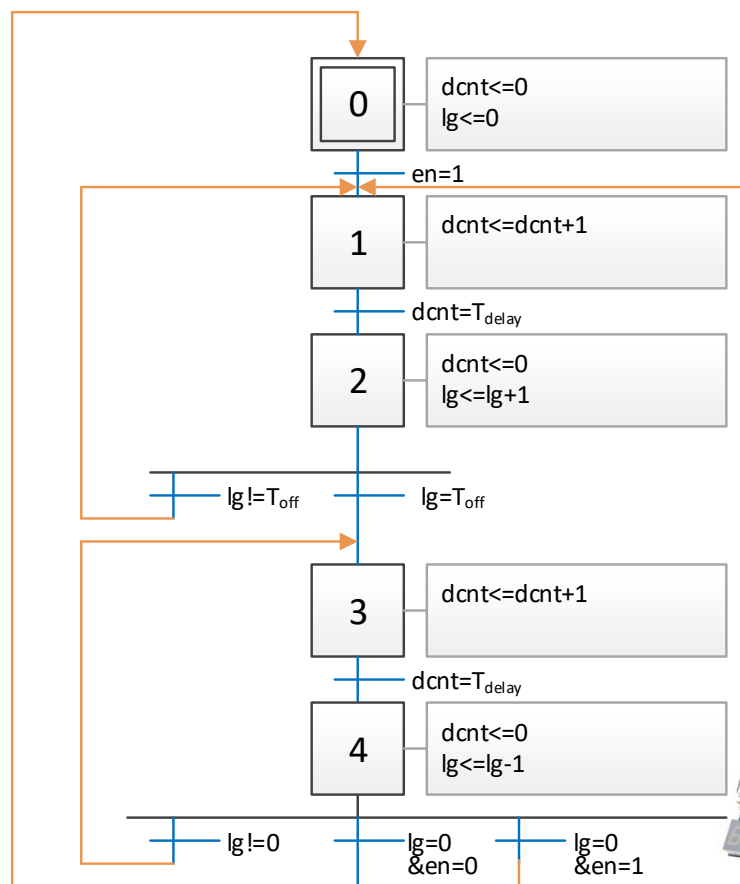


呼吸燈控制器

- Interface

Sig.	Dir.	Bit	Desc.
clk	in	1	時脈輸入
rstn	in	1	重置
en	in	1	致能
op	out	1	輸出

- Grafect 離散事件



呼吸燈控制器

- VHDL Source Code
 - breathing.vhdl

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity breathing is
generic(
    lg_dw : integer := 8;
    dcnt_dw : integer := 16
);
port(
    br_i_clk : in std_logic;
    br_i_rstn : in std_logic;
    br_i_enable : in std_logic;
    br_o_op : out std_logic
);
end entity breathing;

architecture rtl of breathing is
    component pwmc is
        generic(
            p_dw : integer := 8
        );
        port(
            pwmc_i_clk : in std_logic;
            pwmc_i_rstn : in std_logic;
            pwmc_i_enable : in std_logic;
            pwmc_i_offt : in std_logic_vector((p_dw-1) downto 0);
            pwmc_i_period : in std_logic_vector((p_dw-1) downto 0);
            pwmc_o_pwmop : out std_logic
        );
    end component pwmc;

    type grafect is (X0, X1, X2, X3, X4);
    signal w_next_state : grafect;
    signal r_curr_state : grafect;

    signal w_dcnt : std_logic_vector((dcnt_dw-1) downto 0);
    signal r_dcnt : std_logic_vector((dcnt_dw-1) downto 0);
    signal w_lg : std_logic_vector((lg_dw-1) downto 0);
    signal r_lg : std_logic_vector((lg_dw-1) downto 0);

begin
    gstate: process(br_i_clk, br_i_rstn)
    begin
        if br_i_rstn = '0' then
            r_curr_state <= X0;
        elsif rising_edge(br_i_clk) then
            r_curr_state <= w_next_state;
        end if;
    end process;
```

```
control_path: process(r_curr_state, br_i_enable, r_dcnt, r_lg)
begin
    case r_curr_state is
        when X0 =>
            if br_i_enable = '1' then
                w_next_state <= X1;
            else
                w_next_state <= X0;
            end if;
        when X1 =>
            if r_dcnt = x"FE" then -- (x"FFFE") set x"FE" for simulation
                w_next_state <= X2;
            else
                w_next_state <= X1;
            end if;
        when X2 =>
            if r_lg = x"F" then -- (x"F0") set x"F" for simulation
                w_next_state <= X3;
            else
                w_next_state <= X1;
            end if;
        when X3 =>
            if r_dcnt = x"FE" then -- (x"FFFE") set x"FE" for simulation
                w_next_state <= X4;
            else
                w_next_state <= X3;
            end if;
        when X4 =>
            if r_lg = x"01" and br_i_enable = '1' then
                w_next_state <= X1;
            elsif r_lg = x"01" and br_i_enable = '0' then
                w_next_state <= X0;
            else
                w_next_state <= X3;
            end if;
    end case;
end process;

data_path: process(r_curr_state, r_dcnt, r_lg)
begin
    case r_curr_state is
        when X0 =>
            w_dcnt <= (others => '0');
            w_lg <= (others => '0');
        when X1 =>
            w_dcnt <= r_dcnt + '1';
            w_lg <= r_lg;
        when X2 =>
            w_dcnt <= (others => '0');
            w_lg <= r_lg + '1';
        when X3 =>
            w_dcnt <= r_dcnt + '1';
            w_lg <= r_lg;
        when X4 =>
            w_dcnt <= (others => '0');
            w_lg <= r_lg - '1';
    end case;
end process;

outbuf_path: process(br_i_clk, br_i_rstn)
begin
    if br_i_rstn = '0' then
        r_dcnt <= (others => '0');
        r_lg <= (others => '0');
    elsif rising_edge(br_i_clk) then
        r_dcnt <= w_dcnt;
        r_lg <= w_lg;
    end if;
end process;

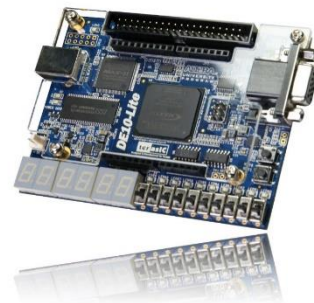
u0: pwmc
generic map(
    p_dw => lg_dw
)
port map(
    pwmc_i_clk => br_i_clk,
    pwmc_i_rstn => br_i_rstn,
    pwmc_i_enable => br_i_enable,
    pwmc_i_offt => r_lg,
    pwmc_i_period => x"FF",
    pwmc_o_pwmop => br_o_op
);

end rtl;
```



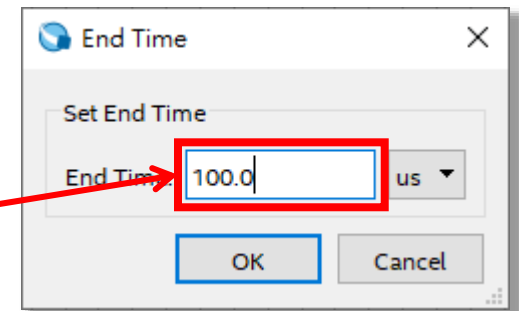
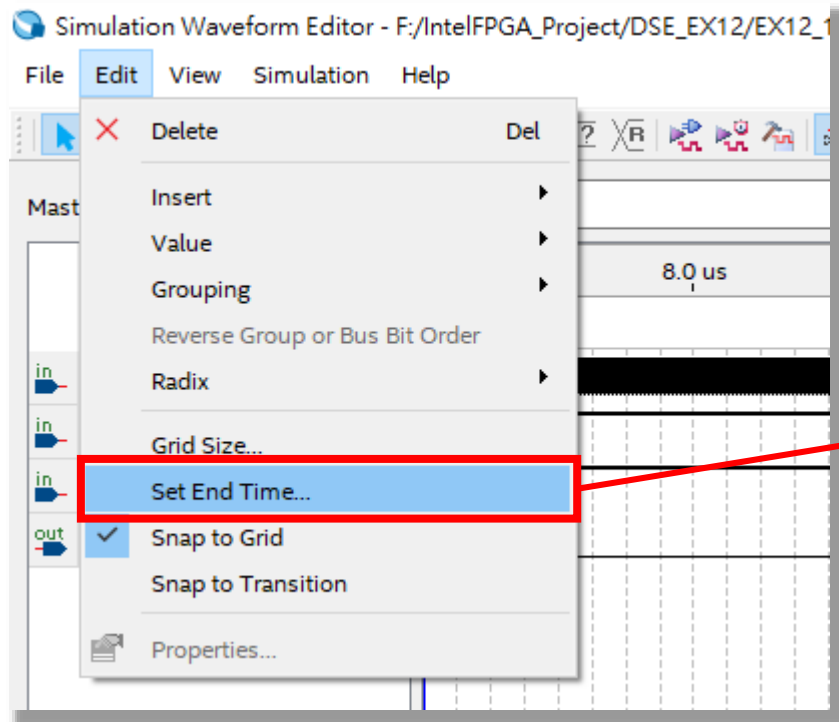
隨堂練習

- 請使用VHDL完成上述PWM控制電路與呼吸燈控制電路，並完成紀錄，包括 **GRAFCET離散事件建模**、**VHDL Source Code**、**模擬波形圖**，並將電路燒至實驗版中進行驗證。
- 本次實驗完成後請，將專案與報告壓縮上傳EE-Class。
- Lecture12_組別XX.ZIP



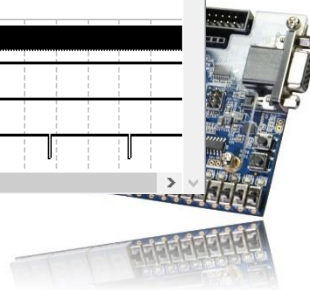
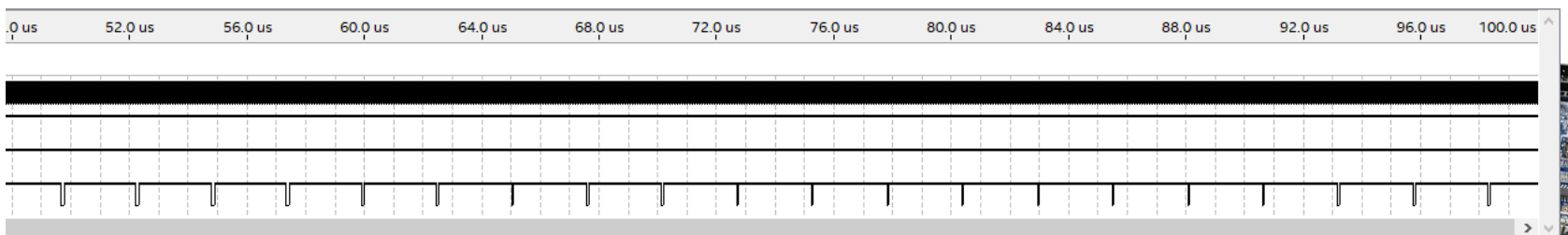
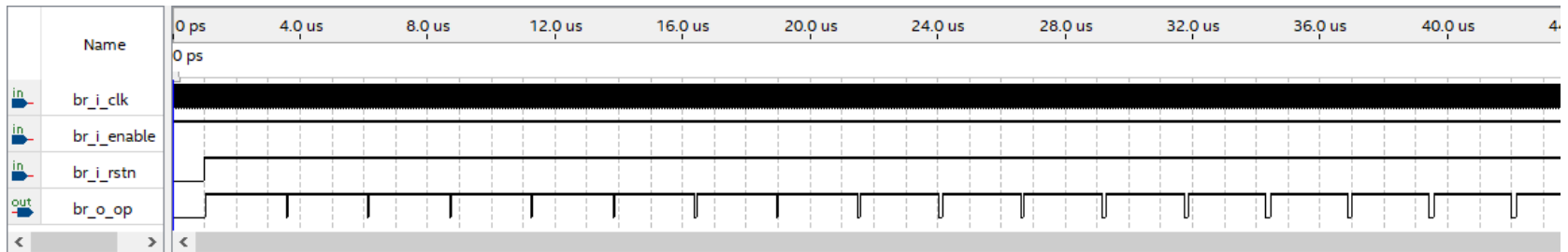
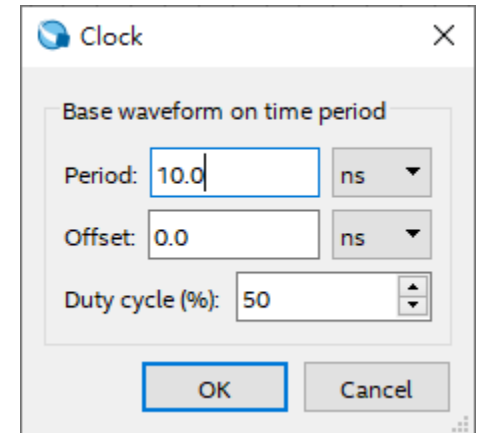
隨堂練習

- 模擬波形圖
 - 設定顯示時間

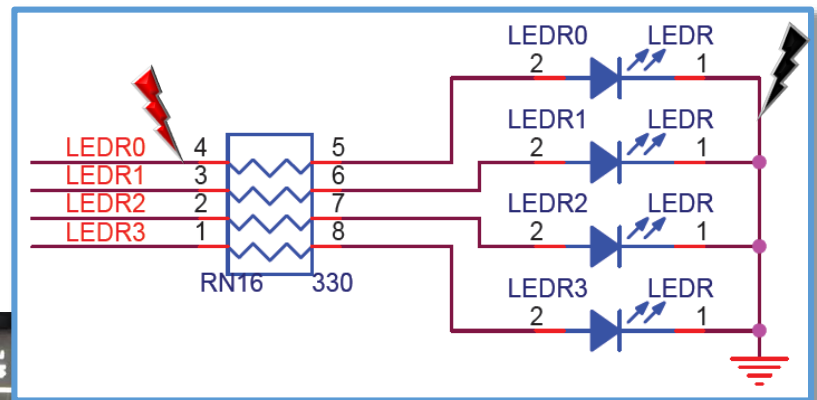


隨堂練習

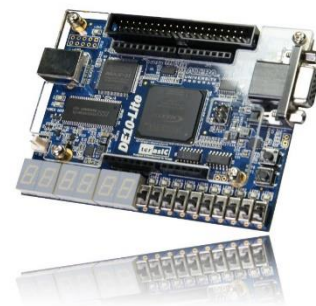
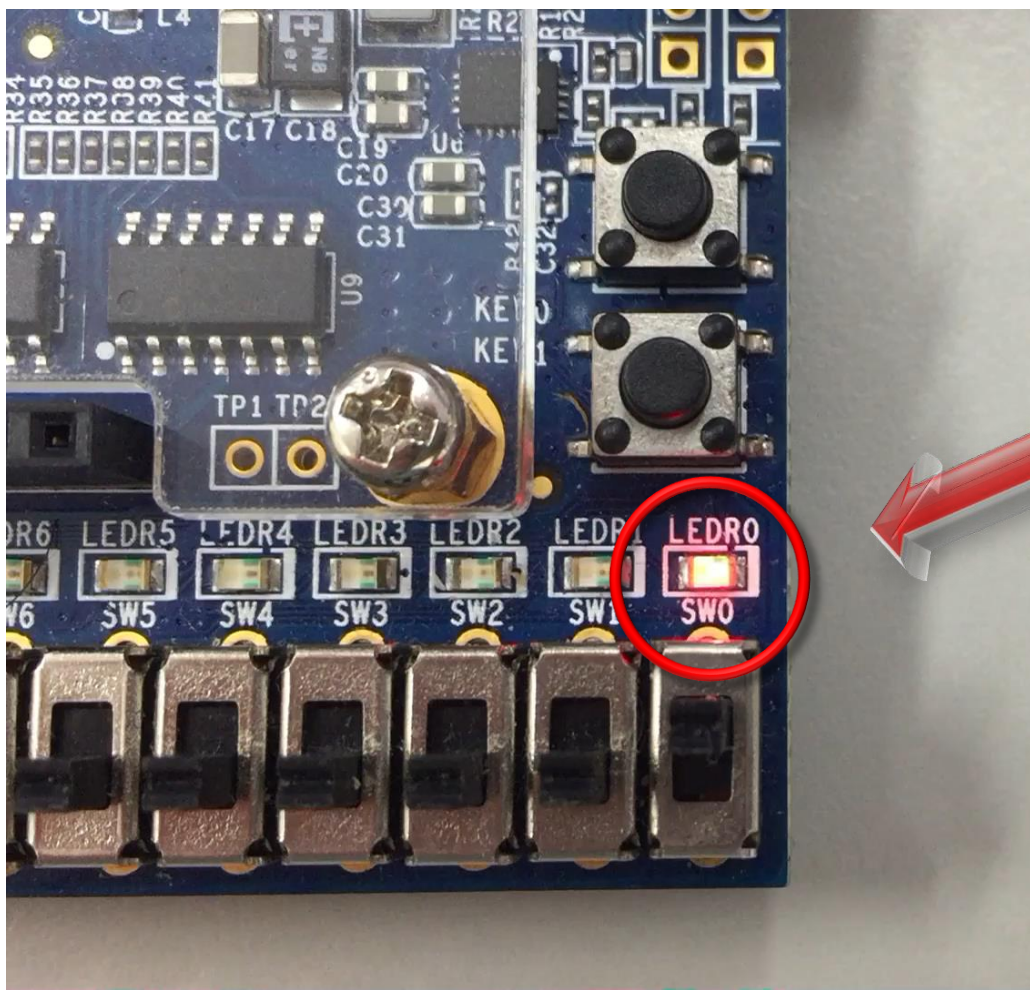
- 模擬波形圖
 - 設定Clock



隨堂練習

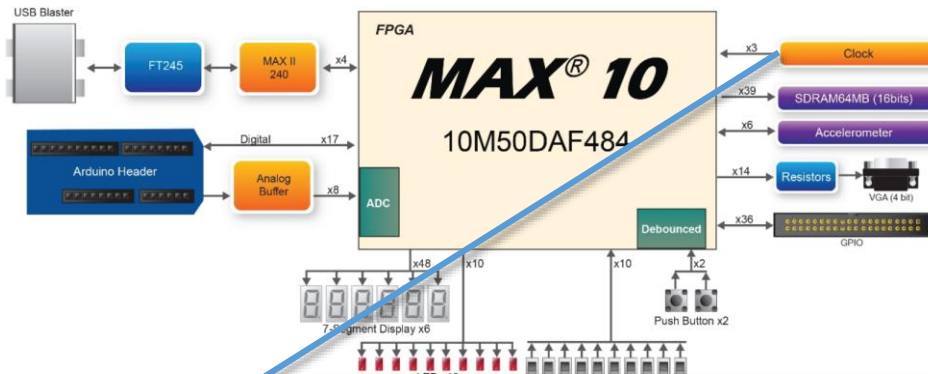


隨堂練習



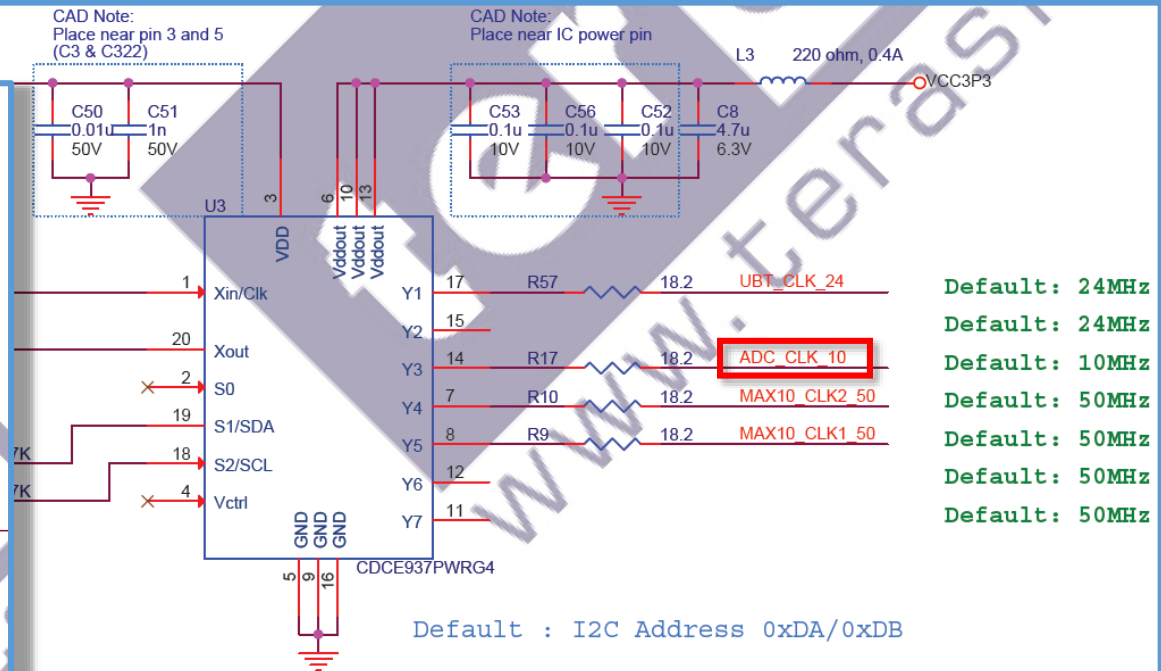
Clocks Circuit of DE10-Lite

(br_i_clk)



MAX10 Clock

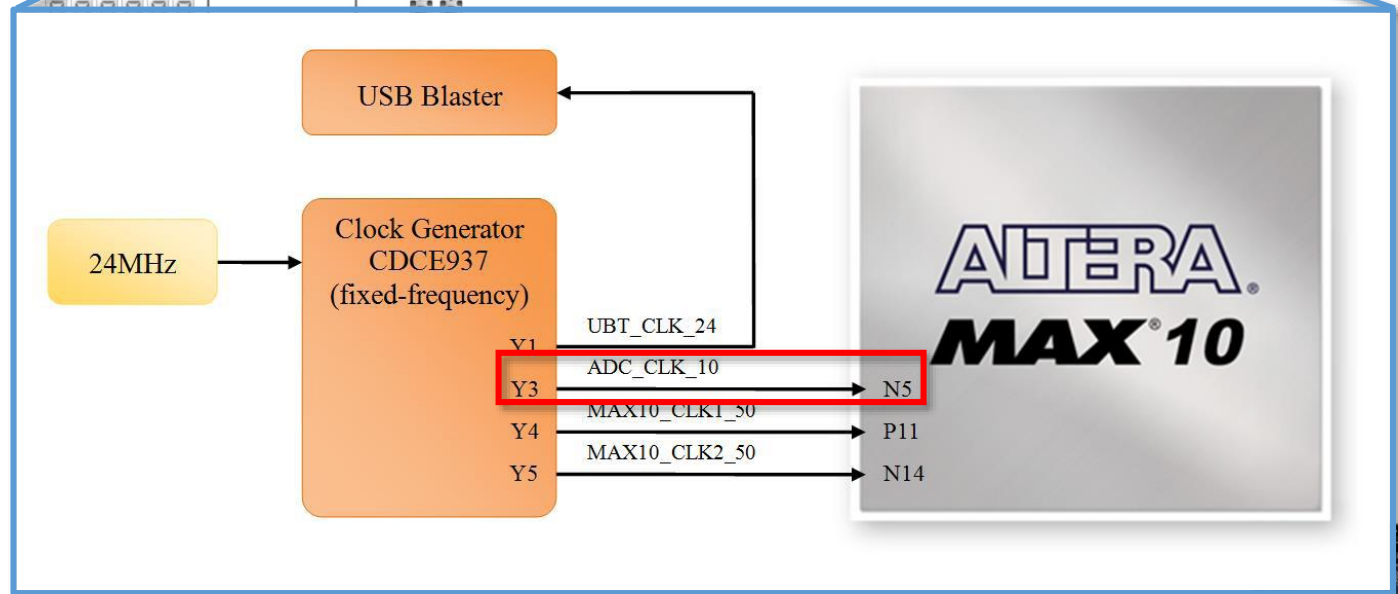
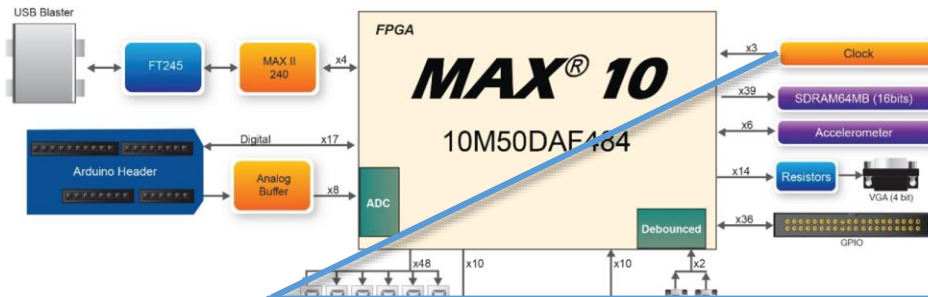
MAX 10 CLOCK			
BANK-2 VCCIO = 3.3V			
DIFFIO_RX_L28N/CLK0N	DIFFIO_RX_L38N/DCLK0	P3	
DIFFIO_RX_L28P/CLK0P	DIFFIO_RX_L38P/DCLK1	R3	
DIFFIO_RX_L36N/CLK1N	DIFFIO_RX_L59N/PLL_L_CLKOUTN	T5	
DIFFIO_RX_L36P/CLK1P	DIFFIO_RX_L59P/PLL_L_CLKOUTP	T6	
BANK-3 VCCIO = 3.3V			
DIFFIO_TX_RX_B18N/CLK6N	DIFFIO_TX_RX_B18P/CLK6P		
DIFFIO_TX_RX_B20N/CLK7N	DIFFIO_TX_RX_B20P/CLK7P		
BANK-4 VCCIO = 3.3V			
DIFFIO_TX_RX_B57N/PLL_B_CLKOUTN	DIFFIO_TX_RX_B57P/PLL_B_CLKOUTP	V17	
BANK-6 VCCIO = 3.3V			
DIFFIO_RX_R38N/CLK2N	DIFFIO_RX_R50N/DCLK2/DOS2R	L15	DRAM_CLK
DIFFIO_RX_R38P/CLK2P	DIFFIO_RX_R50P/DCLK3/DOS2R	L14	
DIFFIO_RX_R40N/CLK3N	DIFFIO_RX_R69N/PLL_R_CLKOUTN	G17	
DIFFIO_RX_R40P/CLK3P	DIFFIO_RX_R69P/PLL_R_CLKOUTP	H17	
BANK-8 VCCIO = 2.5V			
DIFFIO_RX_T38N/CLK4N	DIFFIO_RX_T52N/PLL_T_CLKOUTN	E6	
DIFFIO_RX_T38P/CLK4P	DIFFIO_RX_T52P/PLL_T_CLKOUTP	U6	
DIFFIO_RX_T40N/CLK5N			
DIFFIO_RX_T40P/CLK5P			



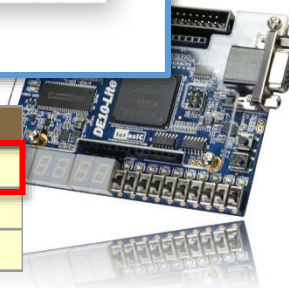
Default: 24MHz
 Default: 24MHz
 Default: 10MHz
 Default: 50MHz
 Default: 50MHz
 Default: 50MHz

Clocks Circuit of DE10-Lite

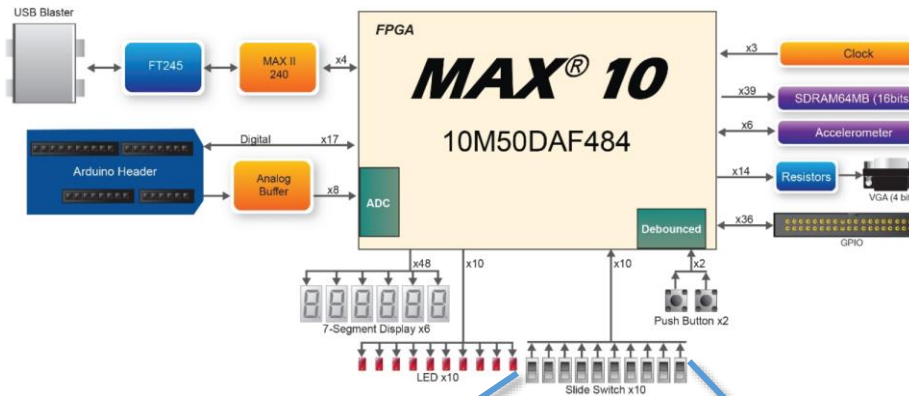
(br_i_clk)



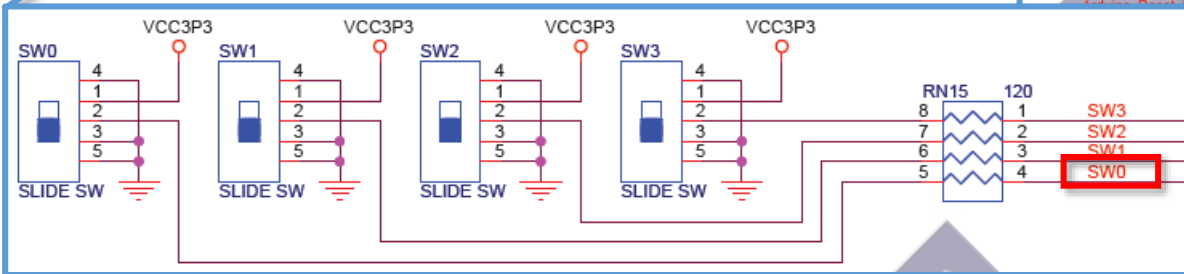
Signal Name	FPGA Pin No.	Description	I/O Standard
ADC_CLK_10	PIN_N5	10 MHz clock input for ADC (Bank 3B)	3.3-V LVTTTL
MAX10_CLK1_50	PIN_P11	50 MHz clock input(Bank 3B)	3.3-V LVTTTL
MAX10_CLK2_50	PIN_N14	50 MHz clock input(Bank 3B)	3.3-V LVTTTL



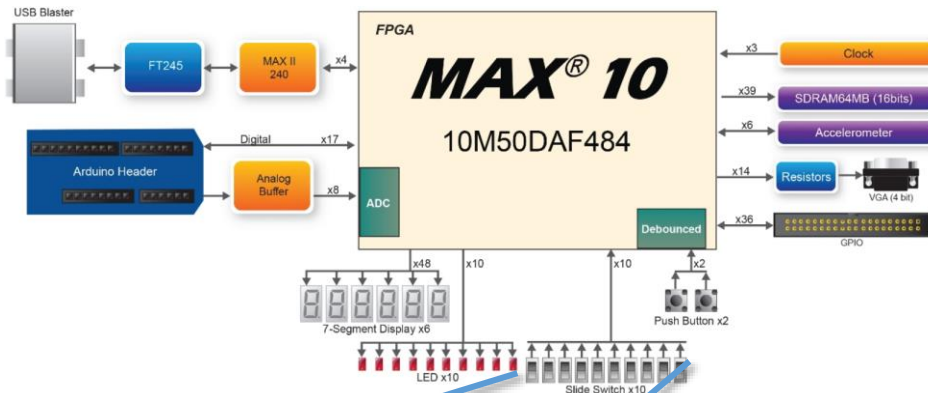
(br_i_enable)



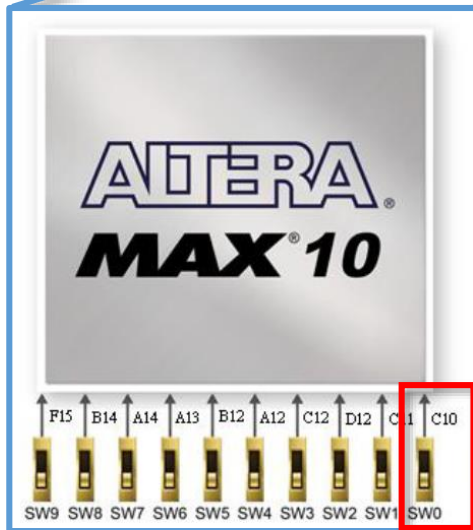
		MAX 10 TOP BANKS	
		BANK-7VCCIO = 3.3V	BANK-8VCCIO = 2.5V
HEX14	A17	DIFFIO_RX_T10N	DIFFIO_RX_T39N
HEX15	A18	DIFFIO_RX_T10P	DIFFIO_RX_T39P
HEX02	C15	DIFFIO_RX_T15N	DIFFIO_RX_T41N
HEX03	C16	DIFFIO_RX_T15P	DIFFIO_RX_T41P
HEX17	A16	DIFFIO_RX_T16N	DIFFIO_RX_T42P
HEX13	B16	DIFFIO_RX_T16P	DIFFIO_RX_T43N
	J13	DIFFIO_RX_T17N	DIFFIO_RX_T43P
	H14	DIFFIO_RX_T17P	DIFFIO_RX_T44N
LEDR5	C13	DIFFIO_RX_T18N	DIFFIO_RX_T45P
HEX00	C14	DIFFIO_RX_T18P	DIFFIO_RX_T45N
SW8	B14	DIFFIO_RX_T19N	DIFFIO_RX_T46P
SW7	A14	DIFFIO_RX_T19P	DIFFIO_RX_T46N
HEX01	E15	DIFFIO_RX_T1N	DIFFIO_RX_T47P
HEX04	E16	DIFFIO_RX_T1P	DIFFIO_RX_T47N
	E13	DIFFIO_RX_T20N	DIFFIO_RX_T48P
LEDR7	D14	DIFFIO_RX_T20P	DIFFIO_RX_T48N
	E12	DIFFIO_RX_T21P	DIFFIO_RX_T49P
LEDR4	D13	DIFFIO_RX_T21N	DIFFIO_RX_T51N
	J12	DIFFIO_RX_T22N	DIFFIO_RX_T51P
	H13	DIFFIO_RX_T22P	DIFFIO_RX_T53N
SW4	A12	DIFFIO_RX_T23N	DIFFIO_RX_T53P
SW6	A13	DIFFIO_RX_T23P	VREFB8N0
SW2	D12	DIFFIO_RX_T24N	IO_BANK8
SW3	C12	DIFFIO_RX_T24P	
LEDR2	A10	DIFFIO_RX_T25N	
LEDR8	A11	DIFFIO_RX_T25P	
SW0	C10	DIFFIO_RX_T26N	
SW1	B11	DIFFIO_RX_T26P	
LEDR9	B11	DIFFIO_RX_T27N	
SW5	B12	DIFFIO_RX_T27P	
	J11	DIFFIO_RX_T28N	
	H12	DIFFIO_RX_T28P	
KEY0	B8	DIFFIO_RX_T31N	
LEDR1	A9	DIFFIO_RX_T31P	
HEX06	C17	DIFFIO_RX_T2N	
HEX05	D17	DIFFIO_RX_T2P	
	C9	DIFFIO_RX_T30N	
LEDR3	B10	DIFFIO_RX_T30P	
KEY1	A7	DIFFIO_RX_T29P	
LEDR0	A8	DIFFIO_RX_T29N	
SW9	F15	DIFFIO_RX_T5N	
	F16	DIFFIO_RX_T5P	
	B19	DIFFIO_RX_T6N	
	C19	DIFFIO_RX_T6P	
	B17	DIFFIO_RX_T7N	
	C18	DIFFIO_RX_T7P	
	A19	DIFFIO_RX_T8N	
	A20	DIFFIO_RX_T8P	
SW3	E14	DIFFIO_RX_T9N	
SW2	D15	DIFFIO_RX_T9P	
SW1	B15	VREFB7N0	
SW0	A15	IO_BANK7	



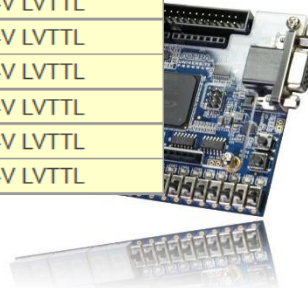
Slide Switches Circuit of DE10-Lite



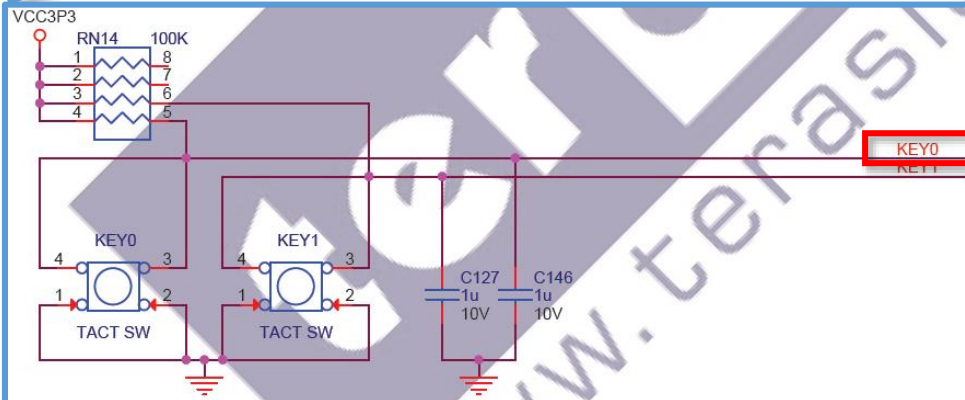
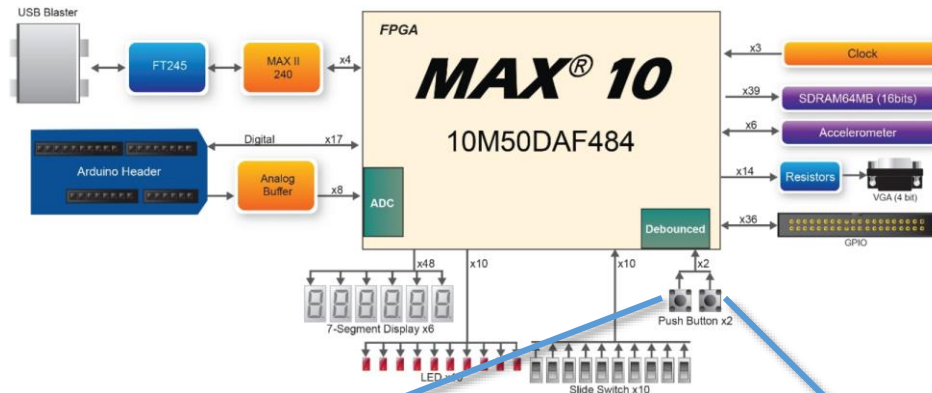
(br_i_enable)



Signal Name	FPGA Pin No.	Description	I/O Standard
SW0	PIN_C10	Slide Switch[0]	3.3-V LVTTTL
SW1	PIN_C11	Slide Switch[1]	3.3-V LVTTTL
SW2	PIN_D12	Slide Switch[2]	3.3-V LVTTTL
SW3	PIN_C12	Slide Switch[3]	3.3-V LVTTTL
SW4	PIN_A12	Slide Switch[4]	3.3-V LVTTTL
SW5	PIN_B12	Slide Switch[5]	3.3-V LVTTTL
SW6	PIN_A13	Slide Switch[6]	3.3-V LVTTTL
SW7	PIN_A14	Slide Switch[7]	3.3-V LVTTTL
SW8	PIN_B14	Slide Switch[8]	3.3-V LVTTTL
SW9	PIN_F15	Slide Switch[9]	3.3-V LVTTTL

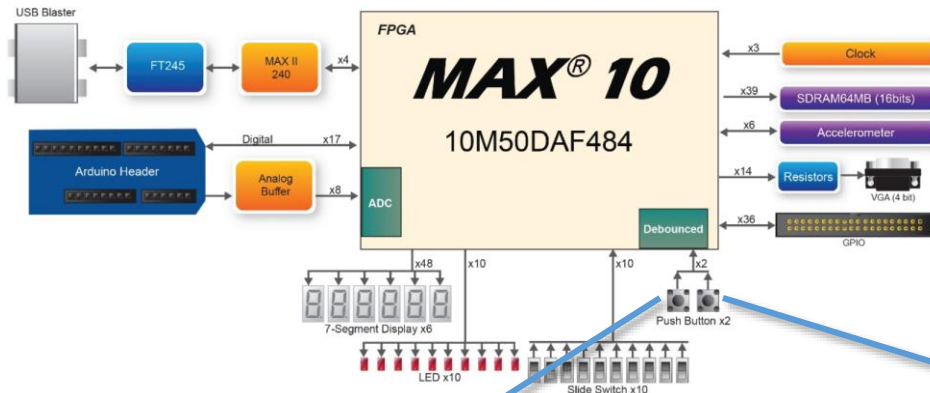


Push Buttons Circuit of DE10-Lite

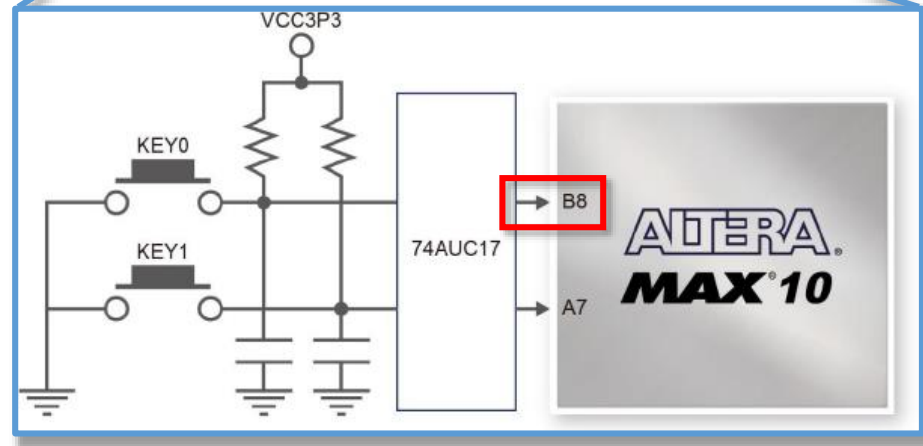


		USD	
		MAX 10 TOP BANKS	
		BANK-7VCCIO = 3.3V	BANK-8VCCIO = 2.5V
HEX14	A17	DIFFIO_RX_T10N	DIFFIO_RX_T39N
HEX15	A18	DIFFIO_RX_T10P	DIFFIO_RX_T39P
HEX02	C15	DIFFIO_RX_T15N	DIFFIO_RX_T41N
HEX03	C16	DIFFIO_RX_T15P	DIFFIO_RX_T41P
HEX17	A16	DIFFIO_RX_T16N	DIFFIO_RX_T42P
HEX13	B16	DIFFIO_RX_T16P	DIFFIO_RX_T43N
	J13	DIFFIO_RX_T17N	DIFFIO_RX_T43P
	H14	DIFFIO_RX_T17P	DIFFIO_RX_T44N
LEDR5	C13	DIFFIO_RX_T18P	DIFFIO_RX_T45N
HEX00	C14	DIFFIO_RX_T18N	DIFFIO_RX_T46P
SW8	B14	DIFFIO_RX_T19N	DIFFIO_RX_T46N
SW7	A14	DIFFIO_RX_T19P	DIFFIO_RX_T47P
HEX01	E15	DIFFIO_RX_T1N	DIFFIO_RX_T47N
HEX04	E16	DIFFIO_RX_T1P	DIFFIO_RX_T48P
	E13	DIFFIO_RX_T20N	DIFFIO_RX_T49N
LEDR7	D14	DIFFIO_RX_T20P	DIFFIO_RX_T49P
	E12	DIFFIO_RX_T21P	DIFFIO_RX_T51N
LEDR4	D13	DIFFIO_RX_T21N	DIFFIO_RX_T51P
	J12	DIFFIO_RX_T22N	DIFFIO_RX_T53N
	H13	DIFFIO_RX_T22P	DIFFIO_RX_T53P
SW4	A12	DIFFIO_RX_T23N	VREFB7N0
SW6	A13	DIFFIO_RX_T23P	IO_BANK8
SW2	D12	DIFFIO_RX_T24N	
SW3	C12	DIFFIO_RX_T24P	
LEDR2	A10	DIFFIO_RX_T25N	
LEDR8	A11	DIFFIO_RX_T25P	
SW0	C10	DIFFIO_RX_T26N	
SW1	C11	DIFFIO_RX_T26P	
LEDR9	B11	DIFFIO_RX_T27N	
SW5	B12	DIFFIO_RX_T27P	
	J11	DIFFIO_RX_T28N	
	H11	DIFFIO_RX_T28P	
KEY0	B8	DIFFIO_RX_T31P	
LEDR1	A9	DIFFIO_RX_T31P	
HEX06	C17	DIFFIO_RX_T2N	
HEX05	D17	DIFFIO_RX_T2P	
	C9	DIFFIO_RX_T30N	
LEDR3	B10	DIFFIO_RX_T30P	
KEY1	A7	DIFFIO_RX_T29P	
LEDR0	A8	DIFFIO_RX_T29N	
SW9	F15	DIFFIO_RX_T5N	
Arduino_Reset_n	F16	DIFFIO_RX_T5P	
HEX22	B19	DIFFIO_RX_T6N	
HEX33	C19	DIFFIO_RX_T6P	
HEX16	B17	DIFFIO_RX_T7N	
HEX10	C18	DIFFIO_RX_T7P	
HEX27	A19	DIFFIO_RX_T8N	
HEX21	A20	DIFFIO_RX_T8P	
LEDR6	E14	DIFFIO_RX_T9N	
HEX07	D15	DIFFIO_RX_T9P	
	B15	VREFB7N0	
	A15	IO_BANK7	

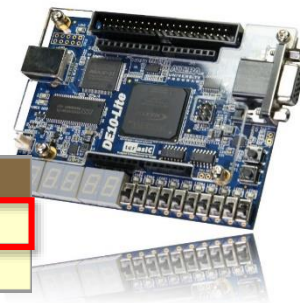
Push Buttons Circuit of DE10-Lite



(br_i_rstn)

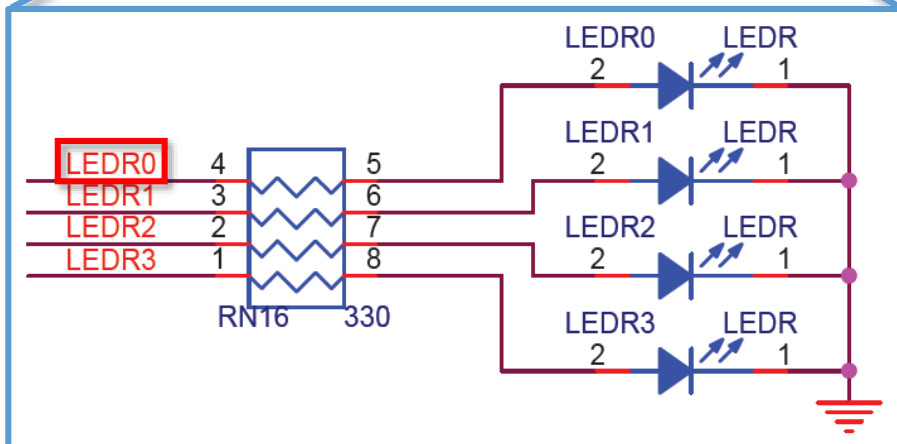
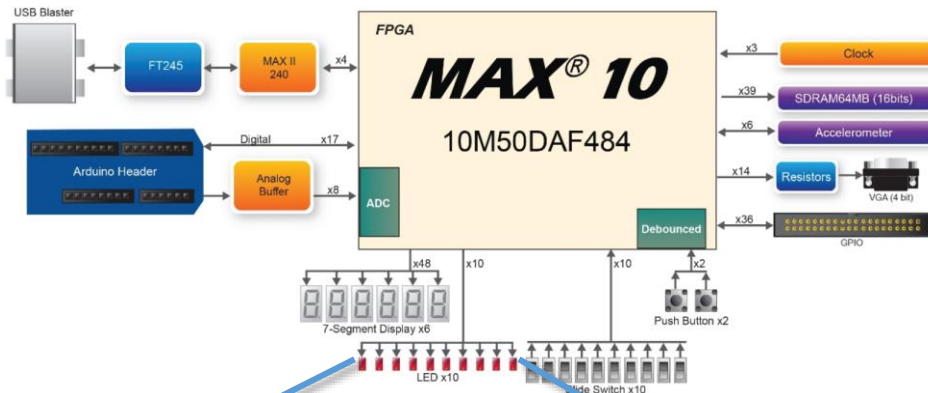


Signal Name	FPGA Pin No.	Description	I/O Standard
KEY0	PIN_B8	Push-button[0]	3.3 V SCHMITT TRIGGER"
KEY1	PIN_A7	Push-button[1]	3.3 V SCHMITT TRIGGER"



LEDs Circuit of DE10-Lite

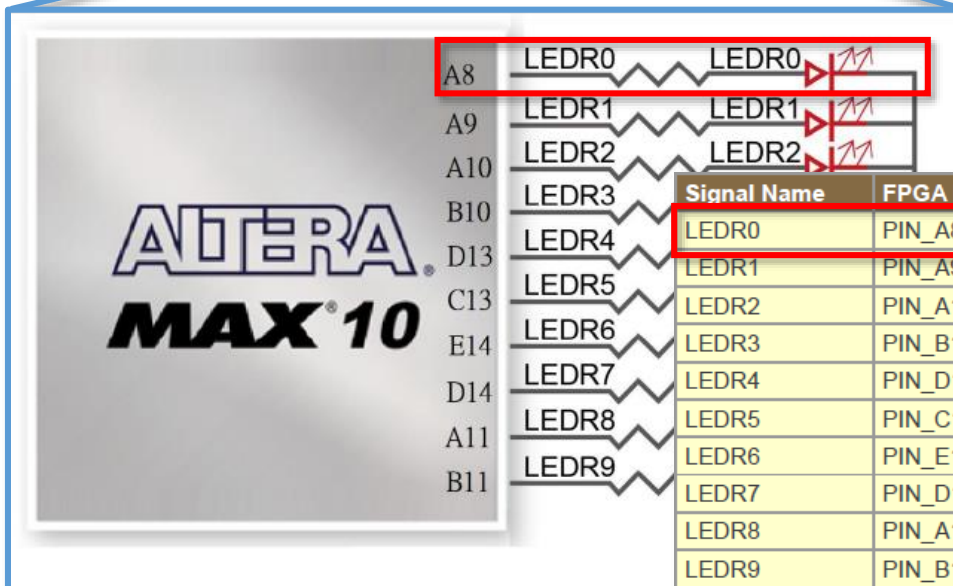
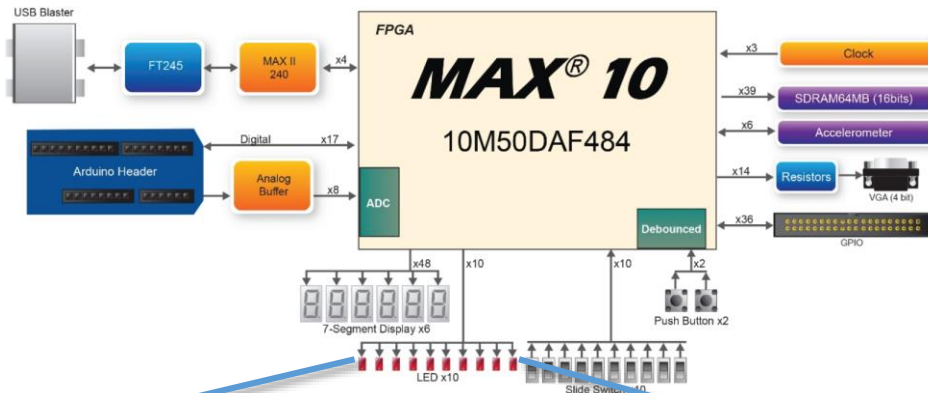
(br_o_op)



MAX 10 TOP BANKS		BANK-7VCCIO = 3.3V		BANK-8VCCIO = 2.5V	
HEX14	A17	DIFFIO_RX_T10N		DIFFIO_RX_T39N	C7
HEX15	A18	DIFFIO_RX_T10P		DIFFIO_RX_T39P	C8
HEX02	C15	DIFFIO_RX_T15N		DIFFIO_RX_T41N	A6
HEX03	C16	DIFFIO_RX_T15P		DIFFIO_RX_T41P	B7
HEX17	A16	DIFFIO_RX_T16N		DIFFIO_RX_T42P	D8
HEX13	B16	DIFFIO_RX_T16P		DIFFIO_RX_T43N	A4
	J13	DIFFIO_RX_T17P		DIFFIO_RX_T44N	A5
	H14	DIFFIO_RX_T18N		DIFFIO_RX_T45P	E9
LEDR5	C13	DIFFIO_RX_T18P		DIFFIO_RX_T46N	A2
HEX00	C14	DIFFIO_RX_T19N		DIFFIO_RX_T46P	A3
SW8	B14	DIFFIO_RX_T19P		DIFFIO_RX_T47N	B3
SW7	A14	DIFFIO_RX_T19P		DIFFIO_RX_T47P	B4
HEX01	E15	DIFFIO_RX_T1P		DIFFIO_RX_T47N	B5
HEX04	E16	DIFFIO_RX_T20N		DIFFIO_RX_T48P	C4
	E13	DIFFIO_RX_T20P		DIFFIO_RX_T49N	E8
LEDR7	D14	DIFFIO_RX_T21P		DIFFIO_RX_T49P	D5
	E12	DIFFIO_RX_T21N		DIFFIO_RX_T51N	C5
LEDR4	D13	DIFFIO_RX_T22N		DIFFIO_RX_T51P	B1
	J12	DIFFIO_RX_T22P		DIFFIO_RX_T53N	B2
	H13	DIFFIO_RX_T23N		DIFFIO_RX_T53P	C2
SW4	A12	DIFFIO_RX_T23P		DIFFIO_RX_T53P	C3
SW6	A13	DIFFIO_RX_T24N		VREFB8N0	D7
SW2	D12	DIFFIO_RX_T24P		IO_BANK8	C6
SW3	C12	DIFFIO_RX_T24P			
LEDR2	A10	DIFFIO_RX_T25N			
LEDR8	A11	DIFFIO_RX_T25P			
SW0	C10	DIFFIO_RX_T28N			
SW1	C11	DIFFIO_RX_T28P			
LEDR9	B11	DIFFIO_RX_T27N			
SW5	B12	DIFFIO_RX_T27P			
	J11	DIFFIO_RX_T28N			
	H12	DIFFIO_RX_T28P			
KEY0	B8	DIFFIO_RX_T31N			
LEDR1	A9	DIFFIO_RX_T31P			
HEX06	C17	DIFFIO_RX_T2N			
HEX05	D17	DIFFIO_RX_T2P			
	C9	DIFFIO_RX_T30N			
LEDR3	B10	DIFFIO_RX_T30P			
KEY1	A9	DIFFIO_RX_T29P			
LEDR0	A8	DIFFIO_RX_T29N			
SW9	F15	DIFFIO_RX_T5N			
Arduino_Reset_n	F16	DIFFIO_RX_T5P			
HEX22	B19	DIFFIO_RX_T6N			
HEX33	C19	DIFFIO_RX_T6P			
HEX16	B17	DIFFIO_RX_T7N			
HEX10	C18	DIFFIO_RX_T7P			
HEX27	A19	DIFFIO_RX_T8N			
HEX21	A20	DIFFIO_RX_T8P			
LEDR6	E14	DIFFIO_RX_T9N			
D15		DIFFIO_RX_T9P			
HEX07	B15	DIFFIO_RX_T9P			
	A15	VREFB7N0			
		IO_BANK7			

LEDs Circuit of DE10-Lite

(br_o_op)



Signal Name	FPGA Pin No.	Description	I/O Standard
LEDR0	PIN_A8	LED [0]	3.3-V LVTTTL
LEDR1	PIN_A9	LED [1]	3.3-V LVTTTL
LEDR2	PIN_A10	LED [2]	3.3-V LVTTTL
LEDR3	PIN_B10	LED [3]	3.3-V LVTTTL
LEDR4	PIN_D13	LED [4]	3.3-V LVTTTL
LEDR5	PIN_C13	LED [5]	3.3-V LVTTTL
LEDR6	PIN_E14	LED [6]	3.3-V LVTTTL
LEDR7	PIN_D14	LED [7]	3.3-V LVTTTL
LEDR8	PIN_A11	LED [8]	3.3-V LVTTTL
LEDR9	PIN_B11	LED [9]	3.3-V LVTTTL