



جامعة حلب في المناطق المحررة  
كلية الهندسة المعلوماتية  
السنة الرابعة

مقرر عملي

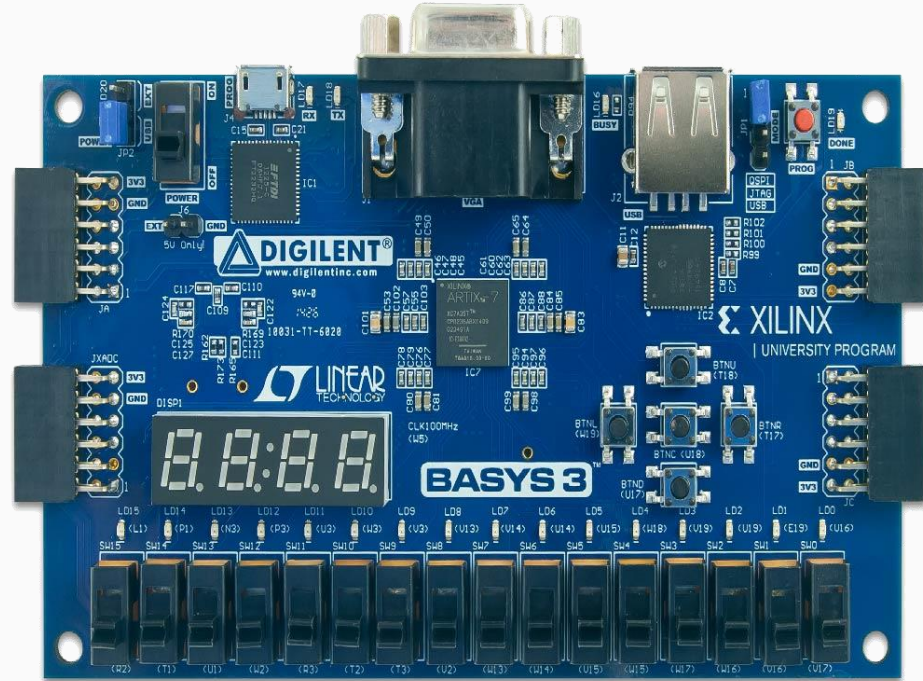
# بنية وتنظيم الحواسيب 2

توليد نبضات PWM

د.م. عبد القادر غزال

م. محمد نور بدوي

العام الدراسي: 2023 - 2024



المحاضرة العملية الثامنة

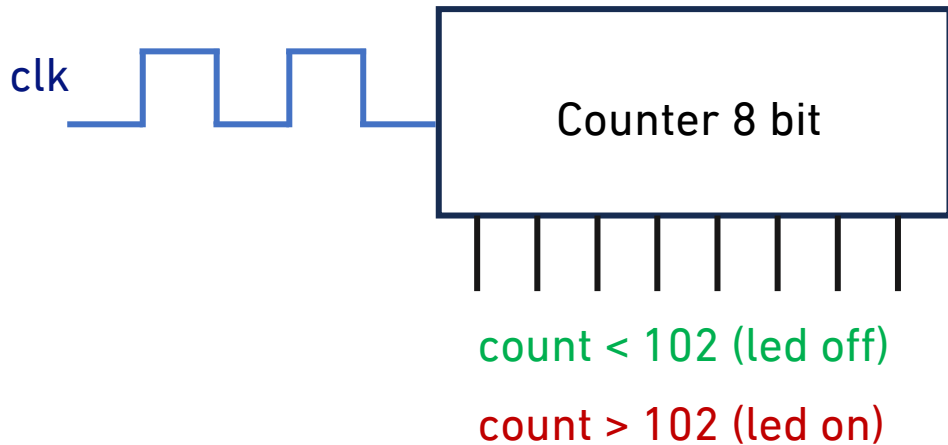


# توليد نبضات PWM

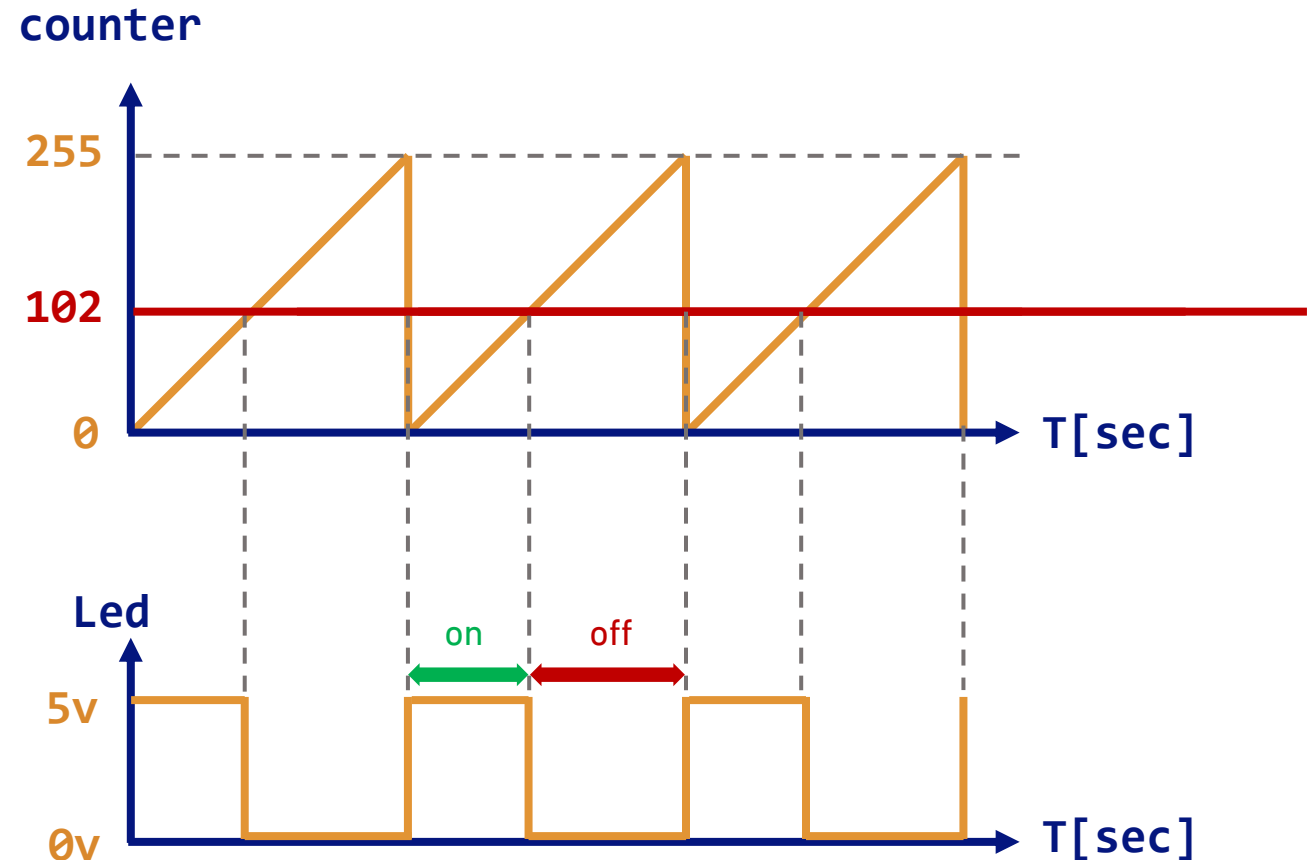


سنقوم في هذا المثال بتوليد نبضات PWM وتمريرها إلى ليد ضوئي باستخدام عداد 8bit ، حيث تكون نسبة تشغيل الليد

Duty Cycle = 40%



$$\text{Count} = 255 * \text{Duty Cycle}$$
$$\text{Count} = 255 * (40/100) = 102$$





# توليد نبضات PWM



سنقوم في هذا المثال بتوليد نبضات PWM وتمريرها إلى ليد ضوئي باستخدام عداد 8bit ، حيث تكون نسبة تشغيل الـ LED

Duty Cycle = 40%

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE ieee.std_logic_unsigned.ALL;

ENTITY pwm_gen IS
    PORT (
        clk : IN STD_LOGIC;
        rst : IN STD_LOGIC;
        pwm_out : OUT STD_LOGIC);
END pwm_gen;

ARCHITECTURE Behavioral OF pwm_gen IS
    SIGNAL counter : STD_LOGIC_VECTOR
(7 DOWNTO 0) := "00000000";
BEGIN
    PROCESS (clk, rst)
    BEGIN
        IF rst = '1' THEN
            counter <= (OTHERS => '0');

```

```
ELSIF rising_edge(clk) THEN
    counter <= counter + 1;
END IF;
END PROCESS;
pwm_out <= '1' WHEN counter < 102 ELSE '0';
END Behavioral;
```

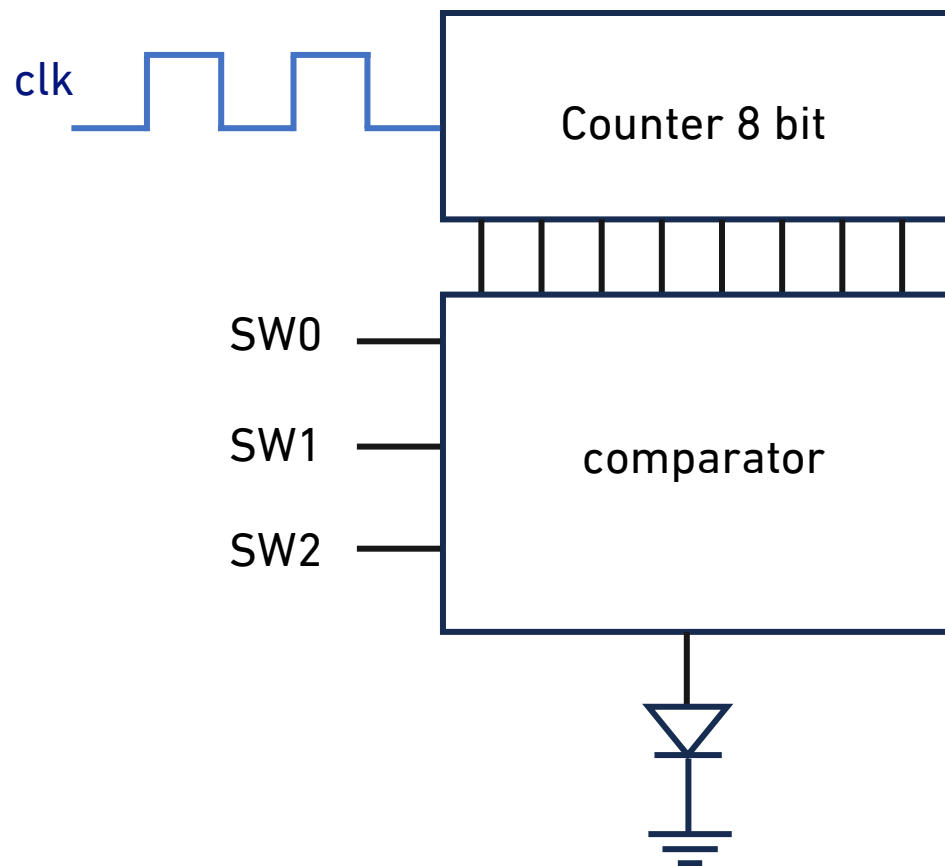




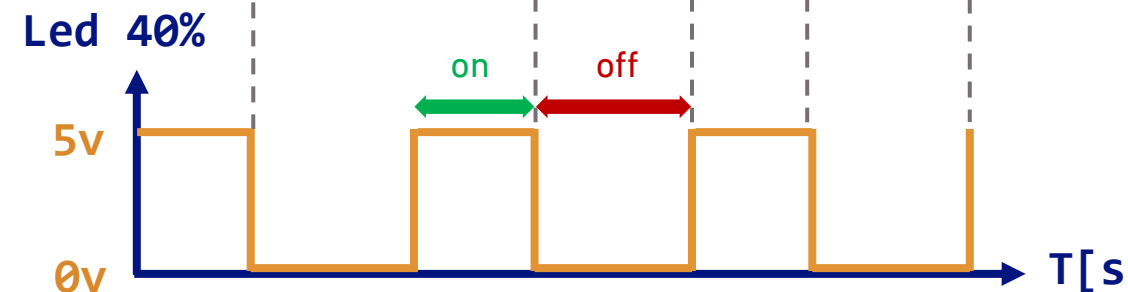
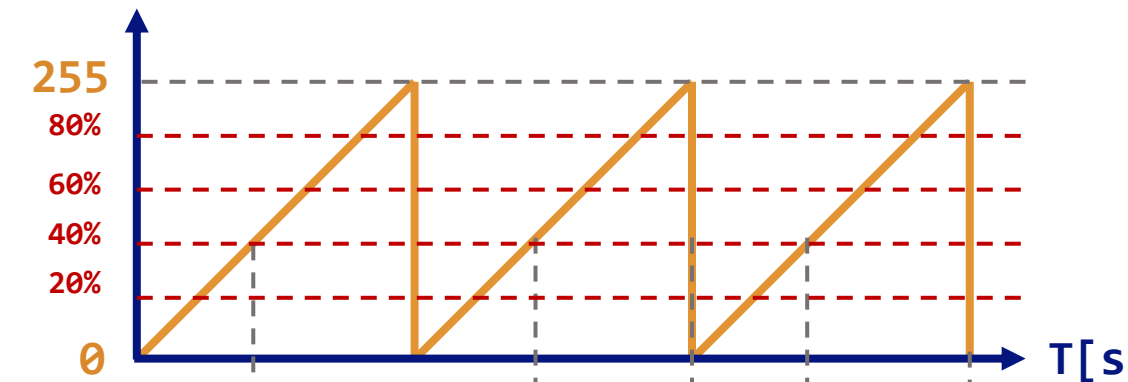
# توليد نبضات PWM



سنقوم بالتعديل على المثال السابق بحيث تكون نسبة تشغيل متغيرة وليست ثابتة، بحيث يمكن التبديل بين 6 نسب تشغيل على التوالي (0% - 20% - 40% - 60% - 80% - 100%)



counter





# توليد نبضات PWM



```
LIBRARY IEEE;  
USE IEEE.STD_LOGIC_1164.ALL;  
USE ieee.std_logic_unsigned.ALL;
```

```
ENTITY pwm_gen IS  
  PORT (  
    rst : IN STD_LOGIC;  
    pwm_out : OUT STD_LOGIC);  
END pwm_gen;
```

```
ARCHITECTURE Behavioral OF pwm_gen IS  
  SIGNAL counter : STD_LOGIC_VECTOR(7 DOWNTO 0) := "00000000";  
  SIGNAL Switch : STD_LOGIC_VECTOR(2 DOWNTO 0) := "000";  
  SIGNAL Duty : NATURAL := 0;  
  SIGNAL clk : STD_LOGIC;  
BEGIN  
  
  clk_process : PROCESS  
  BEGIN
```

سنقوم بالتعديل على المثال السابق بحيث تكون نسبة تشغيل متغيرة  
وليست ثابتة، بحيث يمكن التبديل بين 6 نسب تشغيل على التوالي  
(0% - 20% - 40% - 60% - 80% - 100%)



# توليد نبضات PWM

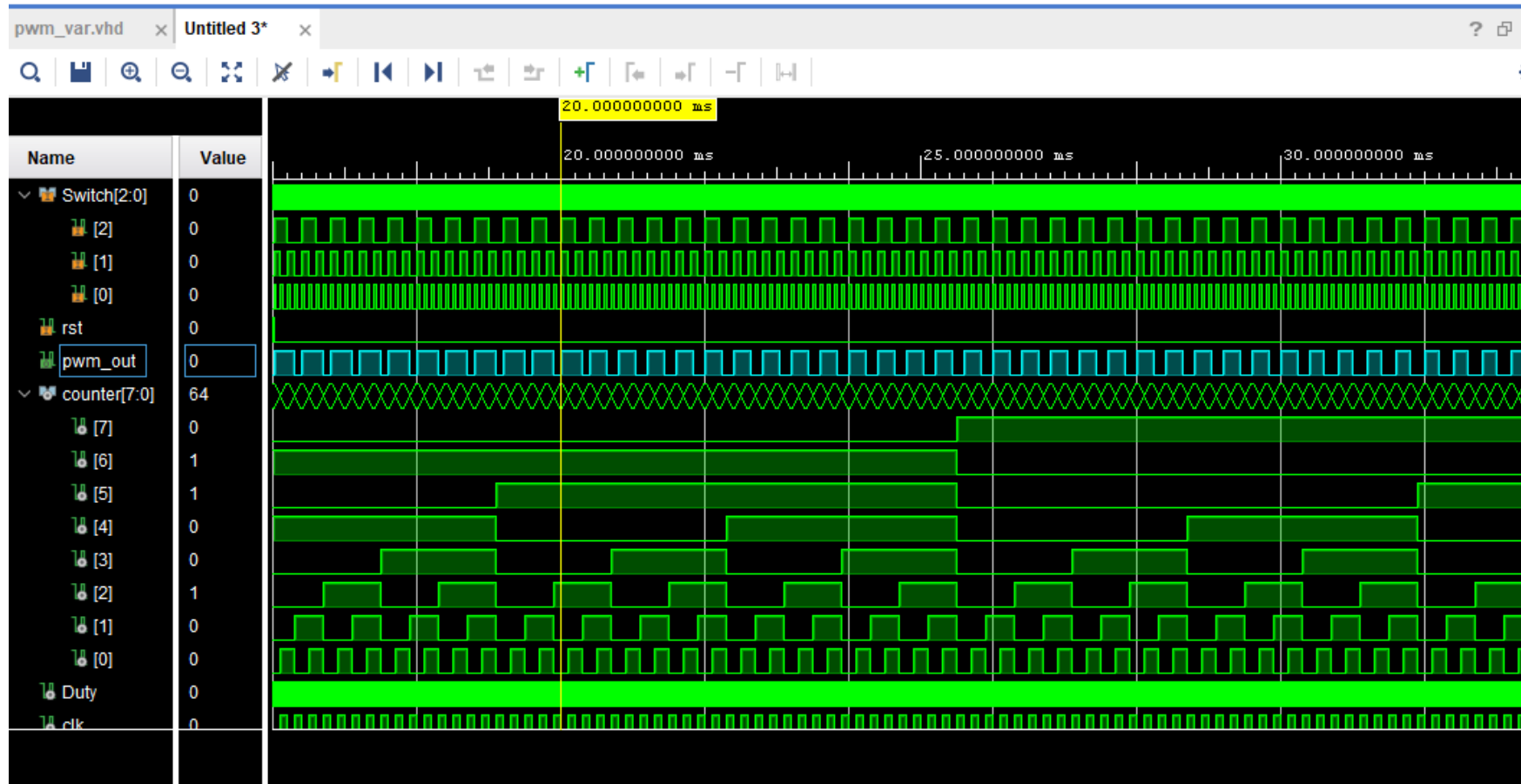


```
clk <= '0';  
  WAIT FOR 100us;  
  --for 100 us signal is '0'.  
  clk <= '1';  
  WAIT FOR 100us;  
  --for next 100 us signal is '1'.  
END PROCESS clk_process;
```

```
choose_process : PROCESS (Switch)  
BEGIN  
  
  CASE Switch IS  
    WHEN "000" => Duty <= 0;  
    WHEN "001" => Duty <= 51;  
    WHEN "010" => Duty <= 102;  
    WHEN "011" => Duty <= 153;  
    WHEN "100" => Duty <= 204;  
    WHEN OTHERS => Duty <= 255;
```

```
  END CASE;  
END PROCESS choose_process;
```

```
pwmGenerate : PROCESS (clk, rst)  
BEGIN  
  IF rst = '1' THEN  
    counter <= (OTHERS => '0');  
  ELSIF rising_edge(clk) THEN  
    counter <= counter + 1;  
  END IF;  
END PROCESS pwmGenerate;  
-- pwm Generate  
pwm_out <= '1' WHEN ( counter < Duty  
  and Duty > 0 )  
ELSE '0';  
END Behavioral;
```







# انتهت المحاضرة