

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه
خواجه نصیرالدین طوسی

نیمسال اول تحصیلی

1402-1403

تهیه کننده: رامین توکلی

درس: FPGA

شماره دانشجویی: ۹۹۲۵۰۶۳

استاد درس: دکتر حسینی نژاد

تمرین شماره دو

دانشکده مهندسی برق

سوال ۱) مدار کانولوشن را به شکل ساختاری طراحی و شبیه سازی کنید.

در این طراحی دو ورودی ۸ بیتی، یک ورودی کلاک، و یک خروجی نهایی مدار کانولوشن خواهیم داشت. در این جا قصد این است که ورودی به صورت ترتیبی با هر لبه کلاک وارد شده، در هم ضرب شوند و با مقادیر قبلی جمع شوند. فرمول کلی به صورت زیر است:

10

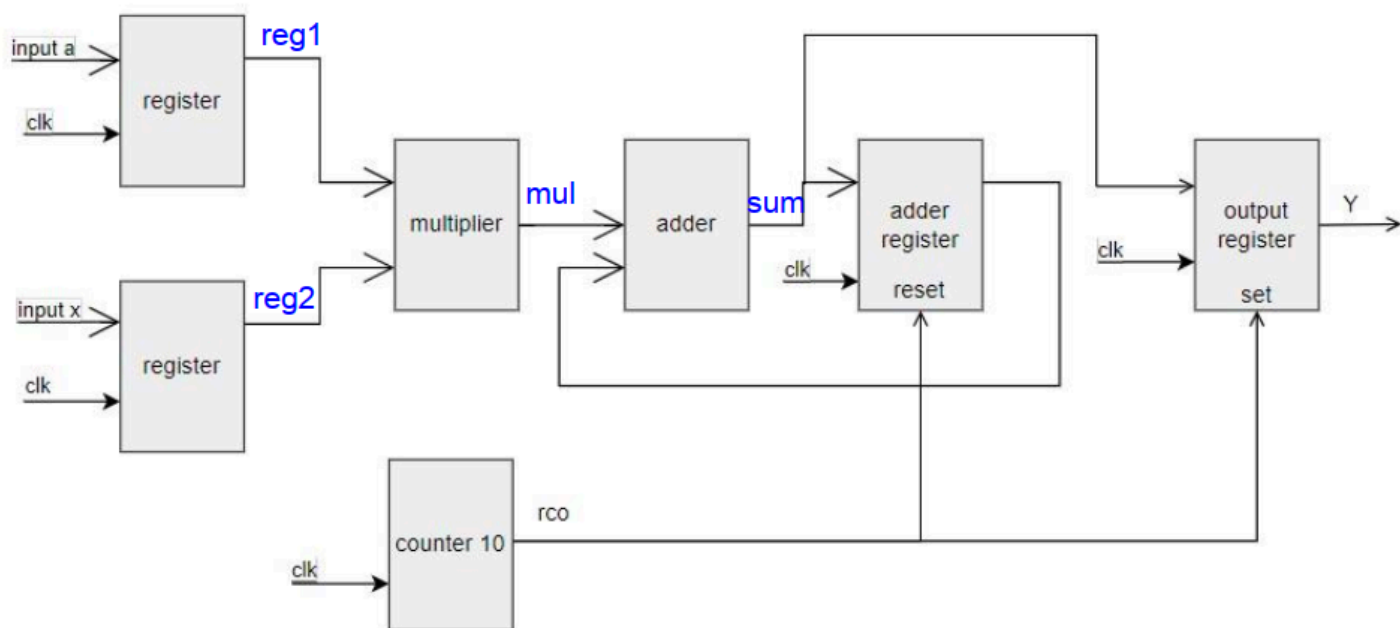
$$y = \sum_{i=1}^{10} a_i x_i = a_1 x_1 + a_2 x_2 + \dots + a_{10} x_{10}$$

برای این طراحی بخش های زیر لازم است:

- یک شمارنده تا عدد ۱۰ برای شمارش تعداد ورودی های داده شده به مدار.
- یک جمع کننده برای جمع دو عدد ۸ بیتی.
- یک ضرب کننده برای ضرب دو عدد ۸ بیتی.
- یک رجیستر برای نگه داشتن مقادیر ورودی ها و نگه داری مجموع ضرب ورودی ها. رجیستر ها دارای یک پایه **set** و یک پایه **reset** هستند.

ماژول های گفته شده را جداگانه طراحی کرده و به صورت ساختاری به هم متصل کنید.

یک نقشه کلی از طراحی بالا به صورت زیر خواهد بود:



```

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 CONVOLUTION is
7
8  port(clkp : in std_logic;
9        input_a : in std_logic_vector(7 downto 0);
10       input_x : in std_logic_vector(7 downto 0);
11       Y : out std_logic_vector( 19 downto 0));
12
13  end CONVOLUTION;
14
15  architecture Structral of CONVOLUTION is
16
17  -----
18  signal R : std_logic ;
19  signal reg1, reg2 : std_logic_vector(7 downto 0);
20  signal mult : std_logic_vector(15 downto 0);
21  signal sum : std_logic_vector(19 downto 0) ;
22  signal add_reg : std_logic_vector(19 downto 0);
23  -----
24
25  -- component decleration
26  -----
27  component Multiplier
28  generic(n: integer := 8);
29  port(IN1 : in std_logic_vector(n-1 downto 0);
30        IN2 : in std_logic_vector(n-1 downto 0);
31        MUL : out std_logic_vector( n+n-1 downto 0));
32  end component;
33  -----
34  component Adder
35  Generic(n: integer range 0 to 63 := 19);
36  port(IN1 : in std_logic_vector(n-4 downto 0);
37        IN2 : in std_logic_vector(n downto 0);
38        ADD : out std_logic_vector( n downto 0));
39  end component;
40  -----
41  component Counter
42  port(
43        clk: in std_logic;
44        rco: out std_logic);
45  end component;
46  -----

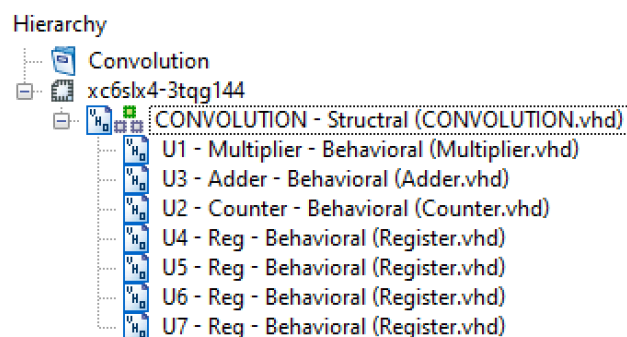
```

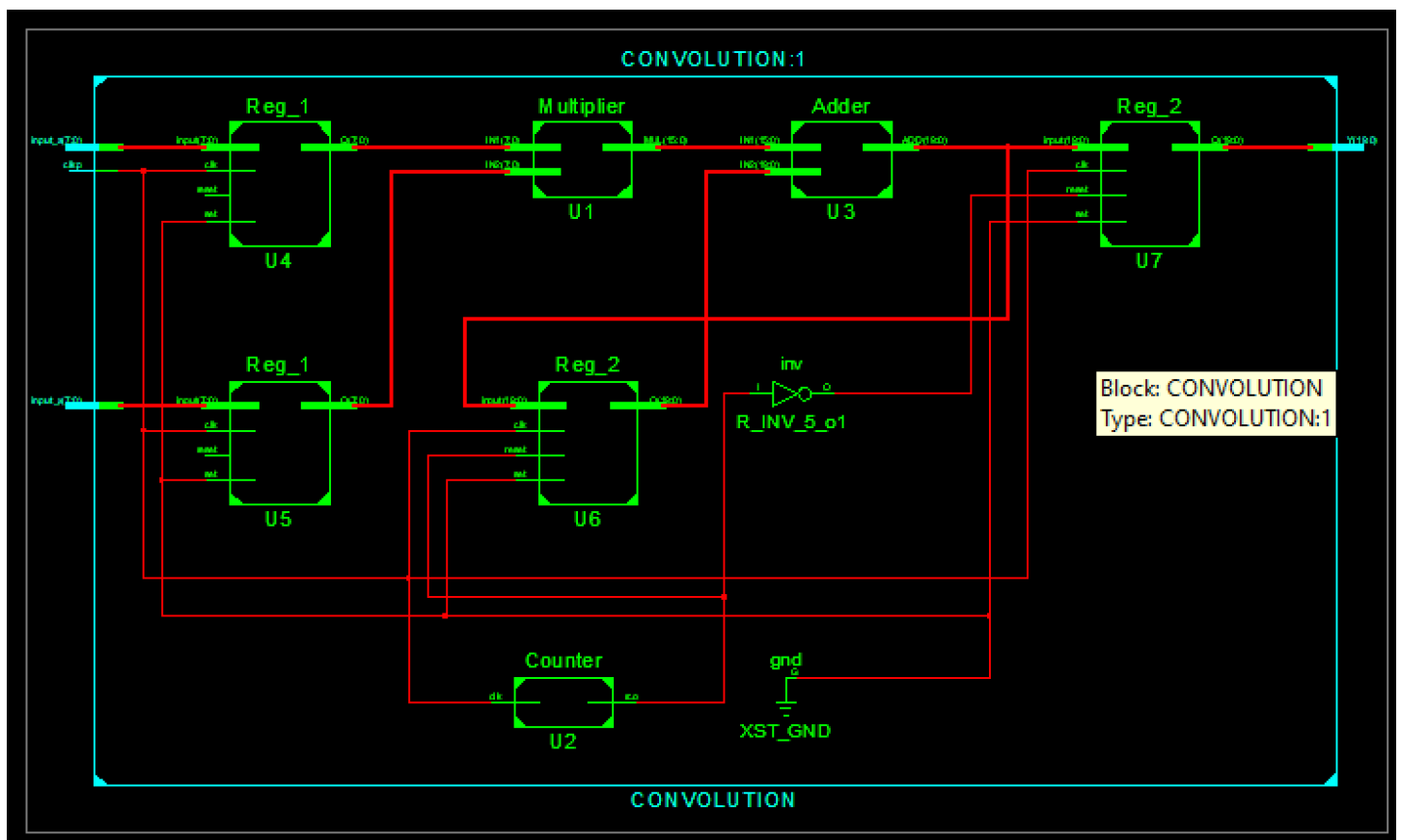
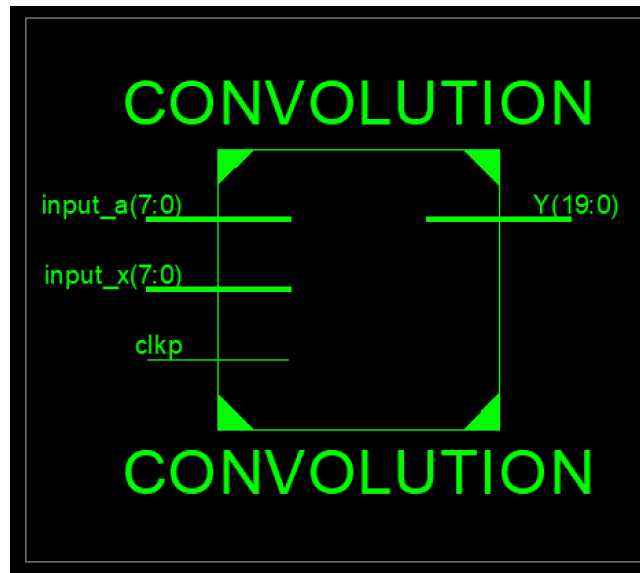
```

47 component Reg
48 Generic ( n: integer range 0 to 63 := 8);
49 port( input : in std_logic_vector(n-1 downto 0);
50       set : in std_logic; -- active high
51       reset: in std_logic; -- avtive low
52       clk : in std_logic;
53       Q : out std_logic_vector(n-1 downto 0));
54 end component;
55 -----
56 begin
57
58 U1 : Multiplier port map ( reg1, reg2, mult);
59 U3 : Adder port map ( mult, add_reg, sum);
60 U2 : Counter port map ( clkp,R);
61
62 U4: Reg
63 generic map ( n => 8)
64 port map ( input_a, '0', '1', clkp, reg1);
65
66 U5: Reg
67 generic map ( n => 8)
68 port map ( input_x, '0', '1', clkp, reg2);
69
70 U6: Reg
71 generic map ( n => 20)
72 port map ( sum, '0', R, clkp, add_reg);
73
74 U7: Reg
75 generic map ( n => 20)
76 port map ( sum, '0', (not R), clkp, Y);
77
78 end Structral;
79

```

نمایش Hierarchy طراحی:





```

=====
*                               HDL Synthesis                               *
=====

Synthesizing Unit <CONVOLUTION>.
  Related source file is "C:\xilinx\ise-project\FPGA_class\HW2\Convolution\CONVOLUTION.vhd".
  Summary:
    no macro.
Unit <CONVOLUTION> synthesized.

Synthesizing Unit <Multiplier>.
  Related source file is "C:\xilinx\ise-project\FPGA_class\HW2\Convolution\Multiplier.vhd".
  n = 8
  Found 8x8-bit multiplier for signal <MUL> created at line 17.
  Summary:
    inferred    1 Multiplier(s).
Unit <Multiplier> synthesized.

Synthesizing Unit <Adder>.
  Related source file is "C:\xilinx\ise-project\FPGA_class\HW2\Convolution\Adder.vhd".
  n = 19
  Found 20-bit adder for signal <ADD> created at line 20.
  Summary:
    inferred    1 Adder/Subtractor(s).
Unit <Adder> synthesized.

Synthesizing Unit <Counter>.
  Related source file is "C:\xilinx\ise-project\FPGA_class\HW2\Convolution\Counter.vhd".
  Found 1-bit register for signal <R>.
  Found 4-bit register for signal <count>.
  Found 4-bit adder for signal <count[3]_GND_9_o_add_0_OUT> created at line 29.
  Summary:
    inferred    1 Adder/Subtractor(s).
    inferred    5 D-type flip-flop(s).
Unit <Counter> synthesized.

Synthesizing Unit <Reg_1>.
  Related source file is "C:\xilinx\ise-project\FPGA_class\HW2\Convolution\Register.vhd".
  n = 8
  Found 8-bit register for signal <Qout>.
  Summary:
    inferred    8 D-type flip-flop(s).
Unit <Reg_1> synthesized.

Synthesizing Unit <Reg_2>.
  Related source file is "C:\xilinx\ise-project\FPGA_class\HW2\Convolution\Register.vhd".
  n = 20
  Found 20-bit register for signal <Qout>.
  Summary:
    inferred    20 D-type flip-flop(s).
Unit <Reg_2> synthesized.

=====
HDL Synthesis Report

Macro Statistics
# Multipliers                : 1
  8x8-bit multiplier          : 1
# Adders/Subtractors         : 2
  20-bit adder                : 1
  4-bit adder                 : 1
# Registers                  : 6
  1-bit register              : 1
  20-bit register             : 2
  4-bit register              : 1
  8-bit register              : 2

```

```

=====
*                               Advanced HDL Synthesis                               *
=====

```

Synthesizing (advanced) Unit <CONVOLUTION>.

The following registers are absorbed into accumulator <U6/Qout>: 1 register on signal <U6/Qout>.
Unit <CONVOLUTION> synthesized (advanced).

Synthesizing (advanced) Unit <Counter>.

The following registers are absorbed into counter <count>: 1 register on signal <count>.
Unit <Counter> synthesized (advanced).

```

=====
Advanced HDL Synthesis Report

```

Macro Statistics

```

# Multipliers                      : 1
  8x8-bit multiplier              : 1
# Adders/Subtractors              : 2
  20-bit adder                    : 1
  4-bit adder                     : 1
# Counters                        : 1
  4-bit up counter                : 1
# Accumulators                    : 1
  20-bit up accumulator           : 1
# Registers                       : 37
  Flip-Flops                     : 37

```

```

=====
*                               Design Summary                               *
=====

```

Top Level Output File Name : CONVOLUTION.ngc

Primitive and Black Box Usage:

```

-----
# BELS                      : 127
#   GND                      : 1
#   INV                      : 2
#   LUT1                     : 8
#   LUT2                     : 33
#   LUT3                     : 1
#   LUT4                     : 3
#   MUXCY                    : 38
#   VCC                      : 1
#   XORCY                    : 40
# FlipFlops/Latches         : 61
#   FD                      : 17
#   FDR                     : 44
# Clock Buffers             : 1
#   BUFGP                   : 1
# IO Buffers                 : 36
#   IBUF                    : 16
#   OBUF                    : 20
# DSPs                      : 1
#   DSP48A1                  : 1

```

Device utilization summary:

```

-----
Selected Device : 6slx4tqgl44-3

```

Slice Logic Utilization:

Number of Slice Registers:	61	out of	4800	1%
Number of Slice LUTs:	47	out of	2400	1%
Number used as Logic:	47	out of	2400	1%

Slice Logic Distribution:

Number of LUT Flip Flop pairs used:	63			
Number with an unused Flip Flop:	2	out of	63	3%
Number with an unused LUT:	16	out of	63	25%
Number of fully used LUT-FF pairs:	45	out of	63	71%
Number of unique control sets:	4			

IO Utilization:

Number of IOs:	37			
Number of bonded IOBs:	37	out of	102	36%

Specific Feature Utilization:

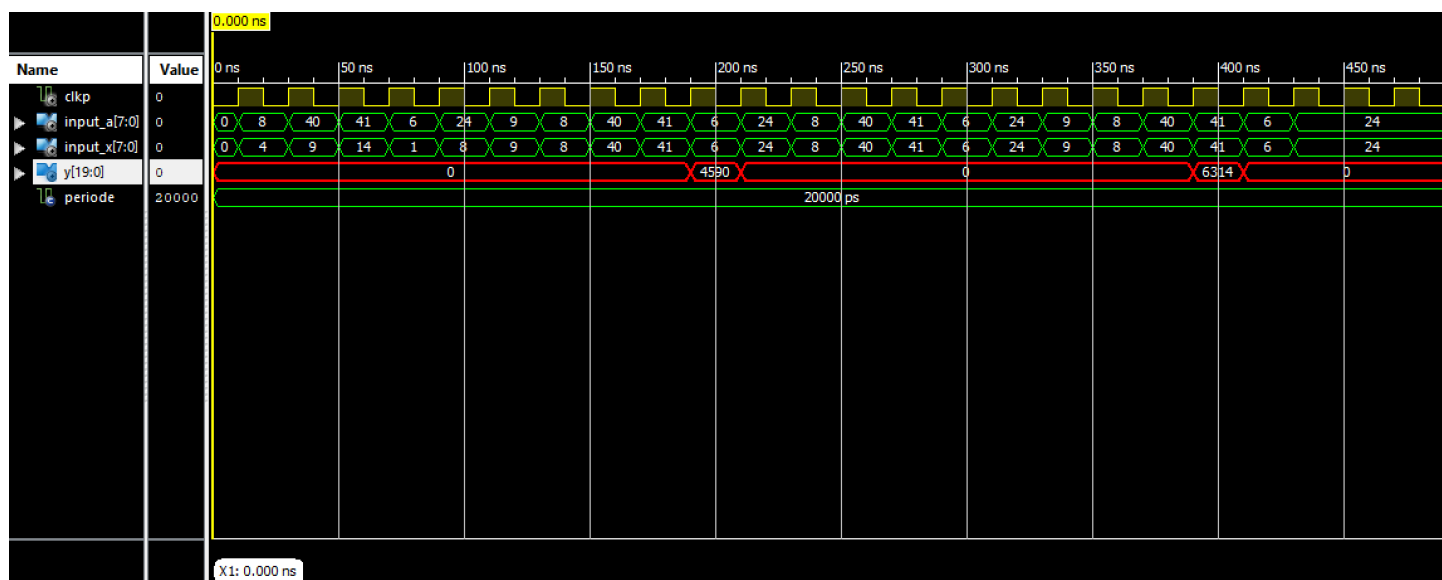
Number of BUFG/BUFGCTRLs:	1	out of	16	6%
Number of DSP48A1s:	1	out of	8	12%

Partition Resource Summary:

```

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 TB_CONVOLUTION IS
7  GENERIC( periode : time := 20 ns);
8  END TB_CONVOLUTION;
9
10 ARCHITECTURE behavior OF TB_CONVOLUTION IS
11
12     -- Component Declaration for the Unit Under Test (UUT)
13
14     COMPONENT CONVOLUTION
15     PORT(
16         clkp : IN  std_logic;
17         input_a : IN  std_logic_vector(7 downto 0);
18         input_x : IN  std_logic_vector(7 downto 0);
19         Y : OUT  std_logic_vector(19 downto 0)
20     );
21     END COMPONENT;
22
23
24     --Inputs
25     signal clkp : std_logic := '0';
26     signal input_a : std_logic_vector(7 downto 0) := (others => '0');
27     signal input_x : std_logic_vector(7 downto 0) := (others => '0');
28
29     --Outputs
30     signal Y : std_logic_vector(19 downto 0);
31
32 BEGIN
33
34
35     -- Instantiate the Unit Under Test (UUT)
36     uut: CONVOLUTION PORT MAP (
37         clkp => clkp,
38         input_a => input_a,
39         input_x => input_x,
40         Y => Y
41     );
42
43     clkp <= not clkp after periode/2;
44     input_a <= "00001000" after 10 ns, "00101000" after 30 ns, "00101001" after 50 ns, "00000110" after 70 ns, "00011000" after 90 ns,
45     "00001001" after 110 ns, "00001000" after 130 ns, "00101000" after 150 ns, "00101001" after 170 ns, "00000110" after 190 ns, "00011000"
46     "00001000" after 230 ns, "00101000" after 250 ns, "00101001" after 270 ns, "00000110" after 290 ns, "00011000" after 310 ns,
47     "00001001" after 330 ns, "00001000" after 350 ns, "00101000" after 370 ns, "00101001" after 390 ns, "00000110" after 410 ns, "00011000"
48     input_x <= "00000100" after 10 ns, "00001001" after 30 ns, "00001110" after 50 ns, "00000001" after 70 ns, "00001000" after 90 ns,
49     "00001001" after 110 ns, "00001000" after 130 ns, "00101000" after 150 ns, "00101001" after 170 ns, "00000110" after 190 ns, "00011000"
50     "00001000" after 230 ns, "00101000" after 250 ns, "00101001" after 270 ns, "00000110" after 290 ns, "00011000" after 310 ns,
51     "00001001" after 330 ns, "00001000" after 350 ns, "00101000" after 370 ns, "00101001" after 390 ns, "00000110" after 410 ns, "00011000"
52
53 END;

```



نتیجه شبیه سازی نشان می دهد که مدار طراحی شده تمامی ۱۰ تا جفت ورودی را پس از ضرب جمع کرده و در خروجی پس از آخرین جمع نشان می دهد.

البته همانطور که مشخص است در ورودی اول به علت تاخیر یک کلاک بین ورودی تا خروجی جمعی که در خروجی قرار می گیرد ۹ عدد را جمع می کند ولی این مشکل از ورودی های بعدی برطرف می شود.