14.1

|  |
| --- |
| library ieee;  use ieee.std\_logic\_1164.all;  entity fourteen\_one is  port (  a, cin: in std\_logic;  s, cout: out std\_logic  );  end fourteen\_one;  architecture rtl of fourteen\_one is  begin  s <= a xor cin;  cout <= a and cin;  end rtl; |

14.5

|  |
| --- |
| library ieee;  use ieee.std\_logic\_1164.all;  entity inc\_forloop is  generic(WIDTH: natural);  port(  a: in std\_logic\_vector(WIDTH-1 downto 0);  s: out std\_logic\_vector(WIDTH-1 downto 0);  cin: in std\_logic;  cout: out std\_logic  );  end inc\_forloop;  architecture inc\_arch of inc\_forloop is  signal tmp: std\_logic\_vector(WIDTH downto 0);    begin  process(a,tmp)  begin  tmp(0) <= cin;  for i in 1 to (WIDTH) loop  tmp(i) <= tmp(i-1) and a(i-1);  s(i-1) <= tmp(i-1) xor a(i-1);  end loop;  end process;  cout <= tmp(WIDTH);    end inc\_arch; |

14.7

|  |
| --- |
| library ieee;  use ieee.std\_logic\_1164.all;  entity inc\_norange is  port(  a: in std\_logic\_vector;  s: out std\_logic\_vector;  cin: in std\_logic;  cout: out std\_logic  );  end inc\_norange;  architecture inc\_arch of inc\_norange is  constant WIDTH: natural := a'length;  signal tmp: std\_logic\_vector(WIDTH downto 0);    begin  loop\_name: for i in 0 downto WIDTH-1 generate  s(i) <= tmp(i) xor a(i);  tmp(i+1) <= a(i) xor tmp(i);  end generate;  tmp(0) <= cin;  cout <= tmp(WIDTH);    end inc\_arch; |

14.6

|  |
| --- |
| library ieee;  use ieee.std\_logic\_1164.all;  entity inc\_clever is  generic(WIDTH: natural);  port(  a: in std\_logic\_vector;  s: out std\_logic\_vector;  cin: std\_logic;  cout: out std\_logic  );  end inc\_clever;  architecture inc\_arch of inc\_clever is  signal tmp: std\_logic\_vector(WIDTH downto 0);    begin  tmp(0) <= cin;  s <= a xor tmp(WIDTH-1 downto 0);  tmp(WIDTH downto 1) <= tmp(WIDTH-1 downto 0) and a;    end inc\_arch; |

14.2

|  |
| --- |
| library ieee;  use ieee.std\_logic\_1164.all;  entity inc\_generic is  generic(WIDTH: natural);  port(  a: in std\_logic\_vector(WIDTH-1 downto 0);  s: out std\_logic\_vector(WIDTH-1 downto 0);  cin: in std\_logic;  cout: out std\_logic  );  end inc\_generic;  architecture inc\_arch of inc\_generic is  signal tmp: std\_logic\_vector(WIDTH downto 0);    begin    loop\_name: for i in 0 downto WIDTH-1 generate  s(i) <= tmp(i) xor a(i);  tmp(i+1) <= a(i) xor tmp(i);  end generate;  tmp(0) <= cin;  cout <= tmp(WIDTH);    end inc\_arch; |

14.3

|  |
| --- |
| library ieee;  use ieee.std\_logic\_1164.all;  entity inc\_component is  generic(WIDTH: natural);  port(  a: in std\_logic\_vector(WIDTH-1 downto 0);  s: out std\_logic\_vector(WIDTH-1 downto 0);  cin: in std\_logic;  cout: out std\_logic  );  end inc\_component;  architecture inc\_arch of inc\_component is  signal tmp: std\_logic\_vector(WIDTH downto 0);    begin  tmp(0) <= cin;  loop\_name: for i in 1 to (WIDTH) generate  each\_module: entity work.fourteen\_one(rtl)  port map(a=>a(i-1),cin=>tmp(i-1),cout=>tmp(i),s=>s(i-1));  end generate;  cout <= tmp(WIDTH);    end inc\_arch; |

14.4

|  |
| --- |
| library ieee;  use ieee.std\_logic\_1164.all;  entity inc\_conditional is  generic(WIDTH: natural);  port(  a: in std\_logic\_vector(WIDTH-1 downto 0);  s: out std\_logic\_vector(WIDTH-1 downto 0);  cin: in std\_logic;  cout: out std\_logic  );  end inc\_conditional;  architecture inc\_arch of inc\_conditional is  signal tmp: std\_logic\_vector(WIDTH downto 0);    begin  loop\_name: for i in 0 downto WIDTH-1 generate  first\_bound:  if i = 0 generate  tmp(i) <= cin;  end generate;  s(i) <= tmp(i) xor a(i);  tmp(i+1) <= a(i) xor tmp(i);  last\_bound:  if i = WIDTH-1 generate  cout <= tmp(WIDTH);  end generate;  end generate;    end inc\_arch; |