VHDL Encoders

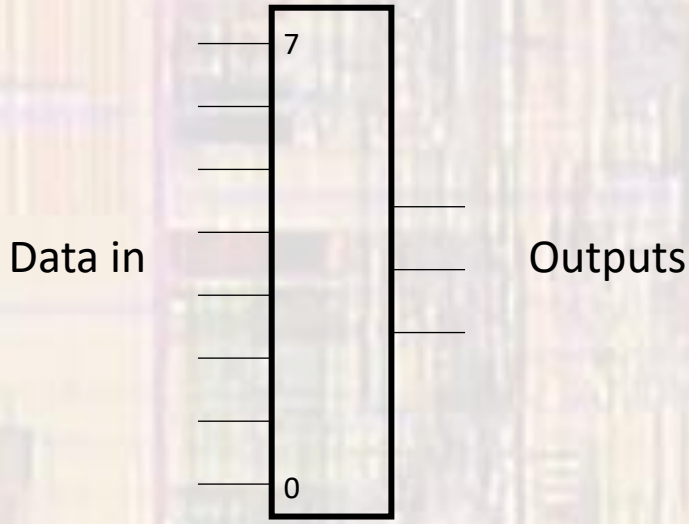
Last updated 1/7/25

VHDL Encoders

- An encoder creates a representation of an N input signal
  - N data inputs
  - E data outputs
  - N:E Encoder

N = 8

E = 3



VHDL Encoders

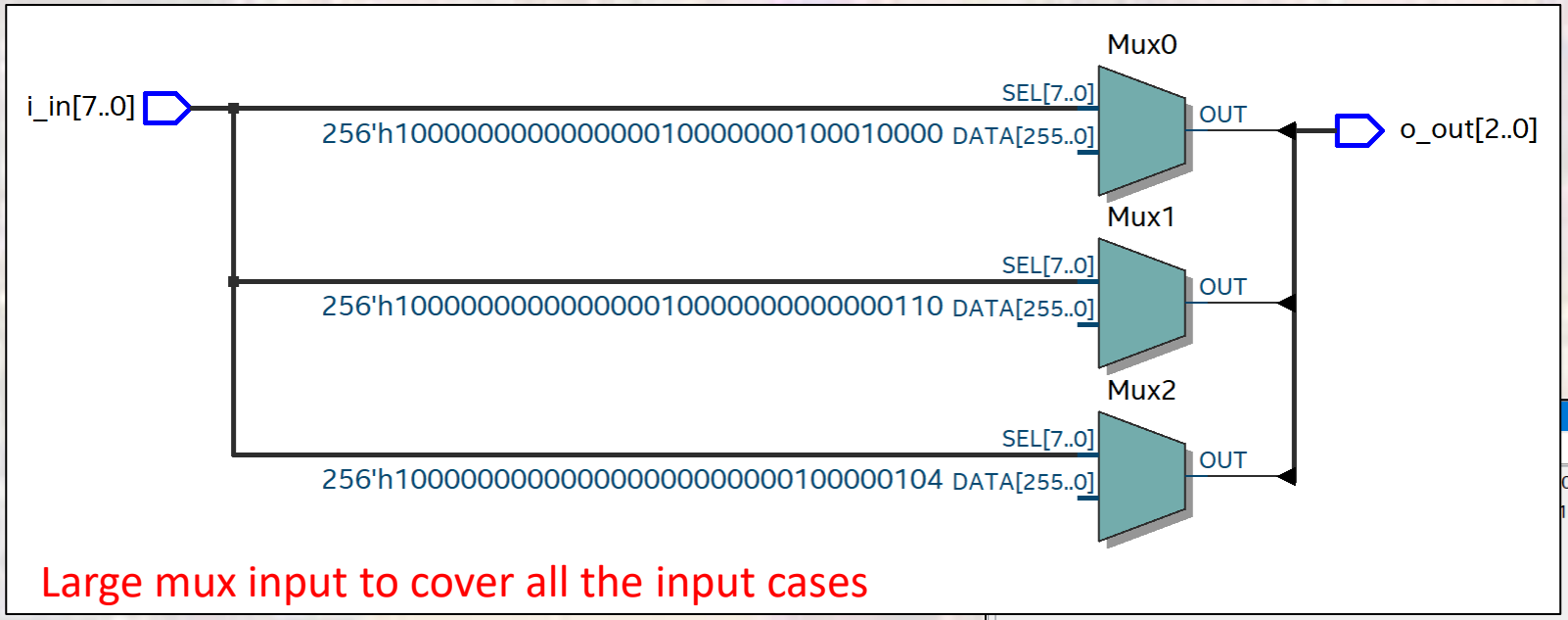
- Binary Encoder – Example – large solution

```
----------------------------------------
--
-- encoder_binary.vhdl
--
-- created 7/5/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- vhdl binary encoder - using with-select
-- brute-force - creates large solution
--
----------------------------------------
--
-- Inputs: in(7-0)
-- Outputs: out(2-0)
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity encoder_binary is
    port (
          i_in:    in std_logic_vector(7 downto 0);
          o_out:   out std_logic_vector(2 downto 0)
          );
end entity;
```

```
architecture behavioral of encoder_binary is

begin

   with i_in select o_out <=
                                "000" when "00000001",
                                "001" when "00000010",
                                "010" when "00000100",
                                "011" when "00001000",
                                "100" when "00010000",
                                "101" when "00100000",
                                "110" when "01000000",
                                "111" when "10000000",
                                "000" when others;

end behavioral;
```

VHDL Encoders

- Binary Encoder – Example – large solution



Large mux input to cover all the input cases

03:12 2025
18 SJ Lite Edition

Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	8
Total registers	0
Total pins	11
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

VHDL Encoders

- Binary Encoder – Example – special case solution

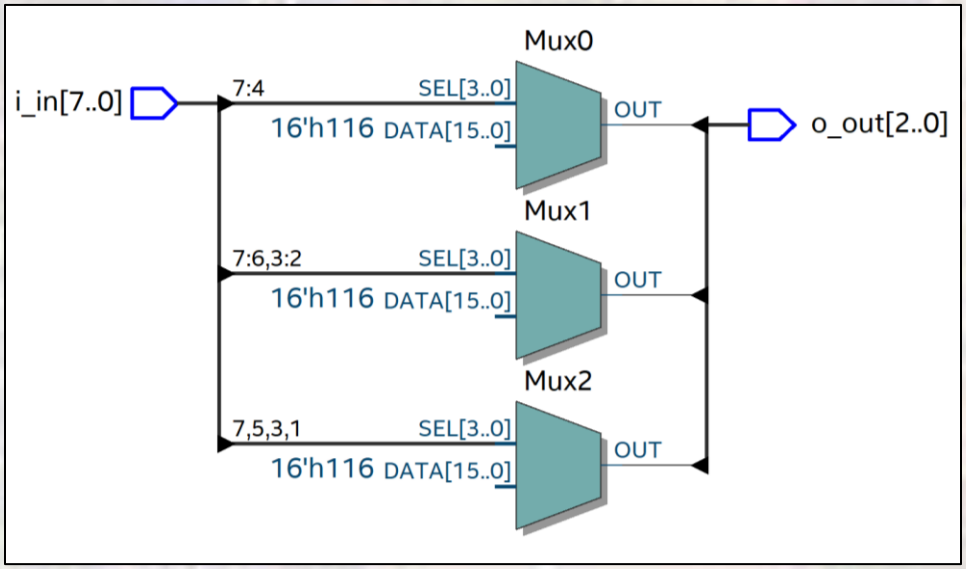
```
----------------------------------------
--
-- encoder_binary.vhdl
--
-- created 7/5/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- vhdl binary encoder - using with-select
-- brute-force - creates large solution without dont'care
--
----------------------------------------
--
-- Inputs: in(7-0)
-- Outputs: out(2-0)
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity encoder_binary is
    port (
          i_in:    in std_logic_vector(7 downto 0);
          o_out:   out std_logic_vector(2 downto 0)
          );
end entity;
```

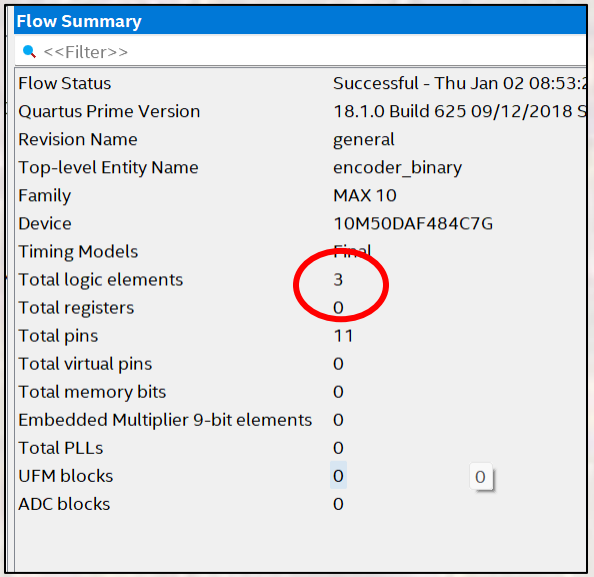
be very careful using the don't care in synthesis

```
architecture behavioral of encoder_binary is

begin

    with i_in select o_out <=
                                    "000" when "00000001",
                                    "001" when "00000010",
                                    "010" when "00000100",
                                    "011" when "00001000",
                                    "100" when "00010000",
                                    "101" when "00100000",
                                    "110" when "01000000",
                                    "111" when "10000000",
                                    "---" when others;
            -- special case to reduce logic

end behavioral;
```

VHDL Encoders

- Binary Encoder – Example – special case solution





Smaller mux input – fewer logic gates

VHDL Encoders

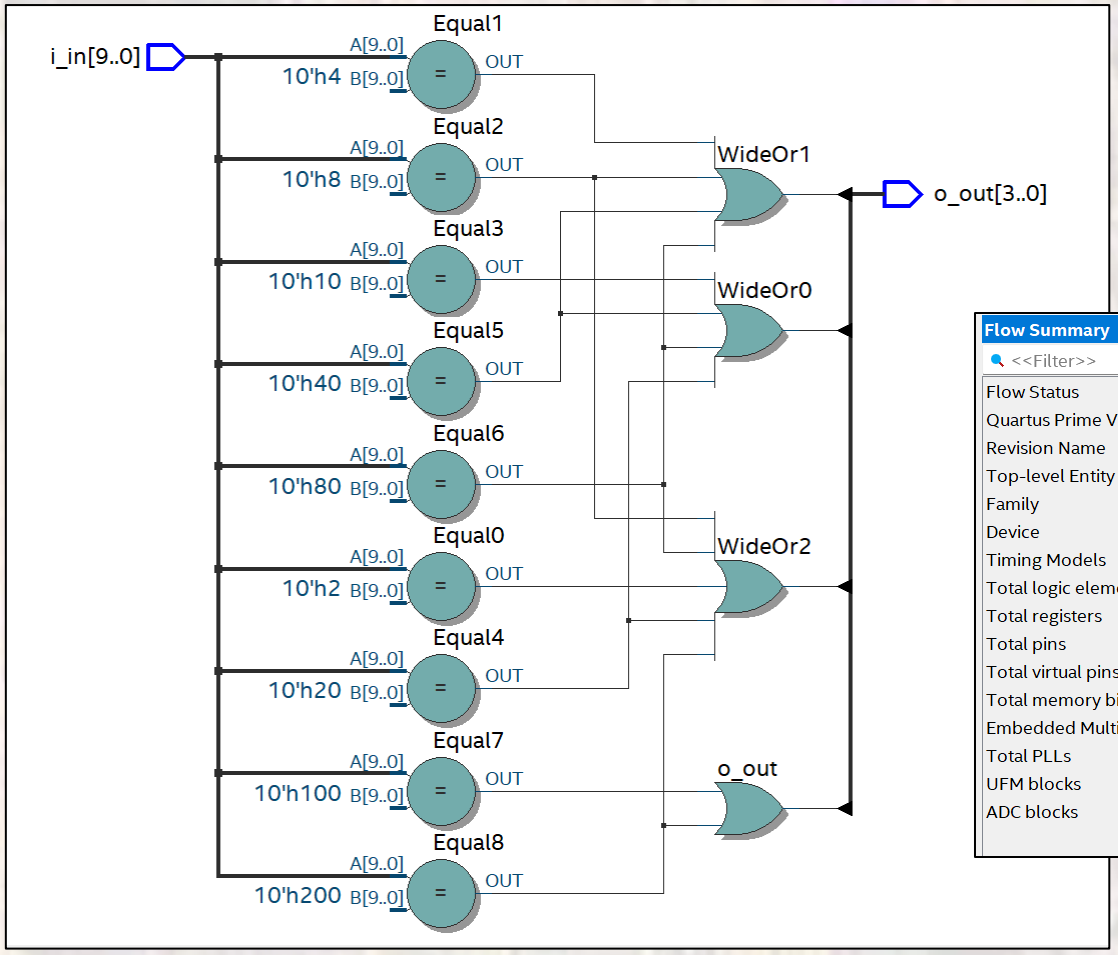
- Decimal to BCD Encoder

```
----------------------------------------
--
-- encoder_decimal_to_bcd.vhdl
--
-- created 7/5/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- vhdl decimal to bcd encoder - using with-select
-- brute-force
--
----------------------------------------
--
-- Inputs: in(7-0)
-- Outputs: out(2-0)
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity encoder_decimal_to_bcd is
   port (
         i_in:    in std_logic_vector(9 downto 0);
         o_out:   out std_logic_vector(3 downto 0)
         );
end entity;
```

```
architecture behavioral of encoder_decimal_to_bcd is

begin

   with i_in select o_out <=
                                 "0000" when "0000000001",
                                 "0001" when "0000000010",
                                 "0010" when "0000000100",
                                 "0011" when "0000001000",
                                 "0100" when "0000010000",
                                 "0101" when "0000100000",
                                 "0110" when "0001000000",
                                 "0111" when "0010000000",
                                 "1000" when "0100000000",
                                 "1001" when "1000000000",
                                 "0000" when others;

end behavioral;
```

VHDL Encoders

- Decimal to BCD Encoder



Flow Summary

<<Filter>>

Flow Status	Successful - Thu Jan 02 09:14:14 2025
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	general
Top-level Entity Name	encoder_decimal_to_bcd
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	14
Total registers	0
Total pins	14
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

VHDL Encoders

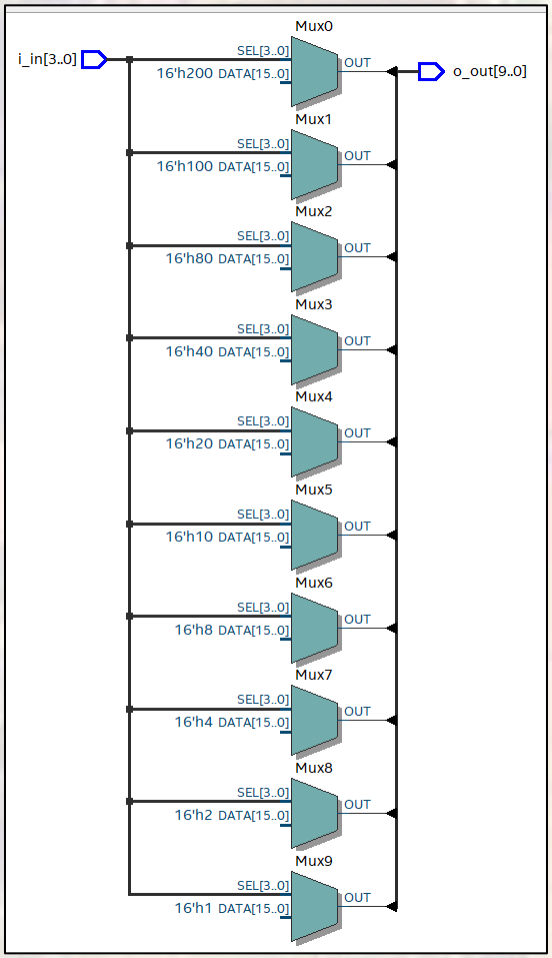
- BCD to Decimal Encoder

```
----------------------------------------
--
-- encoder_bcd_to_decimal.vhdl
--
-- created 7/5/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- vhdl bcd to decimal encoder - using with-select
-- brute-force
--
----------------------------------------
--
-- Inputs: in(9-0)
-- Outputs: out(3-0)
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity encoder_bcd_to_decimal is
   port (
         i_in:    in std_logic_vector(3 downto 0);
         o_out:   out std_logic_vector(9 downto 0)
         );
end entity;
```

```
architecture behavioral of encoder_bcd_to_decimal is

begin

    with i_in select o_out <=
                                "0000000001" when "0000",
                                "0000000010" when "0001",
                                "0000000100" when "0010",
                                "0000001000" when "0011",
                                "0000010000" when "0100",
                                "0000100000" when "0101",
                                "0001000000" when "0110",
                                "0010000000" when "0111",
                                "0100000000" when "1000",
                                "1000000000" when "1001",
                                "0000000000" when others;

end behavioral;
```

VHDL Encoders

- BCD to Decimal Encoder



Flow Summary

<<Filter>>

Flow Status	Successful - Thu Jan 02 10:04:13 2025
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	general
Top-level Entity Name	encoder_bcd_to_decimal
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	10
Total registers	0
Total pins	14
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

VHDL Encoders

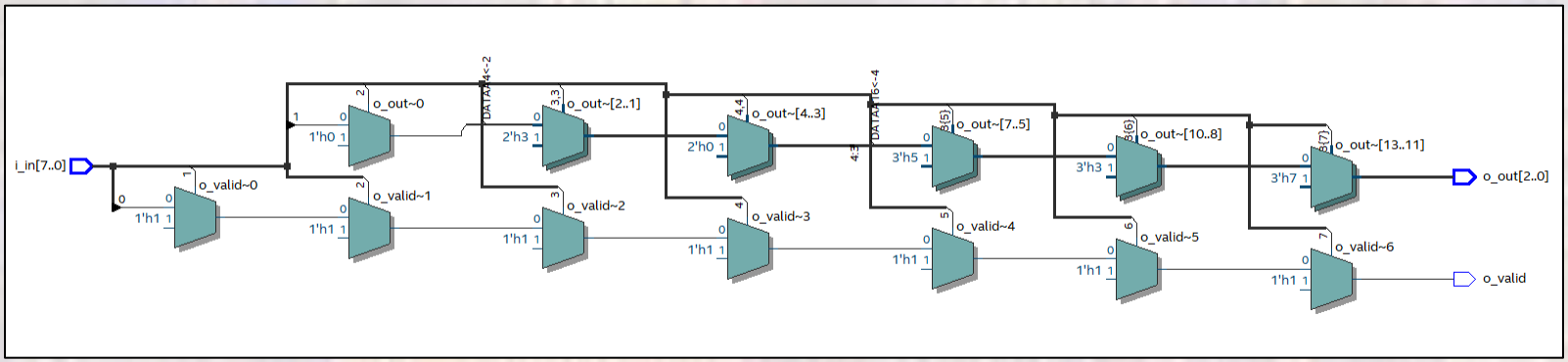
- Priority Encoder

```
----------------------------------------
--
-- encoder_priority.vhdl
--
-- created 7/5/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- vhdl priority encoder - using when-else
-- brute-force
--
----------------------------------------
--
-- Inputs: in(7-0)
-- Outputs: out(2-0)
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity encoder_priority is
   port (
         i_in:    in std_logic_vector(7 downto 0);
         o_out:   out std_logic_vector(2 downto 0);
         o_valid: out std_logic
         );
end entity;
```

```
architecture behavioral of encoder_priority is

begin

   o_out <= "111" when (i_in(7) = '1') else
            "110" when (i_in(6) = '1') else
            "101" when (i_in(5) = '1') else
            "100" when (i_in(4) = '1') else
            "011" when (i_in(3) = '1') else
            "010" when (i_in(2) = '1') else
            "001" when (i_in(1) = '1') else
            "000" when (i_in(0) = '1') else
            "000";

   o_valid <=  '1' when (i_in(7) = '1') else
               '1' when (i_in(6) = '1') else
               '1' when (i_in(5) = '1') else
               '1' when (i_in(4) = '1') else
               '1' when (i_in(3) = '1') else
               '1' when (i_in(2) = '1') else
               '1' when (i_in(1) = '1') else
               '1' when (i_in(0) = '1') else
               '0';

end behavioral;
```

VHDL Encoders

- Priority Encoder



Flow Summary

<<Filter>>

Flow Status	Successful - Thu Jan 02 11:35:40 2025
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	general
Top-level Entity Name	encoder_priority
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	7
Total registers	0
Total pins	12
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

VHDL Encoders

- Encoders can have ambiguous or incorrect results
  - More than 1 input high (Binary, BCD)
  - No inputs high
- Special care must be taken to account for these situations
  - In the encode itself
  - In the logic driving the encoder
  - In the logic following the encoder