VHDL Synthesis
Simple Constructs

Last updated 1/7/25

VHDL Synthesis - Simple Constructs

VHDL is not a programming language
It is a hardware description language

VHDL Synthesis - Simple Constructs

- Create a hardware description for an electronic version of a binary 4 input combination lock

1/0 inputs

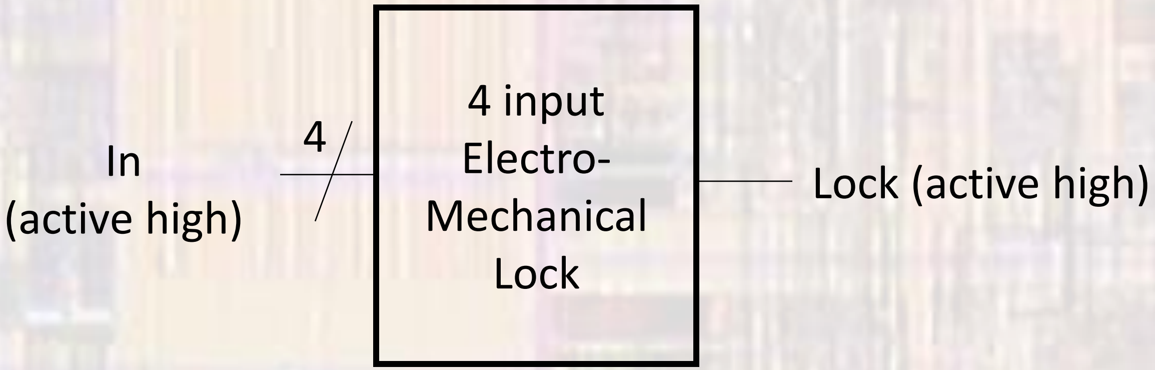
VHDL Synthesis - Simple Constructs

- Create a hardware description for an electronic version of a binary 4 input combination lock
  - Design Decisions
    - Default state (locked or unlocked)?
    - Input signal activation condition (high or low)?
    - Output signal activation condition (lock or unlock)?
  - Additional use case information
    - Target customer is private and commercial gym locker rooms
    - Normally the lockers will be unlocked so customers can place items in the locker and then lock it

    - Default to unlocked
    - Output signal will be active high to lock

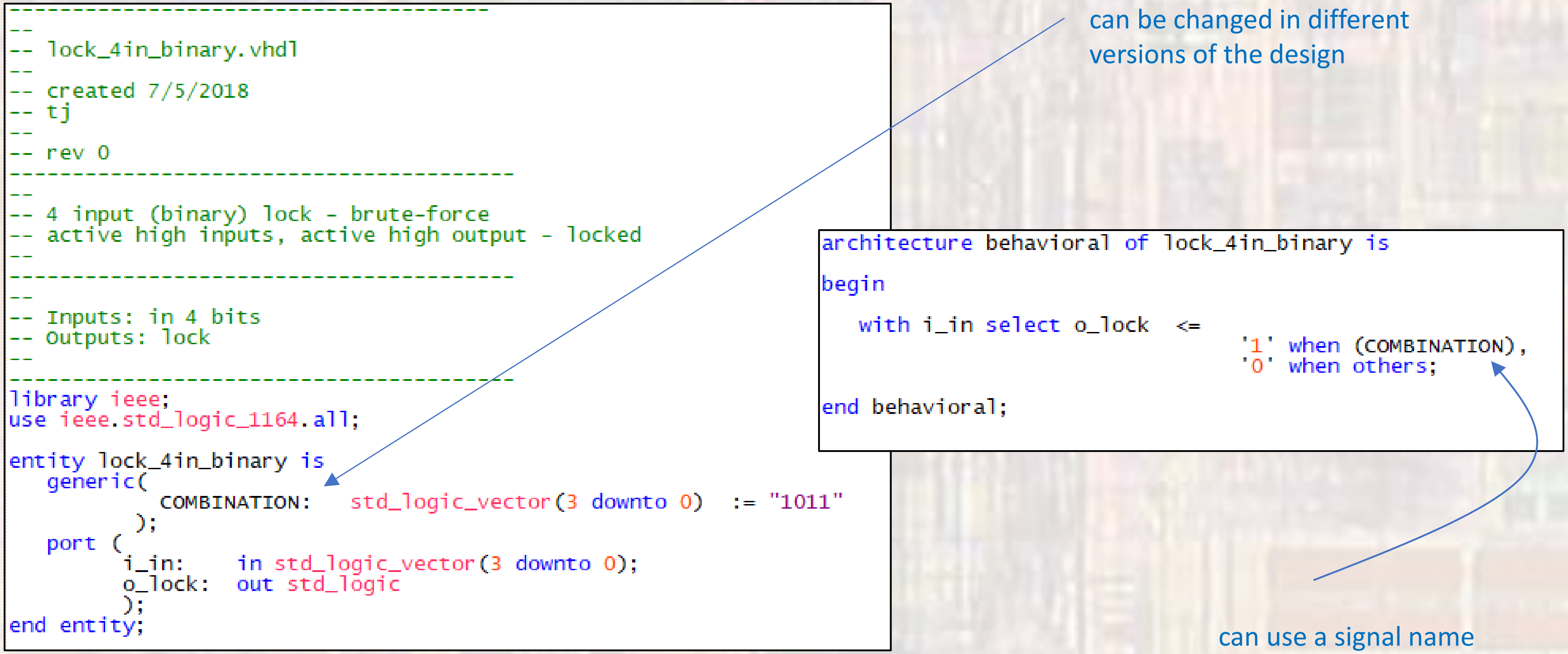
VHDL Synthesis - Simple Constructs

- Create a hardware description for an electronic version of a binary 4 input combination lock
  - Block Diagram



VHDL Synthesis - Simple Constructs

- Create a hardware description for an electronic version of a binary 4 input combination lock



```vhdl
--------------------------------------
--
-- lock_4in_binary.vhdl
--
-- created 7/5/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- 4 input (binary) lock - brute-force
-- active high inputs, active high output - locked
--
----------------------------------------
--
-- Inputs: in 4 bits
-- Outputs: lock
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity lock_4in_binary is
   generic(
         COMBINATION:   std_logic_vector(3 downto 0)   := "1011"
         );
   port (
         i_in:     in std_logic_vector(3 downto 0);
         o_lock:   out std_logic
         );
end entity;
```

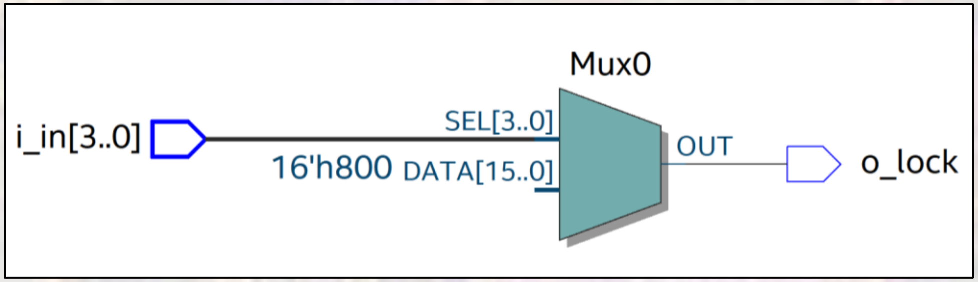
```vhdl
architecture behavioral of lock_4in_binary is

begin

   with i_in select o_lock   <=
                                  '1' when (COMBINATION),
                                  '0' when others;

end behavioral;
```

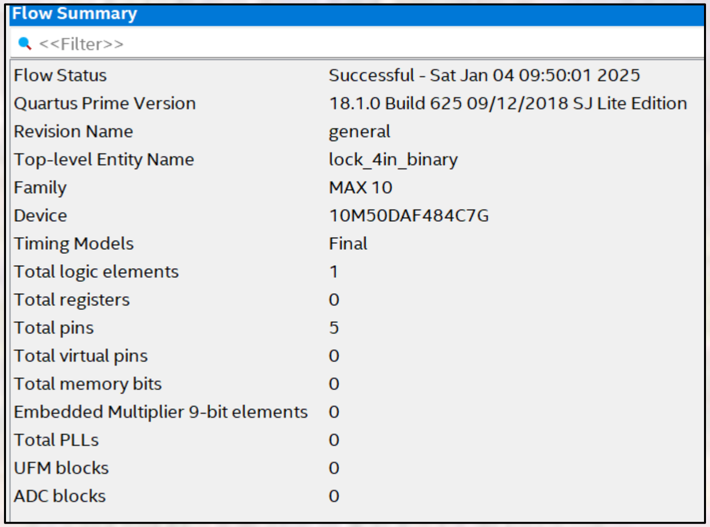
Use generics for values that can be changed in different versions of the design

can use a signal name instead of an explicit vector

VHDL Synthesis - Simple Constructs

- Create a hardware description for an electronic version of a binary 4 input combination lock

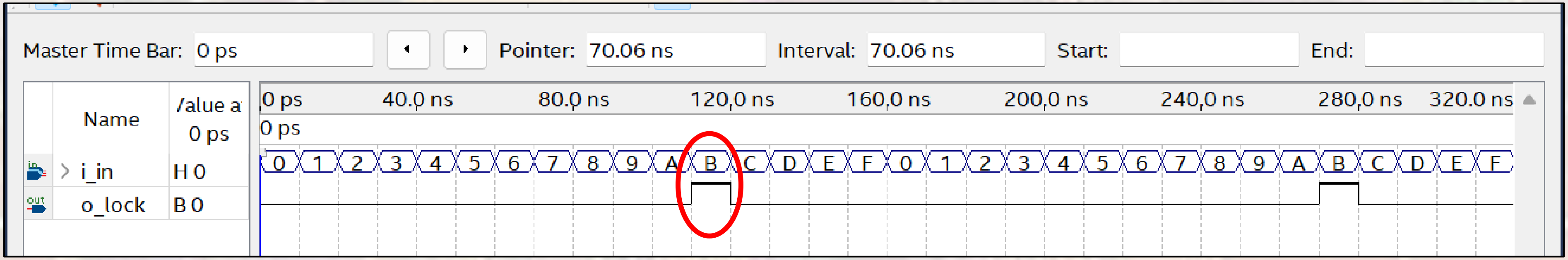




Flow Summary

<<Filter>>

Flow Status	Successful - Sat Jan 04 09:50:01 2025
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	general
Top-level Entity Name	lock_4in_binary
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	1
Total registers	0
Total pins	5
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

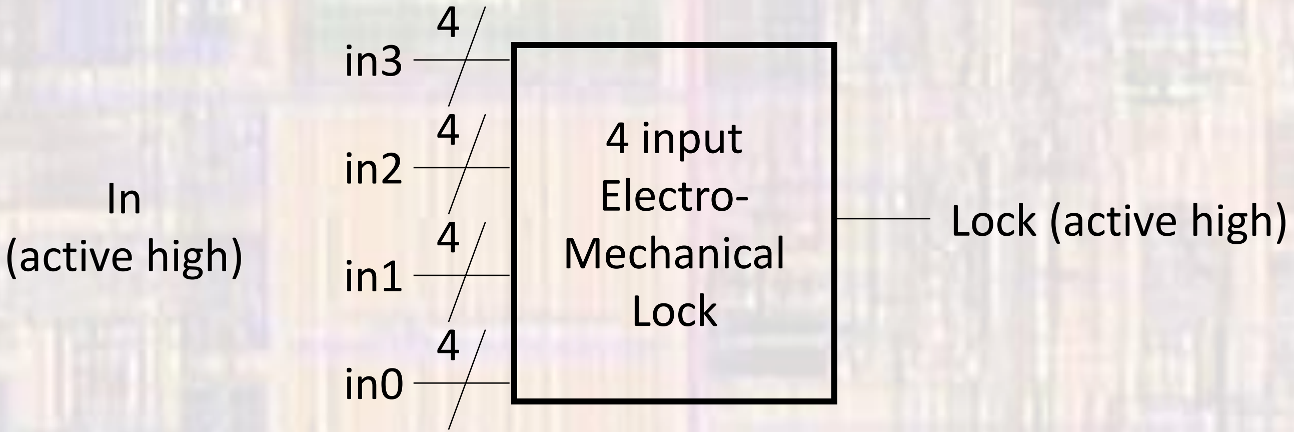


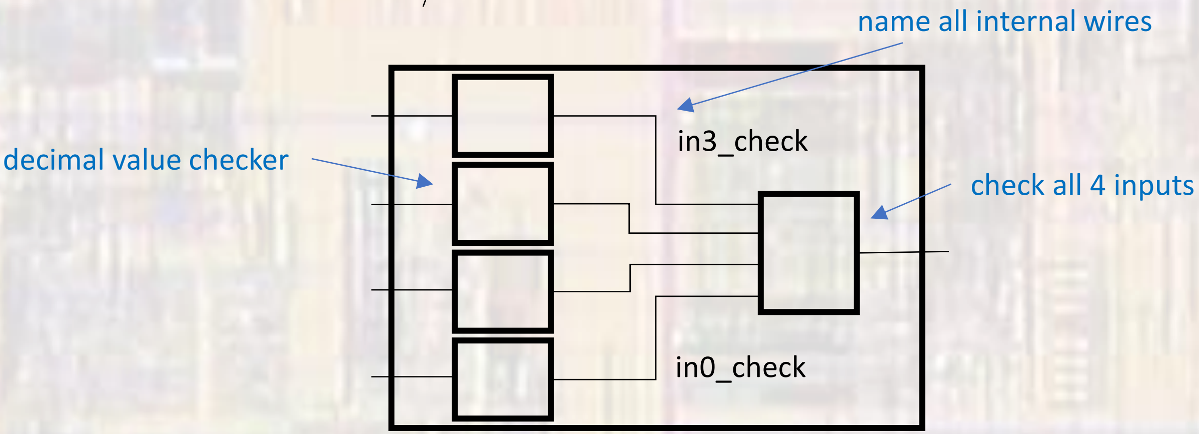
Master Time Bar: 0 ps Pointer: 70.06 ns Interval: 70.06 ns Start: End:

Name	Value at 0 ps
i_in	H 0
o_lock	B 0

VHDL Synthesis - Simple Constructs

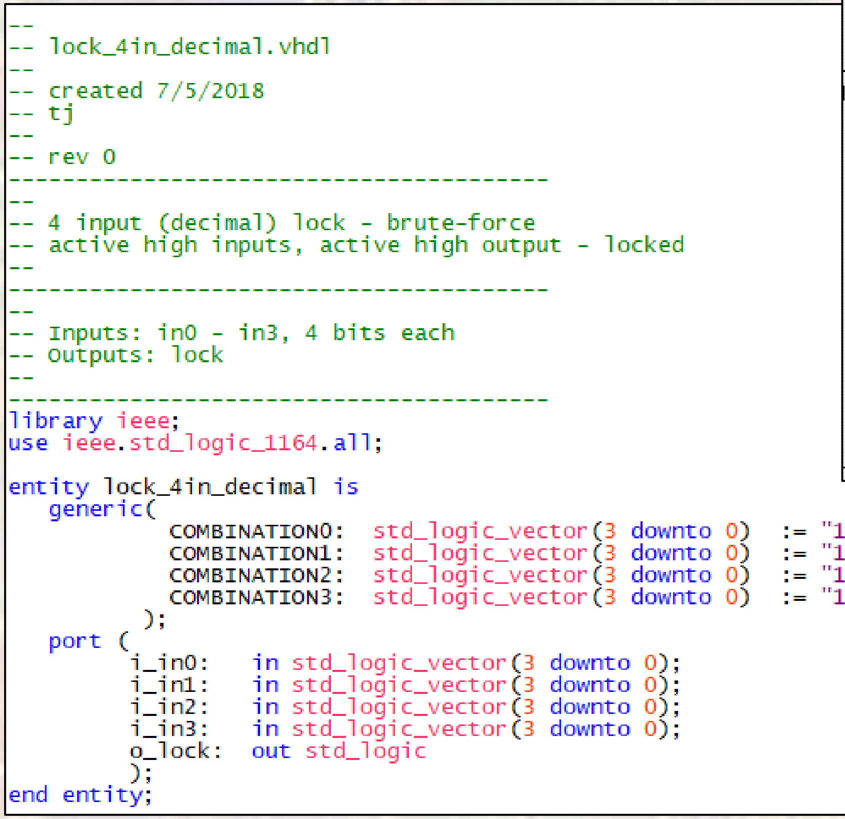
- Create a hardware description for an electronic version of a decimal 4 input combination lock
  - Block Diagram



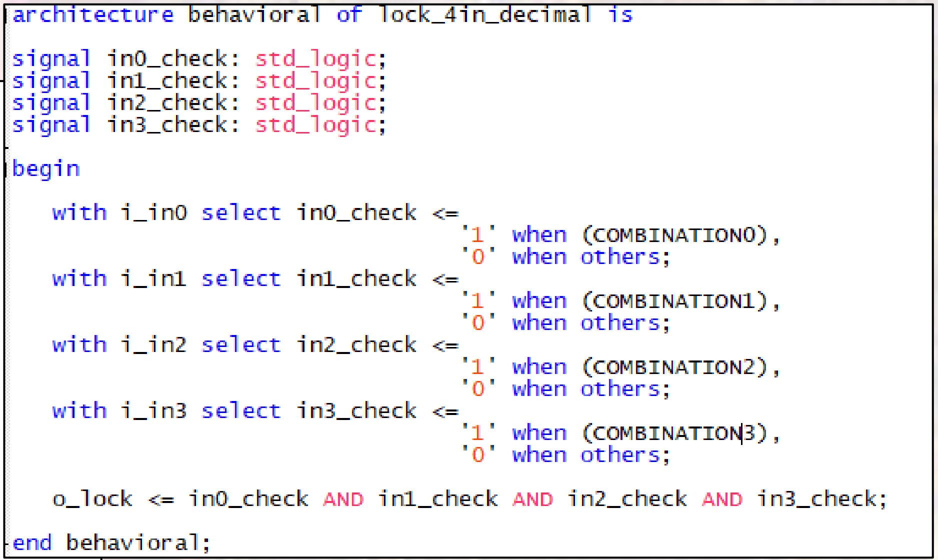


VHDL Synthesis - Simple Constructs

- Create a hardware description for an electronic version of a decimal 4 input combination lock



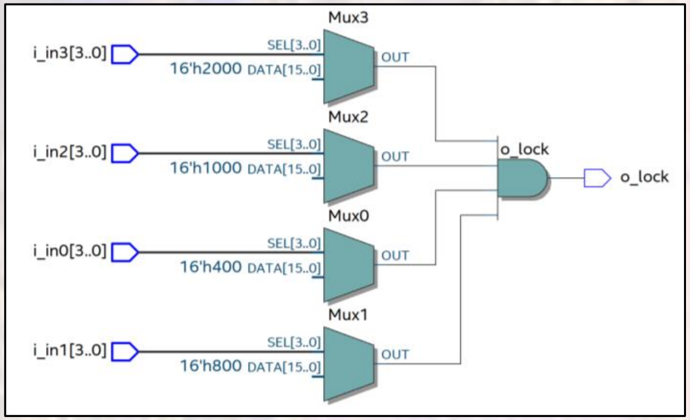
```
--
-- lock_4in_decimal.vhdl
--
-- created 7/5/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- 4 input (decimal) lock - brute-force
-- active high inputs, active high output - locked
--
----------------------------------------
--
-- Inputs: in0 - in3, 4 bits each
-- Outputs: lock
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity lock_4in_decimal is
   generic(
            COMBINATION0:  std_logic_vector(3 downto 0)   := "1010";
            COMBINATION1:  std_logic_vector(3 downto 0)   := "1011";
            COMBINATION2:  std_logic_vector(3 downto 0)   := "1100";
            COMBINATION3:  std_logic_vector(3 downto 0)   := "1101"
          );
   port (
         i_in0:    in std_logic_vector(3 downto 0);
         i_in1:    in std_logic_vector(3 downto 0);
         i_in2:    in std_logic_vector(3 downto 0);
         i_in3:    in std_logic_vector(3 downto 0);
         o_lock:   out std_logic
         );
end entity;
```

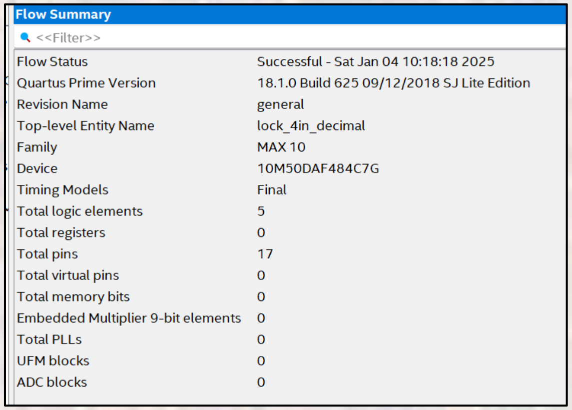


```
architecture behavioral of lock_4in_decimal is

signal in0_check: std_logic;
signal in1_check: std_logic;
signal in2_check: std_logic;
signal in3_check: std_logic;

begin

   with i_in0 select in0_check <=
                                 '1' when (COMBINATION0),
                                 '0' when others;
   with i_in1 select in1_check <=
                                 '1' when (COMBINATION1),
                                 '0' when others;
   with i_in2 select in2_check <=
                                 '1' when (COMBINATION2),
                                 '0' when others;
   with i_in3 select in3_check <=
                                 '1' when (COMBINATION3),
                                 '0' when others;

   o_lock <= in0_check AND in1_check AND in2_check AND in3_check;

end behavioral;
```

VHDL Synthesis - Simple Constructs

- Create a hardware description for an electronic version of a decimal 4 input combination lock

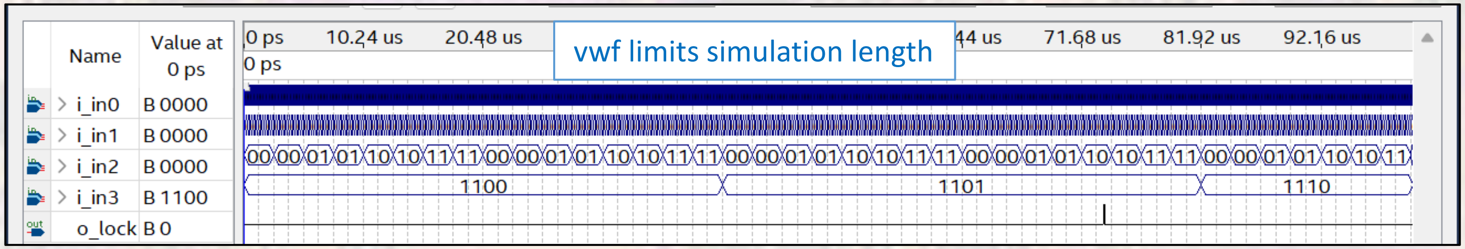




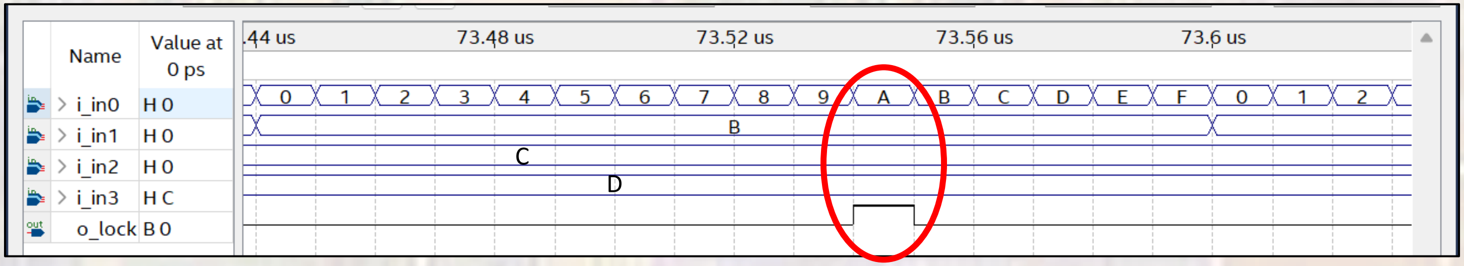
Flow Summary

<<Filter>>

Flow Status	Successful - Sat Jan 04 10:18:18 2025
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	general
Top-level Entity Name	lock_4in_decimal
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	5
Total registers	0
Total pins	17
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0



vwf limits simulation length

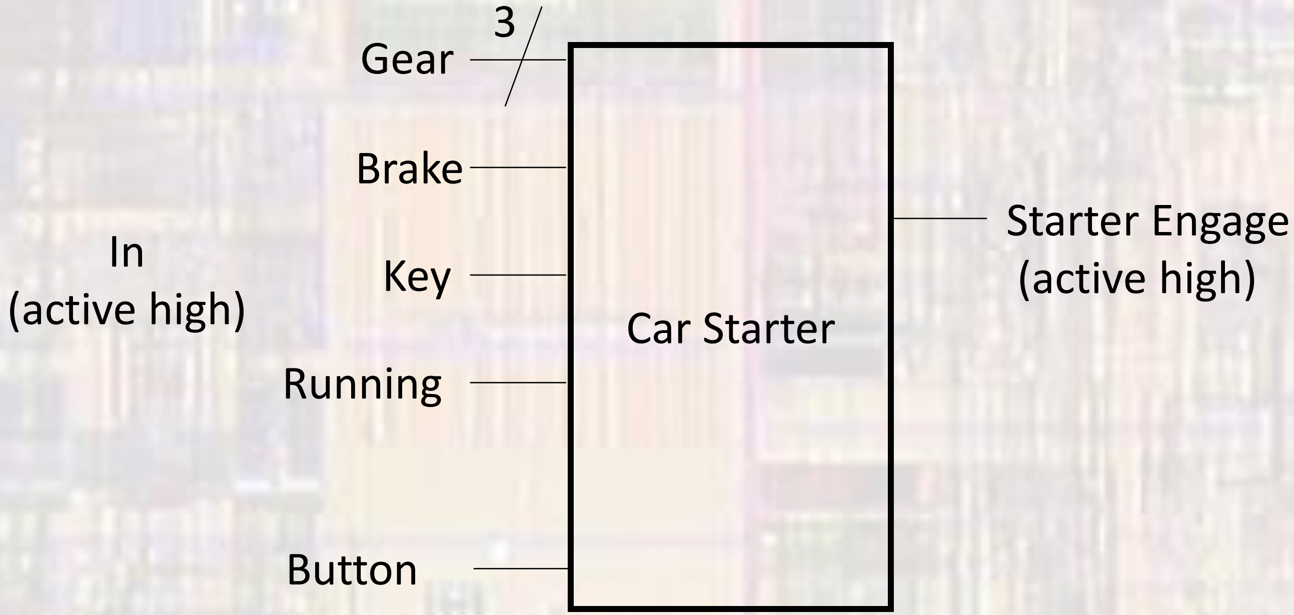


VHDL Synthesis - Simple Constructs

- Create a hardware description for a car starter mechanism
  - Start conditions
    - Brake on
    - Key in range
    - Not running
    - In park
    - Button pushed
    - All inputs active high
    - Output active high

VHDL Synthesis - Simple Constructs

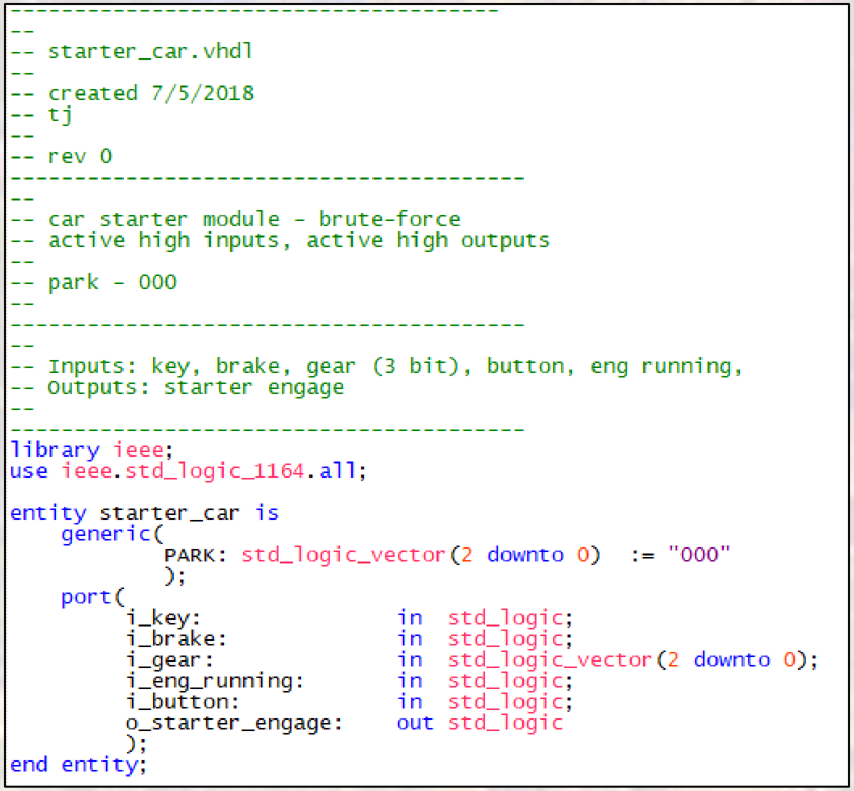
- Create a hardware description for a car starter mechanism
  - Block Diagram



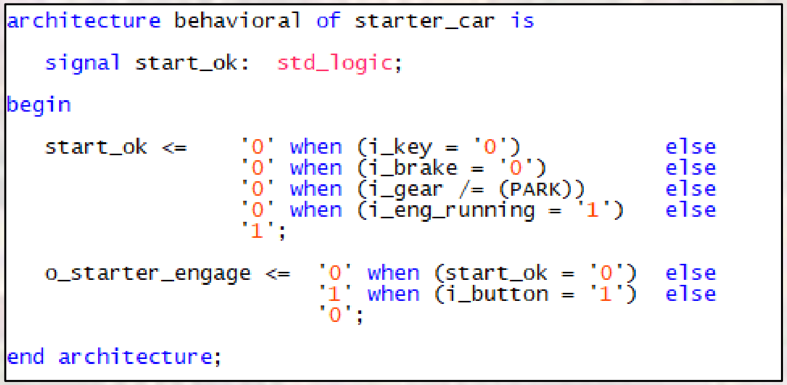
Gear encoding: 000 – Park, 001 – Neutral, 100 – Reverse, 010 - L1, 011 – L2

VHDL Synthesis - Simple Constructs

- Create a hardware description for a car starter mechanism



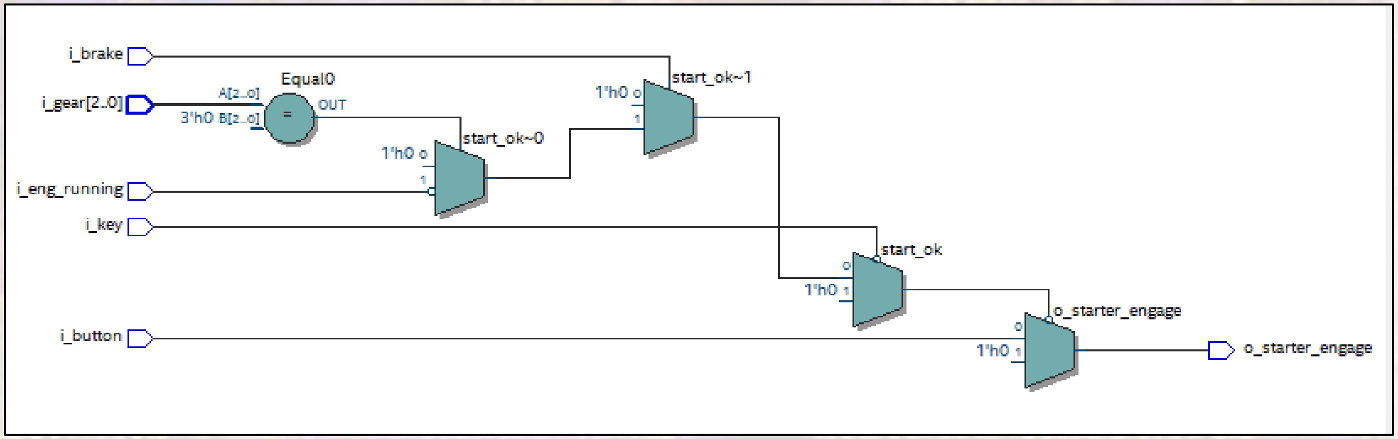
```
--------------------------------------
--
-- starter_car.vhdl
--
-- created 7/5/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- car starter module - brute-force
-- active high inputs, active high outputs
--
-- park - 000
--
----------------------------------------
--
-- Inputs: key, brake, gear (3 bit), button, eng running,
-- Outputs: starter engage
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity starter_car is
    generic(
            PARK: std_logic_vector(2 downto 0)  := "000"
            );
    port(
         i_key:                 in  std_logic;
         i_brake:               in  std_logic;
         i_gear:                in  std_logic_vector(2 downto 0);
         i_eng_running:         in  std_logic;
         i_button:              in  std_logic;
         o_starter_engage:      out std_logic
         );
end entity;
```

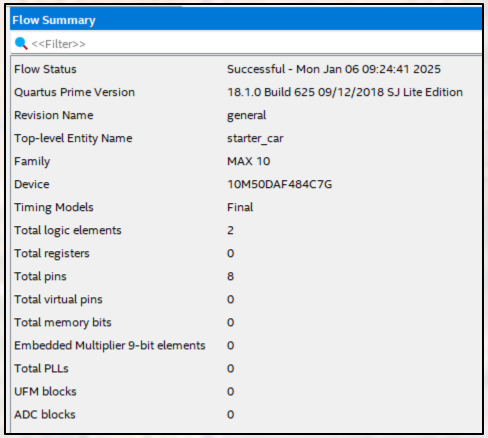


```
architecture behavioral of starter_car is

   signal start_ok:  std_logic;

begin

   start_ok <=       '0' when (i_key = '0')             else
                     '0' when (i_brake = '0')           else
                     '0' when (i_gear /= (PARK))        else
                     '0' when (i_eng_running = '1')     else
                     '1';

   o_starter_engage <=     '0' when (start_ok = '0')    else
                           '1' when (i_button = '1')    else
                           '0';

end architecture;
```

VHDL Synthesis - Simple Constructs

- Create a hardware description for a car starter mechanism



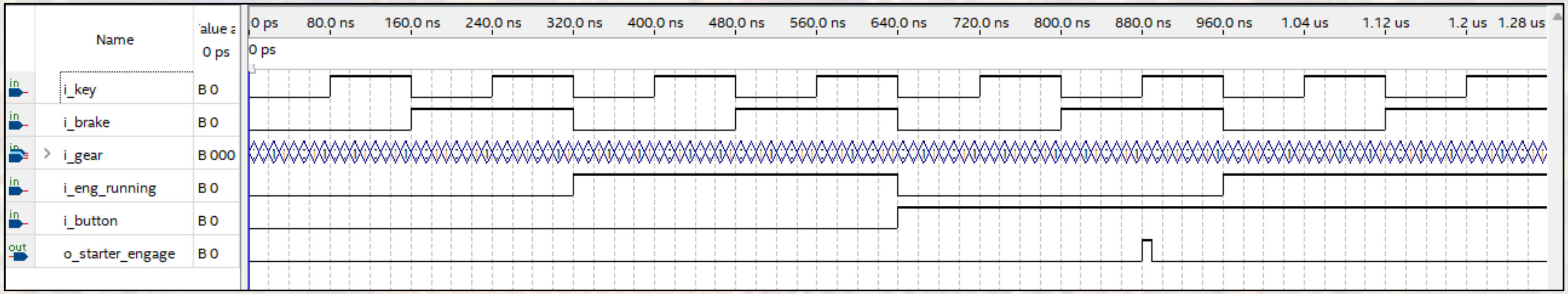

Flow Summary

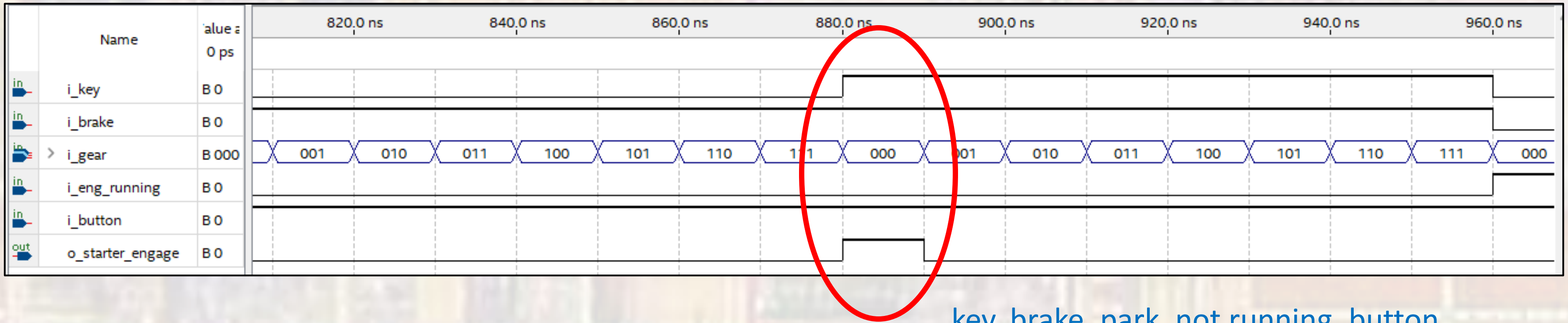
<<Filter>>

Flow Status	Successful - Mon Jan 06 09:24:41 2025
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	general
Top-level Entity Name	starter_car
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	2
Total registers	0
Total pins	8
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

VHDL Synthesis - Simple Constructs

- Create a hardware description for a car starter mechanism





key, brake, park, not running, button