XILINX CELL LIBRARY USED IN THE SP700 DESIGN

William Bricken
May 2003

The SP700 design uses the cell library provided by Xilinx for specifying designs for the Virtex-II parts using VHDL. It consists of 375 cell elements. Combinational logic is expressed in the semantics of conventional Boolean logic, temporal FF logic uses a nine-valued logic that includes many signal error conditions. Library cells include

-- all varieties of two and three input logic functions
-- selected logic functions up to eight inputs
-- selected four, five and six input functions with specific structure
-- many FF varieties in various combinations of
  input/output polarity
  set/reset control
  enable logic
-- a few large specialized functions
-- a few functional specifications, mainly adder components
-- several multiple-output varieties of the above

Three listings of the cell library are attached below:

I. SELECTED VHDL CELL LIBRARY SPECIFICATIONS
II. SP700 CELL LIBRARY TRANSFORMATION TABLE FOR ILOC
III. SP700 CELL LIBRARY TYPE SUMMARY

SELECTED VHDL CELL LIBRARY SPECIFICATIONS

These VHDL entity specifications are selected from a dozen large files (several hundred pages), with many semantic redundancies. They include standard logic functions, FFs, larger functions, and what appear to be Virtex specific functions.

There were only three specific parsing problems associated with converting VHDL to ILOC:

1) VHDL architecture specifications were parsed by hand (I did not write a behavioral VHDL to ILOC parser), so that semantically complex entities could be separated. Thus, some complex VHDL behavioral constructs eliminated some entities from being parsed. See F985.

2) FF logic uses a nine-valued logic. There is no equivalent to (extra)logical "X" or "U" or "Z" currently in ILOC. Thus, these behaviors were eliminated from ILOC FF semantics using a dummy variable "*-*". See any VHDL FF specification, such as F601. See F924 for (extra)logical "U".

3) I was unclear how to convert dynamic VHDL signal inversion, so I finessed this using a dummy variable "*INV*". See F774, particularly "when 11 => Q <= not Q;".

A specific case when (extra)logical values become relevant is in FUNC_565, which has this semantics:

```
            ((D0 and not A) or (D1 and A) or (D0 and D1))
```

In a Boolean logic, the clause "(D0 and D1)" is redundant. In a three-valued logic, it is not.

SP700 CELL LIBRARY TRANSFORMATION TABLE FOR ILOC

Each VHDL entity specification is converted into an ILOC internal form using a pattern look-up table. Entries in this table show the cell library names and their associated ILOC parens form. The logic function is named in a comment. Thus, for example:

```
            ((F202 F222 L202)                 ;2nor
             '(N01 (H01 H02) )   )
```

identifies a set of cell library names {F202, F222, L202} that all have the same logical semantics, that of two-input NOR. The label "N01" names the cell output and the labels "H01" and "H02" name the cell inputs. The form of 2NOR in parens notation is "(H01 H02)", while the form of a logic equation that connects input to output is

```
            '(<output-name> <parens-form-with-input-names>)
```

The indentation of the table is solely for readability.

SP700 CELL LIBRARY TYPE SUMMARY

The cell type summary is used by ILOC for type checking. Each cell is identified by its VHDL name and by its type and arity. Thus, for example

```
            (F202  '(NOR 2))
```

identifies the type of the cell named "F202" as a logical NOR function with two inputs.

This listing perhaps gives the clearest overview of the constituent structure of the Xilinx cell library.

I. SELECTED VHDL CELL LIBRARY SPECIFICATIONS

```
----- CELL F091 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F091 is
    port(
      N01 : out std_logic ;
      N02 : out std_logic );
end F091;

architecture A of F091 is
begin
   N01 <= '1';
   N02 <= '0';
end A;


----- CELL F101 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F101 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic );
end F101;

architecture A of F101 is
  begin
    P_F101: N01 <= not(H01) ;
  end A;


----- CELL F202 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F202 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic );
end F202;

architecture A of F202 is
begin
  P_F202: N01 <= H01 nor H02 ;
end A;


----- CELL F204 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F204 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic );
end F204;

architecture A of F204 is
begin
  P_F204: N01 <= not(H01 or H02 or H03 or H04) ;
end A;
```

```
----- CELL F212 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F212 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic );
end F212;

architecture A of F212 is
begin
  P_F212: N01 <= H01 or H02 ;
end A;


----- CELL F303 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F303 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic );
end F303;

architecture A of F303 is
begin
  P_F303: N01 <= not(H01 and H02 and H03) ;
end A;


----- CELL F313 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F313 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic );
end F313;

architecture A of F313 is
begin
  P_F313: N01 <= H01 and H02 and H03 ;
end A;


----- CELL F517 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F517 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic );
end F517;

architecture A of F517 is
begin
  P_F517: N01 <= not(H01 xor H02 xor H03);
end A;
```

```
----- CELL F401 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F401 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic );
end F401;

architecture A of F401 is
begin
  P_F401: N01 <= H01 and (H02 xor H03) ;
end A;


----- CELL F422 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F422 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic );
end F422;

architecture A of F422 is
begin
  P_F422: N01 <= not(H01 or H02 or (H03 and H04)) ;
end A;


----- CELL F424 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F424 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic );
end F424;

architecture A of F424 is
begin
  P_F424: N01 <= (H01 and H02) nor (H03 and H04) ;
end A;


----- CELL F432NF -----
library IEEE; use IEEE.std_logic_1164.all;

entity F432NF is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic );
end F432NF;
```

```
architecture A of F432NF is
begin
  P_F432NF: N01 <= not((not H01) and H02 and ((not H03) or (not H04))) ;
end A;


----- CELL F433NE -----
library IEEE; use IEEE.std_logic_1164.all;

entity F433NE is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic );
end F433NE;

architecture A of F433NE is
begin
  P_F433NE: N01 <= H01 nand ((not H02) or H03 or H04) ;
end A;


----- CELL F436 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F436 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic ;
      H05 : in  std_logic ;
      H06 : in  std_logic );
end F436;

architecture A of F436 is
begin
  P_F436: N01 <= (H01 or H02 or H03) nand (H04 or H05 or H06) ;
end A;


----- CELL F438 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F438 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic ;
      H05 : in  std_logic ;
      H06 : in  std_logic );
end F438;

architecture A of F438 is
begin
  P_F438: N01 <= not((H01 or H02) and (H03 or H04) and (H05 or H06)) ;
end A;
```

```
----- CELL F462 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F462 is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic ;
      H05 : in  std_logic ;
      H06 : in  std_logic );
end F462;

architecture A of F462 is
begin
  P_F462: N01 <= ((H01 and H02 and H03) or (H04 and H05)) nor H06 ;
end A;


----- CELL F521 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F521 is
    port(
      N01 : out std_logic ;  -- S
      N02 : out std_logic ;  -- COUT
      H01 : in  std_logic ;  -- A
      H02 : in  std_logic ;  -- B
      H03 : in  std_logic ); -- CIN
end F521;

architecture A of F521 is
begin
   P_F521 : process (H01,H02,H03)
     begin
       N01 <= H01 xor H02 xor H03 ;
       N02 <= (H01 and H02) or (H03 and (H02 or H01)) ;
     end process P_F521 ;
end A;


----- CELL F523 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F523 is
  port(
      N01 : out std_logic ;
      N02 : out std_logic ;
      N03 : out std_logic ;
      N04 : out std_logic ;
      N05 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic ;
      H05 : in  std_logic ;
      H06 : in  std_logic ;
      H07 : in  std_logic ;
      H08 : in  std_logic ;
      H09 : in  std_logic );
end F523;
```

```
architecture A of F523 is

use IEEE.std_logic_arith.all;

begin

  P_F523 : process (H01,H02,H03,H04,H05,H06,H07,H08,H09)
      variable A, B, S : unsigned(5 downto 0) ;
    begin
      A(0) := H09 ;
      A(1) := H01 ;
      A(2) := H03 ;
      A(3) := H05 ;
      A(4) := H07 ;
      A(5) := '0' ;

      B(0) := '1' ;
      B(1) := H02 ;
      B(2) := H04 ;
      B(3) := H06 ;
      B(4) := H08 ;
      B(5) := '0' ;

      S := A + B ;

      N01 <= S(1) ;
      N02 <= S(2) ;
      N03 <= S(3) ;
      N04 <= S(4) ;
      N05 <= S(5) ;
    end process P_F523 ;

end A ;


----- CELL F528 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F528 is
    port(
      N01 : out std_logic ;  -- S
      N02 : out std_logic ;  -- COUT
      H01 : in  std_logic ;  -- A
      H02 : in  std_logic ;  -- B
      H03 : in  std_logic ); -- CIN
end F528;

architecture A of F528 is
begin
   P_F528 : process (H01,H02,H03)
     begin
       N01 <= H01 xor H02 xor H03 ;
       N02 <= (H01 and H02) or (H03 and (H02 or H01)) ;
     end process P_F528 ;
end A;
```

```
----- CELL F555 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F555 is
    port(
      N01 : out std_logic ;
      N02 : out std_logic ;
      N03 : out std_logic ;
      N04 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic ;
      H05 : in  std_logic ;
      H06 : in  std_logic ;
      H07 : in  std_logic ;
      H08 : in  std_logic ;
      H09 : in  std_logic );
end F555;

architecture A of F555 is
  use work.cb10vx_functions.all ;
begin
  P_F555_0: N01 <= not (FUNC_565(D0 => H01 , D1 => H02, A => H09) ) ;
  P_F555_1: N02 <= not (FUNC_565(D0 => H03 , D1 => H04, A => H09) ) ;
  P_F555_2: N03 <= not (FUNC_565(D0 => H05 , D1 => H06, A => H09) ) ;
  P_F555_3: N04 <= not (FUNC_565(D0 => H07 , D1 => H08, A => H09) ) ;
end A;


----- CELL F601 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F601 is
    port(
      N01 : out std_logic ;
      N02 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic );
end F601;

architecture A of F601 is
  signal Q : std_logic ;
begin
  P_F601: process(H01,H02)
  begin
    if H02 = '1' then
      Q <= H01 ;
   -- pragma translate_off
    elsif H02 /= '0' then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F601 ;
    N01 <= Q ;
    N02 <= not Q ;
end A;
```

```
----- CELL F605 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F605 is
    port(
      N01 : out std_logic ;
      N02 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic );
end F605;

architecture A of F605 is
  signal Q : std_logic ;
begin
  P_F605: process(H01,H02,H03)
  begin
    if H03 = '0' then
      Q <= '0' ;
   -- pragma translate_off
    elsif H03 /= '1' then
      Q <= 'X' ;
   -- pragma translate_on
    elsif H02 = '0' then
      Q <= H01 ;
   -- pragma translate_off
    elsif H02 /= '1' then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F605 ;
    N01 <= Q ;
    N02 <= not Q ;
end A;


----- CELL F601SQ -----
library IEEE; use IEEE.std_logic_1164.all;

entity F601SQ is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic );
end F601SQ;

architecture A of F601SQ is
  use work.cb10vx_functions.all ;
  signal Q : std_logic ;
begin
  P_F601SQ: process(H01,H02,H03,H04)
  begin
    if H03 = '1' then
      Q <= FUNC_565(D0 => H01, D1 => H02, A => H04) ;
   -- pragma translate_off
    elsif H03 /= '0' then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F601SQ ;
    N01 <= Q ;
end A;
```

```
----- CELL F603SQ -----
library IEEE; use IEEE.std_logic_1164.all;

entity F603SQ is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic ;
      H05 : in  std_logic );
end F603SQ;

architecture A of F603SQ is
  use work.cb10vx_functions.all ;
  signal Q : std_logic ;
begin
  P_F603SQ: process(H01,H02,H03,H04,H05)
  begin
    if H04 = '0' then
      Q <= '0' ;
   -- pragma translate_off
    elsif H04 /= '1' then
      Q <= 'X' ;
   -- pragma translate_on
    elsif H03 = '1' then
      Q <= FUNC_565(D0 => H01, D1 => H02, A => H05) ;
   -- pragma translate_off
    elsif H03 /= '0' then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F603SQ ;
    N01 <= Q ;
end A;


----- CELL F611 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F611 is
    port(
      N01 : out std_logic ;
      N02 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic );
end F611;

architecture A of F611 is
  signal Q : std_logic ;
begin
  P_F611: process(H02)
  begin
    if H02'event and (H02 = '1') then
      Q <= H01 ;
   -- pragma translate_off
    elsif H02'event and (H02 /= '0') then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F611 ;
    N01 <= Q ;
    N02 <= not Q ;
end A;
```

```
----- CELL F612 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F612 is
    port(
      N01 : out std_logic ;
      N02 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic );
end F612;

architecture A of F612 is
  signal Q : std_logic ;
begin
  P_F612: process(H02,H03)
  begin
    if H03 = '1' then
      Q <= '0' ;
   -- pragma translate_off
    elsif H03 /= '0' then
      Q <= 'X' ;
   -- pragma translate_on
    elsif H02'event and (H02 = '1') then
      Q <= H01 ;
   -- pragma translate_off
    elsif H02'event and (H02 /= '0') then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F612 ;
    N01 <= Q ;
    N02 <= not Q ;
end A;


----- CELL F614 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F614 is
    port(
      N01 : out std_logic ;
      N02 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic );
end F614;

architecture A of F614 is
  signal Q : std_logic ;
begin
  P_F614: process(H02,H03,H04)
  begin
    if H03 = '1' then
      Q <= '0' ;
   -- pragma translate_off
    elsif H03 /= '0' then
      Q <= 'X' ;
   -- pragma translate_on
    elsif H04 = '1' then
      Q <= '1' ;
   -- pragma translate_off
    elsif H04 /= '0' then
      Q <= 'X' ;
   -- pragma translate_on
    elsif H02'event and (H02 = '1') then
      Q <= H01 ;
   -- pragma translate_off
    elsif H02'event and (H02 /= '0') then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F614 ;
    N01 <= Q ;
    N02 <= not Q ;
end A;


-- pragma translate_off
----- CELL F637 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F637 is
    port(
      N01 : out std_logic ;
      N02 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic );
end F637;

architecture A of F637 is
   signal TMP : std_logic ;
   signal SEL : std_logic_vector(1 downto 0) ;

begin
  SEL <= H03 & H04 ;

  process(H02,SEL)
  begin
    case SEL is
      when "11" =>
          if H02'event and (H02 = '0') then
            TMP <= H01 ;
          elsif H02'event and (H02 = 'X') then
            TMP <= 'X' ;
          end if ;
      when "10" => TMP <= '1' ;
      when "01" => TMP <= '0' ;
      when "00" => TMP <= 'X' ;
      when others => TMP <= 'X' ;
    end case ;
  end process;

  N01 <= TMP ;
  N02 <= not(TMP) ;
end A;


----- CELL F661NQ -----
library IEEE; use IEEE.std_logic_1164.all;

entity F661NQ is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic );
end F661NQ;
```

```
architecture A of F661NQ is
  signal Q : std_logic ;
begin
  P_F661NQ: process(H02)
  begin
    if H02'event and (H02 = '0') then
      Q <= H01 ;
   -- pragma translate_off
    elsif H02'event and (H02 /= '1') then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F661NQ ;
    N01 <= Q ;
end A;


----- CELL F665NB -----
library IEEE; use IEEE.std_logic_1164.all;

entity F665NB is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic );
end F665NB;

architecture A of F665NB is
  signal Q : std_logic ;
begin
  P_F665NB: process(H02,H03)
  begin
    if H03 = '0' then
      Q <= '0' ;
   -- pragma translate_off
    elsif H03 /= '1' then
      Q <= 'X' ;
   -- pragma translate_on
    elsif H02'event and (H02 = '0') then
      Q <= H01 ;
   -- pragma translate_off
    elsif H02'event and (H02 /= '1') then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F665NB ;
    N01 <= not Q ;
end A;


----- CELL F612HB -----
library IEEE; use IEEE.std_logic_1164.all;

entity F612HB is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic );
end F612HB;

architecture A of F612HB is
```

```
  signal Q : std_logic ;
begin
  P_F612HB: process(H02,H03)
  begin
    if H03 = '1' then
      Q <= '0' ;
   -- pragma translate_off
    elsif H03 /= '0' then
      Q <= 'X' ;
   -- pragma translate_on
    elsif H02'event and (H02 = '1') then
      if H04 = '0' then
        Q <= H01 ;
     -- pragma translate_off
      elsif H04 /= '1' then
        Q <= 'X' ;
     -- pragma translate_on
      end if ;
   -- pragma translate_off
    elsif H02'event and (H02 /= '0') then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F612HB ;
    N01 <= not Q ;
end A;


----- CELL F767 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F767 is
   port(
      N01, N02      : out   std_logic := 'U';
      H01, H02, H03 : in    std_logic := 'U');
end F767;

architecture A of F767 is
   signal TMP : std_logic := 'U' ;
begin
  process(H01,H02,H03)
   variable SEL : std_logic_vector(1 downto 0) := "XX" ;
  begin
    SEL := H02 & H03 ;
    case SEL is
      when "11" =>
          if H01'event and (H01 = '0') then
            TMP <= not TMP ;
          elsif H01'event and (H01 = 'X') then
            TMP <= 'X' ;
          end if ;
      when "10" => TMP <= '1' ;
      when "01" => TMP <= '0' ;
      when "00" => TMP <= 'X' ;
      when others => TMP <= 'X' ;
    end case ;
  end process;

  N01 <= TMP ;
  N02 <= not(TMP) ;
end A;
```

```
----- CELL F616HQ -----
library IEEE; use IEEE.std_logic_1164.all;

entity F616HQ is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic );
end F616HQ;

architecture A of F616HQ is
  signal Q : std_logic ;
begin
  P_F616HQ: process(H02,H03)
  begin
    if H03 = '0' then
      Q <= '1' ;
   -- pragma translate_off
    elsif H03 /= '1' then
      Q <= 'X' ;
   -- pragma translate_on
    elsif H02'event and (H02 = '1') then
      if H04 = '0' then
        Q <= H01 ;
     -- pragma translate_off
      elsif H04 /= '1' then
        Q <= 'X' ;
     -- pragma translate_on
      end if ;
   -- pragma translate_off
    elsif H02'event and (H02 /= '0') then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F616HQ ;
    N01 <= Q ;
end A;


----- CELL F612SQ -----
library IEEE; use IEEE.std_logic_1164.all;

entity F612SQ is
    port(
      N01 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic ;
      H05 : in  std_logic );
end F612SQ;

architecture A of F612SQ is
  use work.cb10vx_functions.all ;
  signal Q : std_logic ;
begin
  P_F612SQ: process(H03,H04)
  begin
    if H04 = '1' then
      Q <= '0' ;
   -- pragma translate_off
    elsif H04 /= '0' then
      Q <= 'X' ;
   -- pragma translate_on
    elsif H03'event and (H03 = '1') then
      Q <= FUNC_565(D0 => H01, D1 => H02, A => H05) ;
   -- pragma translate_off
    elsif H03'event and (H03 /= '0') then
      Q <= 'X' ;
   -- pragma translate_on
    end if ;
  end process P_F612SQ ;
    N01 <= Q ;
end A;


----- CELL F774 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F774 is
    port(
      N01 : out std_logic ;
      N02 : out std_logic ;
      H01 : in  std_logic ;
      H02 : in  std_logic ;
      H03 : in  std_logic ;
      H04 : in  std_logic ;
      H05 : in  std_logic );
end F774;

architecture A of F774 is
   signal Q : std_logic ;
begin
  process(H03,H04,H05)
   variable SEL,JK : std_logic_vector(1 downto 0) ;
  begin
    SEL := H04 & H05 ;
    case SEL is
      when "00" =>
          if H03'event and (H03 = '1') then
            JK  := H01 & H02 ;
            case JK is
              when "00"   =>
              when "01"   => Q <= '0' ;
              when "10"   => Q <= '1' ;
              when "11"   => Q <= not Q ;
              when others => Q <= 'X' ;
            end case ;
   -- pragma translate_off
          elsif H03'event and (H03 /= '0') then
            Q <= 'X' ;
   -- pragma translate_on
          end if ;
      when "01" => Q <= '1' ;
      when "10" => Q <= '0' ;
      when "11" => Q <= 'X' ;
      when others => Q <= 'X' ;
    end case ;
  end process;

  N01 <= Q ;
  N02 <= not(Q) ;
end A ;
```

```
----- CELL F985 -----
library IEEE; use IEEE.std_logic_1164.all;

entity F985 is
   port(
      N01   : out   std_logic := 'U';
      N02   : out   std_logic := 'U';
      N03   : out   std_logic := 'U';
      H01   : in    std_logic := 'U';
      H02   : in    std_logic := 'U';
      H03   : in    std_logic := 'U';
      H04   : in    std_logic := 'U';
      H05   : in    std_logic := 'U';
      H06   : in    std_logic := 'U';
      H07   : in    std_logic := 'U';
      H08   : in    std_logic := 'U';
      H09   : in    std_logic := 'U';
      H10   : in    std_logic := 'U';
      H11   : in    std_logic := 'U');
end F985;

architecture A of F985 is begin

  P_F985 : process(H01, H02, H03, H04, H05, H06, H07, H08, H09, H10, H11)
    variable SWTCH1, SWTCH2, SWTCH3, SWTCH4, SWTCH5 :std_logic_vector(1 downto 0);
    begin
      N01 <= '0' ; N02 <= '0' ; N03 <= '0' ;

      SWTCH1 := H04 & H08 ;  -- A3,B3
      case SWTCH1 is
        when "10"   => N03 <= '1' ; -- A > B
        when "01"   => N01 <= '1' ; -- A < B
        when "00" | "11" =>
          SWTCH2 := H03 & H07 ;  -- A2,B2
          case SWTCH2 is
            when "10"   => N03 <= '1' ; -- A > B
            when "01"   => N01 <= '1' ; -- A < B
            when "00" | "11" =>
              SWTCH3 := H02 & H06 ;  -- A1,B1
              case SWTCH3 is
                when "10"   => N03 <= '1' ; -- A > B
                when "01"   => N01 <= '1' ; -- A < B
                when "00" | "11" =>
                  SWTCH4 := H01 & H05 ;  -- A0,B0
                  case SWTCH4 is
                    when "10"   => N03 <= '1' ; -- A > B
                    when "01"   => N01 <= '1' ; -- A < B
                    when "00" | "11" =>
                      case H10 is
                        when '1'    => N02 <= '1' ; -- A = B
                        when '0'    =>
                          SWTCH5 := H09 & H11 ;  --  A<B,A>B
                          case SWTCH5 is
                            when "00"   => N03 <= '1' ; N01 <= '1' ;
                            when "01"   => N03 <= '1' ; -- A > B
                            when "10"   => N01 <= '1' ; -- A < B
                            when "11"   =>
                            when others => N01 <= 'X' ; N02 <= 'X' ; N03 <= 'X' ;
                          end case ;
                        when others => N01 <= 'X' ; N02 <= 'X' ; N03 <= 'X' ;
                      end case ;
                    when others => N01 <= 'X' ; N02 <= 'X' ; N03 <= 'X' ;
                  end case ;
                when others => N01 <= 'X' ; N02 <= 'X' ; N03 <= 'X' ;
              end case ;
            when others => N01 <= 'X' ; N02 <= 'X' ; N03 <= 'X' ;
          end case ;
        when others => N01 <= 'X' ; N02 <= 'X' ; N03 <= 'X' ;
      end case ;

    end process P_F985 ;

end A;


----- CELL F924 -----
library IEEE;
use IEEE.std_logic_1164.all;

entity F924 is
   port(
      N01  : out   std_logic := 'U';
      N02  : out   std_logic := 'U';
      N03  : out   std_logic := 'U';
      N04  : out   std_logic := 'U';
      N05  : out   std_logic := 'U';
      N06  : out   std_logic := 'U';
      N07  : out   std_logic := 'U';
      N08  : out   std_logic := 'U';
      H01  : in    std_logic := 'U';
      H02  : in    std_logic := 'U';
      H03  : in    std_logic := 'U';
      H04  : in    std_logic := 'U';
      H05  : in    std_logic := 'U');
end F924;

architecture A of F924 is

begin

  P_F924: process(H05)
  begin
    if H05'event and (H05 = '1') then
      N01 <= H01 ;
      N05 <= not H01 ;
      N02 <= H02 ;
      N06 <= not H02 ;
      N03 <= H03 ;
      N07 <= not H03 ;
      N04 <= H04 ;
      N08 <= not H04 ;
    elsif H05'event and (H05 = 'X') then
      N01 <= 'X' ;
      N02 <= 'X' ;
      N03 <= 'X' ;
      N04 <= 'X' ;
      N05 <= 'X' ;
      N06 <= 'X' ;
      N07 <= 'X' ;
      N08 <= 'X' ;
    end if ;
  end process P_F924 ;
end A;


----- CELL FUNCTION FUNC_565 -----
  function FUNC_565(D0,D1,A : std_logic) return std_logic is
    begin
      return ((D0 and not A) or (D1 and A) or (D0 and D1)) ;
    end ;
```

II. SP700 CELL LIBRARY TRANSFORMATION TABLE FOR ILOC

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; CELL LIBRARY PATTERNS

                                           ((CONC)              ;concurrent passthru
'(N01 ((H01)) )   )
                                           ((F096)                    ;1 output only
'(N01 ( )  )   )
                                           ((F097)                    ;0 output only
'(N01 (( )) )   )
                                 ((F111 F112 F153 F153K F154K F156K F158K L111)  ;buf
'(N01 ((H01)) )   )
                 ((F101 F102 F143 F143K F144K F145K F146K F148 F148K F14C L101)  ;inv
'(N01 (H01)  )   )
                                           ((F202 F222 L202)                ;2nor
'(N01 (H01  H02 ) )   )
                                           ((F202N1 F222N1 L202N1)          ;2norN1
'(N01 (H01 (H02)) )   )
                                           ((F203 F223 F2C3 L203)           ;3nor
'(N01 (H01  H02  H03 ) )   )
                                           ((F203N1 F223N1 L203N1)          ;3norN1
'(N01 (H01  H02 (H03)) )   )
                                           ((F203N2 F223N2 L203N2)          ;3norN2
'(N01 (H01 (H02)(H03)) )   )
                                           ((F204 F224 L204)                ;4nor
'(N01 (H01  H02  H03  H04 ) )   )
                                           ((F204N1)                        ;4norN1
'(N01 (H01  H02  H03 (H04)) )   )
                                           ((L204N2)                        ;4norN2
'(N01 (H01  H02 (H03)(H04)) )   )
                                           ((L205)                          ;5nor
'(N01 (H01  H02  H03  H04  H05 ) )   )
                                           ((L205N1)                        ;5norN1
'(N01 (H01  H02  H03  H04 (H05)) )   )
                                           ((F205N2 L205N2)                 ;5norN2
'(N01 (H01  H02  H03 (H04)(H05)) )   )
                                           ((F205N3 L205N3)                 ;5norN3
'(N01 (H01  H02 (H03)(H04)(H05)) )   )
                                           ((F206 L206)                     ;6nor
'(N01 (H01  H02  H03  H04  H05  H06 ) )   )
                                           ((L208)                          ;8nor
'(N01 (H01  H02  H03  H04  H05  H06  H07  H08 ) )   )
                                           ((F212 F232 F2D2K L212)          ;2or
'(N01 ((H01  H02 )) )   )
                                           ((F213 F233 L213)                ;3or
'(N01 ((H01  H02  H03 )) )   )
                                           ((F214 L214)                     ;4or
'(N01 ((H01  H02  H03  H04 )) )   )
                                           ((F214N1 F234N1 L214N1)          ;4orN1
'(N01 ((H01  H02  H03 (H04))) )   )
                                           ((L215)                          ;5or
'(N01 ((H01  H02  H03  H04  H05 )) )   )
                                           ((F235N1 L215N1)                 ;5orN1
'(N01 ((H01  H02  H03  H04 (H05))) )   )
                                           ((F216 L216)                     ;6or
'(N01 ((H01  H02  H03  H04  H05  H06 )) )   )
                                           ((F218 F238 L218)                ;8or
'(N01 ((H01  H02  H03  H04  H05  H06  H07  H08 )) )   )
                                           ((F302 F322 F3C2 L302)           ;2nand
'(N01 (((H01)(H02))) )   )
                                           ((F302N1 F322N1 L302N1)          ;2nandN1
'(N01 (((H01) H02 )) )   )
                                           ((F303 F323 F3C3 L303)           ;3nand
'(N01 (((H01)(H02)(H03))) )   )
```

```
                                           ((F303N1 F323N1 L303N1)          ;3nandN1
'(N01 (((H01)(H02) H03 )) )   )
                                           ((F303N2 F323N2 L303N2)          ;3nandN2
'(N01 (((H01) H02  H03 )) )   )
                                           ((F304 F324 L304)                ;4nand
'(N01 (((H01)(H02)(H03)(H04))) )   )
                                           ((F304N1 F324N1 L304N1)          ;4nandN1
'(N01 (((H01)(H02)(H03) H04 )) )   )
                                           ((F304N2 L304N2)                 ;4nandN2
'(N01 (((H01)(H02) H03  H04 )) )   )
                                           ((F305 L305)                     ;5nand
'(N01 (((H01)(H02)(H03)(H04)(H05))) )   )
                                           ((L305N1)                        ;5nandN1
'(N01 (((H01)(H02)(H03)(H04) H05 )) )   )
                                           ((F305N2 F325N2 L305N2)          ;5nandN2
'(N01 (((H01)(H02)(H03) H04  H05 )) )   )
                                           ((F305N3 F325N3 L305N3)          ;5nandN3
'(N01 (((H01)(H02) H03  H04  H05 )) )   )
                                           ((L306)                          ;6nand
'(N01 (((H01)(H02)(H03)(H04)(H05)(H06))) )   )
                                           ((L308)                          ;8nand
'(N01 (((H01)(H02)(H03)(H04)(H05)(H06)(H07)(H08))) )   )
                                           ((F312 F332 F3D2K L312)          ;2and
'(N01 ((H01)(H02)) )   )
                                           ((F313 F333 L313)                ;3and
'(N01 ((H01)(H02)(H03)) )   )
                                           ((F314 F334 L314)                ;4and
'(N01 ((H01)(H02)(H03)(H04)) )   )
                                           ((F314N1 F334N1 L314N1)          ;4andN1
'(N01 ((H01)(H02)(H03) H04 ) )   )
                                           ((L315)                          ;5and
'(N01 ((H01)(H02)(H03)(H04)(H05)) )   )
                                           ((F315N1 L315N1)                 ;5andN1
'(N01 ((H01)(H02)(H03)(H04) H05 ) )   )
                                           ((L316)                          ;6and
'(N01 ((H01)(H02)(H03)(H04)(H05)(H06)) )   )
                                           ((F318 F338 L318)                ;8and
'(N01 ((H01)(H02)(H03)(H04)(H05)(H06)(H07)(H08)) )   )
                                           ((F511 F511NS F518 F518NS L511)  ;2xor
'(N01  ((H01 H02)((H01)(H02)))  )   )
                                           ((F512 F512NS F519 F519NS L512)  ;2nxor
'(N01 (((H01 H02)((H01)(H02)))) )   )
                                           ((F516 F516ND F516NS L516)       ;3xor
'(N01 (((H01 H02 (H03))(H01 H03 (H02))
        (H02 H03 (H01))((H01)(H02)(H03)))) )   )
                                           ((F517 F517ND F517NSP)           ;3nxor
'(N01  ((H01 H02 (H03))(H01 H03 (H02))
        (H02 H03 (H01))((H01)(H02)(H03)))  )   )
                                           ((F51C)                          ;4xor
'(N01 (((H01 H02)(H03 H04)((H01)(H02))((H03)(H04)))
       (((H01 H02)((H01)(H02)))((H03 H04)((H03)(H04))))) )   )
                                           ((F51D)                          ;4nxor
'(N01 ((((H01 H02)(H03 H04)((H01)(H02))((H03)(H04)))
       (((H01 H02)((H01)(H02)))((H03 H04)((H03)(H04)))))) )   )
                                           ((L401)                          ;exorand
'(N01 ((H01) (H02 H03) ((H02)(H03))) )   )
                                           ((L401NC)                      ;exorN1andN1
'(N01 (H01 ((H02 H03) ((H02)(H03)))) )   )
                                           ((L401ND)                        ;exorN1and
'(N01 ((H01) ((H02 H03) ((H02)(H03)))) )   )
                                           ((L411)                          ;andexor
'(N01 (((H01)(H02)(H03)) (H01 ((H02)(H03)))) )   )
                                           ((F421 F421NP F421NS L421)       ;andnor
'(N01 (H01 ((H02)(H03))) )   )
                                           ((F421NA L421NA)                 ;andnorN1
'(N01 ((H01)((H02)(H03))) )   )
                                           ((F421NB L421NB)               ;andN1norN1
'(N01 ((H01)(H02 (H03))) )   )
                                           ((F421NC L421NC)             ;andN1N2norN1
'(N01 ((H01)(H02 H03)) )   )
                                           ((F421ND L421ND)                 ;andN1nor
'(N01 (H01 (H02 (H03))) )   )
                                           ((F421NE L421NE)               ;andN1N2nor
'(N01 (H01 (H02 H03)) )   )
                                           ((F422 L422 F422NP)              ;and3nor
'(N01 (H01 H02 ((H03)(H04))) )   )
                                           ((F422NA L422NA)                 ;and3norN1
'(N01 ((H01) H02 ((H03)(H04))) )   )
                                           ((F422NB L422NB)              ;and3norN1N2
'(N01 ((H01)(H02)((H03)(H04))) )   )
                                           ((L422NC)                   ;andN13norN1N2
'(N01 ((H01)(H02)(H03 (H04))) )   )
                                           ((F422ND L422ND)          ;andN1N23norN1N2
'(N01 ((H01)(H02)(H03 H04)) )   )
                                           ((L422NF)                   ;andN1N23norN1
'(N01 ((H01) H02 (H03 H04)) )   )
                                           ((L422NG)                        ;andN13nor
'(N01 (H01 H02 (H03 (H04))) )   )
                                           ((F422NH L422NH)               ;andN1N23nor
'(N01 (H01 H02 (H03 H04)) )   )
                                           ((L423)                          ;3andnor
'(N01 (H01 ((H02)(H03)(H04))) )   )
                                           ((F423NA L423NA)                 ;3andnorN1
'(N01 ((H01)((H02)(H03)(H04))) )   )
                                           ((F423NB L423NB)               ;3andN1norN1
'(N01 ((H01)(H02 (H03)(H04))) )   )
                                           ((L423NC)                    ;3andN1N2norN1
'(N01 ((H01)(H02 H03 (H04))) )   )
                                           ((L423ND)                  ;3andN1N2N3norN1
'(N01 ((H01)(H02 H03 H04)) )   )
                                           ((L423NE)                        ;3andN1nor
'(N01 (H01 (H02 (H03)(H04))) )   )
                                           ((F423NF L423NF)               ;3andN1N2nor
'(N01 (H01 (H02 H03 (H04))) )   )
                                           ((F423NG L423NG)             ;3andN1N2N3nor
'(N01 (H01 (H02 H03 H04)) )   )
                                           ((F424 L424 F424NP F424NS)       ;andandnor
'(N01 (((H01)(H02))((H03)(H04))) )   )
                                           ((F424NA L424NA)               ;andN1andnor
'(N01 (( H01 (H02))((H03)(H04))) )   )
                                           ((F424NB L424NB)             ;andN1N2andnor
'(N01 ((H01 H02)((H03)(H04))) )   )
                                           ((F424NC L424NC)         ;andN1N2andN1N2nor
'(N01 ((H01 H02)(H03 H04)) )   )
                                           ((L424ND)                    ;andN1andN1nor
'(N01 ((H01 (H02))(H03 (H04))) )   )
                                           ((L424NE)                  ;andN1andN1N2nor
'(N01 ((H01 (H02))(H03 H04)) )   )
                                           ((F425 L425 F425NP)           ;andandandnor
'(N01 (((H01)(H02))((H03)(H04))((H05)(H06))) )   )
                                           ((F425NF L425NF)  ;andN1N2andN1N2andN1N2nor
'(N01 ((H01 H02)(H03 H04)(H05 H06)) )   )
                                           ((L426)                        ;3and3andnor
'(N01 (((H01)(H02)(H03))((H04)(H05)(H06))) )   )
                                           ((F427 L427)                    ;and3andnor
'(N01 (((H01)(H02))((H03)(H04)(H05))) )   )
                                           ((L427ND)               ;andN1N23andN1N2nor
'(N01 ((H01 H02)(H03 H04 (H05))) )   )
                                           ((L427NE)             ;andN1N23andN1N2N3nor
'(N01 ((H01 H02)(H03 H04 H05)) )   )
```

```
'(N01 (H01 ((H02)(H03)) ((H04)(H05))) )   )

'(N01 ((H01) (H02 H03) (H04 H05)) )   )

'(N01 (((H01)(H02)) ((H03)(H04))
       ((H05)(H06)) ((H07)(H08))) )   )

'(N01 (((H01)(H02 H03))) )   )

'(N01 ((H01 (H02 H03))) )   )

'(N01 ((H01 ((H02) H03))) )   )

'(N01 ((H01 ((H02)(H03)))) )   )

'(N01 (((H01)((H02) H03))) )   )

'(N01 (((H01)((H02)(H03)))) )   )

'(N01 (((H01)(H02)(H03 H04))) )   )

'(N01 ((H01 (H02)(H03 H04))) )   )

'(N01 ((H01 H02 (H03 H04))) )   )

'(N01 ((H01 H02 ((H03)(H04)))) )   )

'(N01 ((H01 (H02)((H03) H04))) )   )

'(N01 ((H01 (H02)((H03)(H04)))) )   )

'(N01 (((H01)(H02)((H03) H04))) )   )

'(N01 (((H01)(H02)((H03)(H04)))) )   )

'(N01 (((H01)(H02 H03 H04))) )   )

'(N01 ((H01 (H02 H03 H04))) )   )

'(N01 ((H01 ((H02)(H03) H04))) )   )

'(N01 ((H01 ((H02)(H03)(H04)))) )   )

'(N01 (((H01)((H02) H03 H04))) )   )

'(N01 (((H01)((H02)(H03) H04))) )   )

'(N01 (((H01)((H02)(H03)(H04)))) )   )

'(N01 (((H01 H02)(H03 H04))) )   )

'(N01 ((((H01) H02)(H03 H04))) )   )

'(N01 ((((H01)(H02))(H03 H04))) )   )

'(N01 ((((H01)(H02))((H03)(H04)))) )   )

'(N01 ((((H01) H02)((H03) H04))) )   )

'(N01 ((((H01) H02)((H03)(H04)))) )   )

'(N01 (((H01 H02)(H03 H04 H05))) )   )

'(N01 ((((H01)(H02))((H03)(H04) H05))) )   )
```

```
((F428 L428 F428NS)            ;andand3nor

((F428NE L428NE)      ;andN1N2andN1N23norN1

((F429 L429)             ;andandandandnor


((F431 L431 F431NP F431NS)       ;ornand

((F431NA L431NA)                 ;ornandN1

((F431NB L431NB)               ;orN1nandN1

((F431NC L431NC)             ;orN1N2nandN1

((F431ND L431ND)                 ;orN1nand

((F431NE L431NE)               ;orN1N2nand

((F432 L432 F432NP)              ;or3nand

((F432NA L432NA)                 ;or3nandN1

((F432NB L432NB)               ;or3nandN1N2

((F432ND L432ND)           ;orN1N23nandN1N2

((L432NE)                      ;orN13nandN1

((L432NF)                    ;orN1N23nandN1

((L432NG)                        ;orN13nand

((F432NH L432NH)               ;orN1N23nand

((F433 L433 F433NS)              ;3ornand

((F433NA L433NA)                 ;3ornandN1

((L433NC)                    ;3orN1N2nandN1

((F433ND L433ND)           ;3orN1N2N3nandN1

((F433NE L433NE)                 ;3orN1nand

((F433NF L433NF)               ;3orN1N2nand

((F433NG L433NG)             ;3orN1N2N3nand

((F434NS L434)                   ;orornand

((F434NA L434NA)               ;orN1ornand

((F434NB L434NB)             ;orN1N2ornand

((F434NC L434NC)         ;orN1N2orN1N2nand

((L434ND)                    ;orN1orN1nand

((L434NE)                  ;orN1orN1N2nand

((F435 L435)                     ;or3ornand

((F435ND L435ND)        ;orN1N23orN1N2nand
                                           ((F435NE L435NE)       ;orN1N23orN1N2N3nand
'(N01 ((((H01)(H02))((H03)(H04)(H05)))) )   )
                                           ((F436 L436)                    ;3or3ornand
'(N01 (((H01 H02 H03)(H04 H05 H06))) )   )
                                           ((L437)                          ;oror3nand
'(N01 (((H01) (H02 H03) (H04 H05))) )   )
                                           ((F437NE L437NE)       ;orN1N2orN1N23nandN1
'(N01 ((H01 ((H02)(H03))((H04)(H05)))) )   )
                                           ((F438 F438NP F438NS L438)     ;ororor3nand
'(N01 (((H01 H02)(H03 H04)(H05 H06))) )   )
                                           ((F438NF L438NF)   ;orN1N2orN1N2orN1N23nand
'(N01 ((((H01)(H02))((H03)(H04))((H05)(H06)))) )   )
                                           ((F442 L442)                   ;4and4andnor
'(N01 (((H01)(H02)(H03)(H04))
       ((H05)(H06)(H07)(H08))) )   )
                                           ((F454 L454)                 ;orororor4nand
'(N01 (((H01 H02)(H03 H04)(H05 H06)(H07 H08))) )   )
                                           ((F462 L462)                   ;3andand3nor
'(N01 (H06 ((H04)(H05))((H01)(H02)(H03))) )   )
                                           ((F565 F565NS F565NSP F565NT L565)  ;3adder
'(N01 (((H03 (H01))((H01)(H02))((H02)(H03)))) )   )
                                   ((F57B F57BNP F57BNS F57BNSP F57BNTK L57B) ;3adder-
'(N01 ((H03 (H01))((H01)(H02))((H02)(H03))) )   )
                                           ((F571NP)                        ;4adder
'(N01 (H04 ((H03 (H01))((H01)(H02))((H02)(H03)))) )   )
                                           ((F56G L56G)                     ;5adder
'(N01 ((((H05)(H03))(H04 H05 (H01))(H04 (H01)(H03))
        (H05 (H04)(H02))(H05 (H01)(H02))
        ((H04)(H02)(H03))((H01)(H02)(H03)))) )   )
                                           ((F57G)                          ;5adder-
'(N01  (((H05)(H03))(H04 H05 (H01))(H04 (H01)(H03))
        (H05 (H04)(H02))(H05 (H01)(H02))
        ((H04)(H02)(H03))((H01)(H02)(H03))) )   )
                                           ((F564 F564NS L564)              ;6adder
'(N01 ((((H01) H05 H06)((H02)(H05) H06)((H03) H05 (H06))
        ((H04)(H05)(H06))((H01)(H02) H06)
        ((H03)(H04)(H06))((H01)(H03) H05)
        ((H02)(H04)(H05))((H01)(H02)(H03)(H04)))) )   )
                                           ((F57A)                          ;6adder-
'(N01  (((H01) H05 H06)((H02)(H05) H06)((H03) H05 (H06))
        ((H04)(H05)(H06))((H01)(H02) H06)
        ((H03)(H04)(H06))((H01)(H03) H05)
        ((H02)(H04)(H05))((H01)(H02)(H03)(H04))) )   )
                                           ((F581)                          ;8parity
'(N01 (((((H01 H02)((H01)(H02)))((H03 H04)((H03)(H04))))
        (((H05 H06)((H05)(H06)))((H07 H08)((H07)(H08))))
        ((H01 H02)((H01)(H02))(H03 H04)((H03)(H04)))
        ((H05 H06)((H05)(H06))(H07 H08)((H07)(H08))))
       ((((H01 H02)((H01)(H02)))((H03 H04)((H03)(H04))))
         ((H01 H02)((H01)(H02))(H03 H04)((H03)(H04))))
        (((H05 H06)((H05)(H06)))((H07 H08)((H07)(H08))))
         ((H05 H06)((H05)(H06))(H07 H08)((H07)(H08)))))) )   )
                                           ((F582)                          ;8parity-
'(N01 ((((((H01 H02)((H01)(H02)))((H03 H04)((H03)(H04))))
         (((H05 H06)((H05)(H06)))((H07 H08)((H07)(H08))))
         ((H01 H02)((H01)(H02))(H03 H04)((H03)(H04)))
         ((H05 H06)((H05)(H06))(H07 H08)((H07)(H08))))
        ((((H01 H02)((H01)(H02)))((H03 H04)((H03)(H04))))
          ((H01 H02)((H01)(H02))(H03 H04)((H03)(H04))))
         (((H05 H06)((H05)(H06)))((H07 H08)((H07)(H08))))
          ((H05 H06)((H05)(H06))(H07 H08)((H07)(H08))))))) )   )
                                           ((F584 L584)                     ;3maj
'(N01 ((((H01)(H02)) ((H03)(H01 H02))))  )   )
```

```
;;;non-register multiple-output-cells
                                           ((F091)                      ;m-output only
'( (N01  ( ) )
   (N02 (( )) ) )   )
                                           ((F521 F521NS L521 L528)         ;m-3adder
'( (N01  (((H01 H02 (H03))(H01 H03 (H02))
           (H02 H03 (H01))((H01)(H02)(H03)))) )
   (N02 (((H01)(H02)) ((H03)(H01 H02)))) ) )   )
                                           ((L522)                          ;m-2adder
'( (N01 ((H01 H02)((H01)(H02))) )
   (N02 ((H01)(H02))  ) )   )
                                           ((F523)                          ;virtex
'( (N01 ((H01 H02)((H01)(H02))) )
   (N02 ((H03 H04)((H03)(H04))) )
   (N03 ((H05 H06)((H05)(H06))) )
   (N04 ((H07 H08)((H07)(H08))) )
   (N05 (( )) ) )   )
;                                           ((F985)                         ;virtex
;'( (N01 (((G4) (H10 H11))                                                  ; NOT USED
;         (G4 ((((H04 H08)((H04)(H08)))
;               ((((H03 H07)((H03)(H07)))
;                 ((((H02 H06)((H02)(H06)))
;                   ((((H01 H05)((H01)(H05))) (H10 H11))
;                    ((H01 H05)((H01)(H05)) H05)))
;                  ((H02 H06)((H02)(H06)) H06)))
;                ((H03 H07)((H03)(H07)) H07)))
;              ((H04 H08)((H04)(H08)) H08)))) )
;   (N02 ((H10)(G4)) )
;   (N03 (((G4) (H10 H09))
;         (G4 ((((H04 H08)((H04)(H08)))
;               ((((H03 H07)((H03)(H07)))
;                 ((((H02 H06)((H02)(H06)))
;                   ((((H01 H05)((H01)(H05))) (H10 H09))
;                    ((H01 H05)((H01)(H05)) H01)))
;                  ((H02 H06)((H02)(H06)) H02)))
;                ((H03 H07)((H03)(H07)) H03)))
;              ((H04 H08)((H04)(H08)) H04)))) ) )   )

                                           ((WSRAMDHSAA32W33C2)                ;memory
'(Nxx RAM33 ((- -)(- -)) ((- -)(- -)) ((- -)(- -)))   )
                                           ((WSRAMDHSAA32W37C2)                ;memory
'(Nxx RAM37 ((- -)(- -)) ((- -)(- -)) ((- -)(- -)))   )
                                           ((WSRAMDHSAA32W41C2)                ;memory
'(Nxx RAM41 ((- -)(- -)) ((- -)(- -)) ((- -)(- -)))   )


;;;;;;;;;;;;;;;;;ORIGINAL REGISTER MODELS

                                           ((F601NQ L601NQ)           ;latch H02=1clock
'(N01 (((H02) H01)( H02  *-*)) )   )
                                           ((F604NQ L604NQ)                 ;latch
'(N01 (( H02  H01)((H02) *-*)) )   )
                                           ((F603SQ)               ;5latch H04=0reset
'(N01 ((H04)((H03)(H05 (H01))((H01)(H02))((H02)(H05)))
       (H03 *-*)) )   )
                                          ((F611NQ F641NQ L611NQ) ;dff H02=clock-event
'(N01  (((H02) H01)(H02 *-*))  )   )
                                           ((F631NB)                        ;dff
'(N01 ((((H02) H01)(H02 *-*))) )   )
                                           ((F611HB)                        ;3dff
'(N01 ((((H02)((H03 H01)((H03) *-*)))
        (H02 *-*))) )   )
                                           ((F611HQ F641HQ)                 ;3dff
'(N01  (((H02)((H03 H01)((H03) *-*)))
        (H02 *-*))  )   )
```

```
                                           ((F612NB)                        ;3dff
'(N01 (( H03  ((H02) H01)(H02 *-*)))  )    )
                                           ((F612NQ F642NQ L612NQ)          ;3dff
'(N01  ( H03  ((H02) H01)(H02 *-*))   )    )
                                           ((L613NQ)                        ;3dff
'(N01 (( H03 (((H02) H01)(H02 *-*)))) )    )
                                           ((F615NQ F645NQ)                 ;3dff
'(N01  ((H03) ((H02) H01)(H02 *-*))  )    )
                                           ((F616NB)                        ;3dff
'(N01 (((H03) ((H02) H01)(H02 *-*))) )    )
                                           ((F611SB)                        ;4dff
'(N01 ((((H03)(H04 (H01))((H01)(H02))((H02)(H04)))
        ( H03  *-*))) )    )
                                           ((F611SQ L611SQ)                 ;4dff
'(N01  (((H03)(H04 (H01))((H01)(H02))((H02)(H04)))
        (H03 *-*))  )    )
                                           ((F612HB)                        ;4dff
'(N01 (( H03  ((H02)((H04 H01)((H04) *-*)))
              (H02 *-*)))  )    )
                                           ((F612HQ F642HQ)                 ;4dff
'(N01  ( H03  ((H02)((H04 H01)((H04) *-*)))
              (H02 *-*))   )    )
                                           ((F613HQ)                        ;4dff
'(N01 (( H03 (((H02)((H04 H01)((H04) *-*)))
              (H02 *-*)))) )    )
                                           ((F615HB)                        ;4dff
'(N01 (((H03)(((H02)((H04 H01)((H04) *-*)))
              (H02 *-*)))) )    )
                                           ((F615HQ F645HQ)                 ;4dff
'(N01  ((H03) ((H02)((H04 H01)((H04) *-*)))
        (H02 *-*))   )    )
                                           ((F616HB)                        ;4dff
'(N01  ((H03)(((H02)((H04 H01)((H04) *-*)))
              (H02 *-*)))  )    )
                                           ((F616HQ)                        ;4dff
'(N01 (((H03) ((H02)((H04 H01)((H04) *-*)))
              (H02 *-*)))  )    )
                                           ((F612SQ L612SQ)                 ;5dff
'(N01  (H04 ((H03)(H05 (H01))((H01)(H02))((H02)(H05)))
            (H03 *-*)) )    )
                                           ((F611TQ F641TQ L611TQ)          ;6dff
'(N01 (((H04)((H06)(H03))(H05 H06 (H01))
        (H05 (H01)(H03))(H06 (H05)(H02))
        (H06 (H01)(H02))((H05)(H02)(H03))
        ((H01)(H02)(H03))) (H04 *-*))   )    )
                                           ((L612TQ)                        ;7dff
'(N01  (H05  ((H04)((H07)(H03))(H06 H07 (H01))
              (H06 (H01)(H03))(H07 (H06)(H02))
              (H07 (H01)(H02))((H06)(H02)(H03))
              ((H01)(H02)(H03)) (H04 *-*)))   )    )
                                           ((L613TQ)                        ;7dff
'(N01 ((H05 (((H04)((H07)(H03))(H06 H07 (H01))
              (H06 (H01)(H03))(H07 (H06)(H02))
              (H07 (H01)(H02))((H06)(H02)(H03))
              ((H01)(H02)(H03)) (H04 *-*))))) )    )

;;;ORIGINAL-register-multiple-output-cells

                                           ((L603NW)              ;m-3latch H03=0reset
'( (N01  ((H03)((H02) H01)(H02 *-*))   )
   (N02 (((H03)((H02) H01)(H02 *-*)))  ) )    )
                                           ((F611 L611NW)                   ;m-dff
'( (N01  (((H02) H01)( H02  *-*))  )
   (N02 ((((H02) H01)( H02  *-*))) ) )    )
                                           ((L631NW)                        ;m-dff
'( (N01  (( H02  H01)((H02) *-*))  )
   (N02 ((( H02  H01)((H02) *-*))) ) )   )
                                           ((F615 L615NW)                   ;m-3dff
'( (N01  ((H03)((H02) H01)( H02 *-*))  )
   (N02 (((H03)((H02) H01)( H02 *-*))) ) )   )
                                           ((F771)                          ;m-3dff
'( (N01  (((H03) (((H02)(*INV* (H01))) (H01 H02 *-*)))
          (H03 *-*)) )
   (N02 ((((H03) (((H02)(*INV* (H01))) (H01 H02 *-*)))
          (H03 *-*))) ) )   )
                                           ((F775)                          ;m-4dff
'( (N01  ((H04)
          (((H03) (((H02)(*INV* (H01))) (H01 H02 *-*)))
           (H03 *-*)))  )
   (N02 (((H04)
          (((H03) (((H02)(*INV* (H01))) (H01 H02 *-*)))
           (H03 *-*)))) ) )   )
                                           ((F791)                          ;m-4dff
'( (N01  (((H03)(H04) *_*)
          (H04 (H03 ((H01)(((H02) *INV*)(H02 *-*)))
                    (H01 *-*)))) )
   (N02 ((((H03)(H04) *_*)
          (H04 (H03 ((H01)(((H02) *INV*)(H02 *-*)))
                    (H01 *-*))))) ) )   )
                                           ((F922)                       ;8dff-special
'( (N01  (H06  ((H05) H01 )(H05 *-*))   )
   (N02  (H06  ((H05) H02 )(H05 *-*))   )
   (N03  (H06  ((H05) H03 )(H05 *-*))   )
   (N04  (H06  ((H05) H04 )(H05 *-*))   )
   (N05 ((H06 (((H05)(H01))(H05 *-*)))) )
   (N06 ((H06 (((H05)(H02))(H05 *-*)))) )
   (N07 ((H06 (((H05)(H03))(H05 *-*)))) )
   (N08 ((H06 (((H05)(H04))(H05 *-*)))) ) )   )
                                           ((F924)                       ;8dff-special
'( (N01 (((H05) H01 )(H05 *-*)) )
   (N02 (((H05) H02 )(H05 *-*)) )
   (N03 (((H05) H03 )(H05 *-*)) )
   (N04 (((H05) H04 )(H05 *-*)) )
   (N05 (((H05)(H01))(H05 *-*)) )
   (N06 (((H05)(H02))(H05 *-*)) )
   (N07 (((H05)(H03))(H05 *-*)) )
   (N08 (((H05)(H04))(H05 *-*)) ) )   )


                                           (OTHERWISE                    ;;unknown
`(Nxx  ,type ((- -)(- -)) ((- -)(- -)) ((- -)(- -)) )   )
```

III. SP700 CELL LIBRARY TYPE SUMMARY

```
  (case type
    (CONC    (list type))               ;concurrent passthru
    (F096    '(OUT1 0))                 ;1 output only
    (F097    '(OUT0 0))                 ;0 output only
    (F101    '(INV 1))
    (F102    '(INV 1))
    (F111    '(BUF 1))
    (F112    '(BUF 1))
    (F143    '(INV 1))
    (F143K   '(INV 1))
    (F144K   '(INV 1))
    (F145K   '(INV 1))
    (F146K   '(INV 1))
    (F148    '(INV 1))
    (F148K   '(INV 1))
    (F14C    '(INV 1))
    (F153    '(BUF 1))
    (F153K   '(BUF 1))
    (F154K   '(BUF 1))
    (F156K   '(BUF 1))
    (F158K   '(BUF 1))
    (F202    '(NOR 2))
    (F202N1  '(NOR 2))
    (F203    '(NOR 3))
    (F203N1  '(NOR 3))
    (F203N2  '(NOR 3))
    (F204    '(NOR 4))
    (F204N1  '(NOR 4))
    (F205N2  '(NOR 5))
    (F205N3  '(NOR 5))
    (F206    '(NOR 6))
    (F212    '(OR 2))
    (F213    '(OR 3))
    (F214    '(OR 4))
    (F214N1  '(OR 4))
    (F216    '(OR 6))
    (F218    '(OR 8))
    (F222    '(NOR 2))
    (F222N1  '(NOR 2))
    (F223    '(NOR 3))
    (F223N1  '(NOR 3))
    (F223N2  '(NOR 3))
    (F224    '(NOR 4))
    (F232    '(OR 2))
    (F233    '(OR 3))
    (F234N1  '(OR 4))
    (F235N1  '(OR 5))
    (F238    '(OR 8))
    (F2C3    '(NOR 3))
    (F2D2K   '(OR 2))
    (F302    '(NAND 2))
    (F302N1  '(NAND 2))
    (F303    '(NAND 3))
    (F303N1  '(NAND 3))
    (F303N2  '(NAND 3))
    (F304    '(NAND 4))
    (F304N1  '(NAND 4))
    (F304N2  '(NAND 4))
    (F305    '(NAND 5))
    (F305N2  '(NAND 5))
    (F305N3  '(NAND 5))
    (F312    '(AND 2))
    (F313    '(AND 3))
```

```
(F314    '(AND 4))
(F314N1  '(AND 4))
(F315N1  '(AND 5))
(F318    '(AND 8))
(F322    '(NAND 2))
(F322N1  '(NAND 2))
(F323    '(NAND 3))
(F323N1  '(NAND 3))
(F323N2  '(NAND 3))
(F324    '(NAND 4))
(F324N1  '(NAND 4))
(F325N2  '(NAND 5))
(F325N3  '(NAND 5))
(F332    '(AND 2))
(F333    '(AND 3))
(F334    '(AND 3))
(F334N1  '(AND 4))
(F338    '(AND 8))
(F3C2    '(NAND 2))
(F3C3    '(NAND 3))
(F3D2K   '(AND 2))
(F421    '(AND-NOR 3))
(F421NA  '(AND-NOR 3))
(F421NB  '(AND-NOR 3))
(F421NC  '(AND-NOR 3))
(F421ND  '(AND-NOR 3))
(F421NE  '(AND-NOR 3))
(F421NP  '(AND-NOR 3))
(F421NS  '(AND-NOR 3))
(F422    '(AND-NOR 4))
(F422NA  '(AND-NOR 4))
(F422NB  '(AND-NOR 4))
(F422ND  '(AND-NOR 4))
(F422NH  '(AND-NOR 4))
(F422NP  '(AND-NOR 4))
(F423NA  '(AND-NOR 4))
(F423NB  '(AND-NOR 4))
(F423NF  '(AND-NOR 4))
(F423NG  '(AND-NOR 4))
(F424    '(AND-NOR 4))
(F424NA  '(AND-NOR 4))
(F424NB  '(AND-NOR 4))
(F424NC  '(AND-NOR 4))
(F424NP  '(AND-NOR 4))
(F424NS  '(AND-NOR 4))
(F425    '(AND-NOR 6))
(F425NF  '(AND-NOR 6))
(F425NP  '(AND-NOR 6))
(F427    '(AND-NOR 5))
(F428    '(AND-NOR 5))
(F428NE  '(AND-NOR 5))
(F428NS  '(AND-NOR 5))
(F429    '(AND-NOR 8))
(F431    '(OR-NAND 3))
(F431NA  '(OR-NAND 3))
(F431NB  '(OR-NAND 3))
(F431NC  '(OR-NAND 3))
(F431ND  '(OR-NAND 3))
(F431NE  '(OR-NAND 3))
(F431NP  '(OR-NAND 3))
(F431NS  '(OR-NAND 3))
(F432    '(OR-NAND 4))
(F432NA  '(OR-NAND 4))
(F432NB  '(OR-NAND 4))
(F432ND  '(OR-NAND 4))
```

```
(F432NH  '(OR-NAND 4))
(F432NP  '(OR-NAND 4))
(F433    '(OR-NAND 4))
(F433NA  '(OR-NAND 4))
(F433ND  '(OR-NAND 4))
(F433NE  '(OR-NAND 4))
(F433NF  '(OR-NAND 4))
(F433NG  '(OR-NAND 4))
(F433NS  '(OR-NAND 4))
(F434NA  '(OR-NAND 4))
(F434NB  '(OR-NAND 4))
(F434NC  '(OR-NAND 4))
(F434NS  '(OR-NAND 4))
(F435    '(OR-NAND 5))
(F435ND  '(OR-NAND 5))
(F435NE  '(OR-NAND 5))
(F436    '(OR-NAND 6))
(F437NE  '(OR-NAND 5))
(F438    '(OR-NAND 6))
(F438NF  '(OR-NAND 6))
(F438NP  '(OR-NAND 6))
(F438NS  '(OR-NAND 6))
(F442    '(OR-NAND 8))
(F454    '(OR-NAND 8))
(F462    '(OR-NAND 6))
(F511    '(XOR 2))
(F511NS  '(XOR 2))
(F512    '(XOR 2))
(F512NS  '(XOR 2))
(F516    '(XOR 3))
(F516ND  '(XOR 3))
(F516NS  '(XOR 3))
(F517    '(XOR 3))
(F517ND  '(XOR 3))
(F517NSP '(XOR 3))
(F518    '(XOR 2))
(F518NS  '(XOR 2))
(F519    '(XOR 2))
(F519NS  '(XOR 2))
(F51C    '(XOR 4))
(F51D    '(XOR 4))
(F564    '(ADDER 6))
(F564NS  '(ADDER 6))
(F565    '(ADDER 3))
(F565NS  '(ADDER 3))
(F565NSP '(ADDER 3))
(F565NT  '(ADDER 3))
(F56G    '(ADDER 5))
(F571NP  '(ADDER 4))
(F57A    '(ADDER 6))
(F57B    '(ADDER 3))
(F57BNP  '(ADDER 3))
(F57BNS  '(ADDER 3))
(F57BNSP '(ADDER 3))
(F57BNTK '(ADDER 3))
(F57G    '(ADDER 5))
(F581    '(PARITY 8))
(F582    '(PARITY 8))
(F584    '(MAJ 3))
(F601NQ  '(LATCH 2))
(F603SQ  '(LATCH 5))
(F604NQ  '(LATCH 2))
(F611HB  '(DFF 3))
(F611HQ  '(DFF 3))
(F611NQ  '(DFF 2))
```

```
(F611SB  '(DFF 4))
(F611SQ  '(DFF 4))
(F611TQ  '(DFF 6))
(F612HB  '(DFF 4))
(F612HQ  '(DFF 4))
(F612NB  '(DFF 3))
(F612NQ  '(DFF 3))
(F612SQ  '(DFF 5))
(F613HQ  '(DFF 4))
(F615HB  '(DFF 4))
(F615HQ  '(DFF 4))
(F615NQ  '(DFF 3))
(F616HB  '(DFF 4))
(F616HQ  '(DFF 4))
(F616NB  '(DFF 3))
(F631NB  '(DFF 2))
(F641HQ  '(DFF 3))
(F641NQ  '(DFF 2))
(F641TQ  '(DFF 6))
(F642HQ  '(DFF 4))
(F642NQ  '(DFF 3))
(F645HQ  '(DFF 4))
(F645NQ  '(DFF 3))
(L101    '(INV 1))
(L111    '(BUF 1))
(L202    '(NOR 2))
(L202N1  '(NOR 2))
(L203    '(NOR 3))
(L203N1  '(NOR 3))
(L203N2  '(NOR 3))
(L204    '(NOR 4))
(L204N2  '(NOR 4))
(L205    '(NOR 5))
(L205N1  '(NOR 5))
(L205N2  '(NOR 5))
(L205N3  '(NOR 5))
(L206    '(NOR 6))
(L208    '(NOR 8))
(L212    '(OR 2))
(L213    '(OR 3))
(L214    '(OR 4))
(L214N1  '(OR 4))
(L215    '(OR 5))
(L215N1  '(OR 5))
(L216    '(OR 6))
(L218    '(OR 8))
(L302    '(NAND 2))
(L302N1  '(NAND 2))
(L303    '(NAND 3))
(L303N1  '(NAND 3))
(L303N2  '(NAND 3))
(L304    '(NAND 4))
(L304N1  '(NAND 4))
(L304N2  '(NAND 4))
(L305    '(NAND 5))
(L305N1  '(NAND 5))
(L305N2  '(NAND 5))
(L305N3  '(NAND 5))
(L306    '(NAND 6))
(L308    '(NAND 8))
(L312    '(AND 2))
(L313    '(AND 3))
(L314    '(AND 4))
(L314N1  '(AND 4))
(L315    '(AND 5))
```

```
(L315N1  '(AND 5))
(L316    '(AND 6))
(L318    '(AND 8))
(L401    '(EXOR-AND 3))
(L401NC  '(EXOR-AND 3))
(L401ND  '(EXOR-AND 3))
(L411    '(AND-EXOR 3))
(L421    '(AND-NOR 3))
(L421NA  '(AND-NOR 3))
(L421NB  '(AND-NOR 3))
(L421NC  '(AND-NOR 3))
(L421ND  '(AND-NOR 3))
(L421NE  '(AND-NOR 3))
(L422    '(AND-NOR 4))
(L422NA  '(AND-NOR 4))
(L422NB  '(AND-NOR 4))
(L422NC  '(AND-NOR 4))
(L422ND  '(AND-NOR 4))
(L422NF  '(AND-NOR 4))
(L422NG  '(AND-NOR 4))
(L422NH  '(AND-NOR 4))
(L423    '(AND-NOR 4))
(L423NA  '(AND-NOR 4))
(L423NB  '(AND-NOR 4))
(L423NC  '(AND-NOR 4))
(L423ND  '(AND-NOR 4))
(L423NE  '(AND-NOR 4))
(L423NF  '(AND-NOR 4))
(L423NG  '(AND-NOR 4))
(L424    '(AND-NOR 4))
(L424NA  '(AND-NOR 4))
(L424NB  '(AND-NOR 4))
(L424NC  '(AND-NOR 4))
(L424ND  '(AND-NOR 4))
(L424NE  '(AND-NOR 4))
(L425    '(AND-NOR 6))
(L425NF  '(AND-NOR 6))
(L426    '(AND-NOR 6))
(L427    '(AND-NOR 5))
(L427ND  '(AND-NOR 5))
(L427NE  '(AND-NOR 5))
(L428    '(AND-NOR 5))
(L428NE  '(AND-NOR 5))
(L429    '(AND-NOR 8))
(L431    '(OR-NAND 3))
(L431NA  '(OR-NAND 3))
(L431NB  '(OR-NAND 3))
(L431NC  '(OR-NAND 3))
(L431ND  '(OR-NAND 3))
(L431NE  '(OR-NAND 3))
(L432    '(OR-NAND 4))
(L432NA  '(OR-NAND 4))
(L432NB  '(OR-NAND 4))
(L432ND  '(OR-NAND 4))
(L432NE  '(OR-NAND 4))
(L432NF  '(OR-NAND 4))
(L432NG  '(OR-NAND 4))
(L432NH  '(OR-NAND 4))
(L433    '(OR-NAND 4))
(L433NA  '(OR-NAND 4))
(L433NC  '(OR-NAND 4))
(L433ND  '(OR-NAND 4))
(L433NE  '(OR-NAND 4))
(L433NF  '(OR-NAND 4))
(L433NG  '(OR-NAND 4))
```

```
    (L434    '(OR-NAND 4))
    (L434NA  '(OR-NAND 4))
    (L434NB  '(OR-NAND 4))
    (L434NC  '(OR-NAND 4))
    (L434ND  '(OR-NAND 4))
    (L434NE  '(OR-NAND 4))
    (L435    '(OR-NAND 5))
    (L435ND  '(OR-NAND 5))
    (L435NE  '(OR-NAND 5))
    (L436    '(OR-NAND 6))
    (L437    '(OR-NAND 5))
    (L438    '(OR-NAND 6))
    (L437NE  '(OR-NAND 5))
    (L438NF  '(OR-NAND 6))
    (L442    '(OR-NAND 8))
    (L454    '(OR-NAND 8))
    (L462    '(OR-NAND 6))
    (L511    '(XOR 2))
    (L512    '(XOR 2))
    (L516    '(XOR 3))
    (L564    '(ADDER 6))
    (L565    '(ADDER 3))
    (L56G    '(ADDER 5))
    (L57B    '(ADDER 3))
    (L584    '(PARITY 3))
    (L601NQ  '(LATCH 2))
    (L604NQ  '(LATCH 2))
    (L611NQ  '(DFF 2))
    (L611NW  '(DFF 2 2))
    (L611SQ  '(DFF 4))
    (L611TQ  '(DFF 6))
    (L612NQ  '(DFF 3))
    (L612SQ  '(DFF 5))
    (L612TQ  '(DFF 7))
    (L613NQ  '(DFF 3))
    (L613TQ  '(DFF 7))
(WSRAMDHSAA32W33C2 '(RAM))             ;memory
(WSRAMDHSAA32W37C2 '(RAM))
(WSRAMDHSAA32W41C2 '(RAM))

;multiple-outputs
    (F091    '(OUT 0 2))               ;output only
    (F521    '(ADDER 3 2))
    (F521NS  '(ADDER 3 2))
    (F611    '(DFF 2 2))
    (F615    '(DFF 3 2))
    (F771    '(DFF 3 2))
    (F775    '(DFF 4 2))
    (F791    '(DFF 4 2))
    (L521    '(ADDER 3 2))
    (L522    '(ADDER 2 2))
    (L528    '(ADDER 3 2))
    (L603NW  '(LATCH 3 2))
    (L615NW  '(DFF 3 2))
    (L631NW  '(DFF 2 2))
    (T       (list type))) )          ;default
```