ECEN 714: Submission for LAB 8

Manav Gurumoorthy
830000011

Contents (Area and Constraint report for the Alarm clock and Cruise Control, STA for the Cruise Control)

Constrain Report

Alarm Clock

```
Information: Updating design information... (UID-85)

****************************************
Report : constraint
        -all_violators
        -verbose
Design : TOP
Version: O-2018.06-SP3
Date   : Fri Nov  8 20:09:06 2019
****************************************

This design has no violated constraints.



****************************************
Report : area
Design : TOP
Version: O-2018.06-SP3
Date   : Fri Nov  8 20:10:07 2019
****************************************

Library(s) Used:

    class (File: /opt/coe/synopsys/syn/O-2018.06-SP3/libraries/syn/class.db)

Number of ports:                          261
Number of nets:                           728
Number of cells:                          492
Number of combinational cells:            444
Number of sequential cells:                33
Number of macros/black boxes:               0
Number of buf/inv:                        104
Number of references:                       6

Combinational area:               707.000000
Buf/Inv area:                     104.000000
Noncombinational area:            231.000000
```

```
Macro/Black Box area:                 0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                    938.000000
Total area:                 undefined
```

No of registers: 33

Cruise Control

```
Information: Updating design information... (UID-85)
Information: Timing loop detected. (OPT-150)
      speed_reg[5]/CLK speed_reg[5]/Q U223/A U223/Y U493/A U493/Y U491/B U491/Y U409/A U409/Y cruisespeed_reg[3]/CLK
cruisespeed_reg[3]/Q U485/A U485/Y U480/A U480/Y U470/A U470/Y U469/A U469/Y U467/B U467/Y U466/C U466/Y
Information: Timing loop detected. (OPT-150)
      speed_reg[4]/CLK speed_reg[4]/Q U494/A U494/Y U493/B U493/Y U491/B U491/Y U409/A U409/Y cruisespeed_reg[3]/CLK
cruisespeed_reg[3]/Q U485/A U485/Y U480/A U480/Y U470/A U470/Y U469/A U469/Y U467/B U467/Y U466/C U466/Y
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell 'speed_reg[5]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell 'speed_reg[4]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell 'speed_reg[3]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell 'speed_reg[0]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell 'speed_reg[1]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell 'speed_reg[2]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell 'speed_reg[7]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell 'speed_reg[6]'
         to break a timing loop. (OPT-314)

****************************************
Report : constraint
        -all_violators
        -verbose
Design : cruise_control
Version: O-2018.06-SP3
Date   : Fri Nov  8 20:41:58 2019
****************************************
```

```
This design has no violated constraints.



****************************************
Report : area
Design : cruise_control
Version: O-2018.06-SP3
Date   : Fri Nov  8 20:42:24 2019
****************************************

Library(s) Used:

    iit018_stdcells (File: /home/grads/m/mana/tutorial/iit018_stdcells.db)

Number of ports:                           58
Number of nets:                           424
Number of cells:                          373
Number of combinational cells:            348
Number of sequential cells:                23
Number of macros/black boxes:               0
Number of buf/inv:                         91
Number of references:                      21

Combinational area:              11447.000000
Buf/Inv area:                     1864.000000
Noncombinational area:             288.000000
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                 11735.000000
Total area:                 undefined
```

No of registers: 23

STA for Cruise Control

```
****************************************
Report : constraint
      -all_violators
```

```
      -verbose
      -max_delay
Design : cruise_control
Version: O-2018.06-SP3
Date   : Mon Nov 11 12:54:44 2019
****************************************
1
```

Min Delay

```
****************************************
Report : timing
      -path_type full
      -delay_type min
      -slack_lesser_than 5.00
      -max_paths 3
      -sort_by slack
Design : cruise_control
Version: O-2018.06-SP3
Date   : Mon Nov 11 13:02:35 2019
****************************************


  Startpoint: reset (input port clocked by clk)
  Endpoint: state_reg[2]
              (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: min

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  input external delay                     0.00       0.00 f
  reset (in)                               0.01       0.01 f
  U530/Y (INVX1)                           0.09       0.10 r
  U527/Y (AND2X1)                          0.07       0.17 r
  state_reg[2]/D (DFFPOSX1)                0.00       0.17 r
  data arrival time                                   0.17

  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  clock reconvergence pessimism            0.00       0.00
  state_reg[2]/CLK (DFFPOSX1)                         0.00 r
```

```
library hold time                          0.00       0.00
data required time                                    0.00
-----------------------------------------------------------
data required time                                    0.00
data arrival time                                    -0.17
-----------------------------------------------------------
slack (MET)                                           0.17


Startpoint: reset (input port clocked by clk)
Endpoint: state_reg[1]
             (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: min

Point                                      Incr       Path
-----------------------------------------------------------
clock clk (rise edge)                      0.00       0.00
clock network delay (ideal)                0.00       0.00
input external delay                       0.00       0.00 f
reset (in)                                 0.01       0.01 f
U530/Y (INVX1)                             0.09       0.10 r
U528/Y (AND2X1)                            0.07       0.17 r
state_reg[1]/D (DFFPOSX1)                  0.00       0.17 r
data arrival time                                     0.17

clock clk (rise edge)                      0.00       0.00
clock network delay (ideal)                0.00       0.00
clock reconvergence pessimism              0.00       0.00
state_reg[1]/CLK (DFFPOSX1)                           0.00 r
library hold time                          0.00       0.00
data required time                                    0.00
-----------------------------------------------------------
data required time                                    0.00
data arrival time                                    -0.17
-----------------------------------------------------------
slack (MET)                                           0.17


Startpoint: reset (input port clocked by clk)
Endpoint: state_reg[0]
             (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
```

```
  Path Type: min

  Point                                   Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                   0.00       0.00
  clock network delay (ideal)             0.00       0.00
  input external delay                    0.00       0.00 f
  reset (in)                              0.01       0.01 f
  U530/Y (INVX1)                          0.09       0.10 r
  U529/Y (AND2X1)                         0.07       0.17 r
  state_reg[0]/D (DFFPOSX1)               0.00       0.17 r
  data arrival time                                  0.17

  clock clk (rise edge)                   0.00       0.00
  clock network delay (ideal)             0.00       0.00
  clock reconvergence pessimism           0.00       0.00
  state_reg[0]/CLK (DFFPOSX1)                        0.00 r
  library hold time                       0.00       0.00
  data required time                                 0.00
  -----------------------------------------------------------
  data required time                                 0.00
  data arrival time                                 -0.17
  -----------------------------------------------------------
  slack (MET)                                        0.17


1
```

Max Delay

```
****************************************
Report : timing
      -path_type full
      -delay_type max
      -slack_lesser_than 6.00
      -max_paths 3
      -sort_by slack
Design : cruise_control
Version: O-2018.06-SP3
Date   : Mon Nov 11 13:03:02 2019
****************************************
```

```
Startpoint: reset (input port clocked by clk)
Endpoint: state_reg[2]
             (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Point                                    Incr       Path
-----------------------------------------------------------
clock clk (rise edge)                    0.00       0.00
clock network delay (ideal)              0.00       0.00
input external delay                     4.00       4.00 f
reset (in)                               0.01       4.01 f
U530/Y (INVX1)                           0.09       4.10 r
U527/Y (AND2X1)                          0.07       4.17 r
state_reg[2]/D (DFFPOSX1)                0.00       4.17 r
data arrival time                                   4.17

clock clk (rise edge)                   10.00      10.00
clock network delay (ideal)              0.00      10.00
clock reconvergence pessimism            0.00      10.00
state_reg[2]/CLK (DFFPOSX1)                        10.00 r
library setup time                      -0.19       9.81
data required time                                  9.81
-----------------------------------------------------------
data required time                                  9.81
data arrival time                                  -4.17
-----------------------------------------------------------
slack (MET)                                         5.64


Startpoint: reset (input port clocked by clk)
Endpoint: state_reg[1]
             (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Point                                    Incr       Path
-----------------------------------------------------------
clock clk (rise edge)                    0.00       0.00
clock network delay (ideal)              0.00       0.00
input external delay                     4.00       4.00 f
reset (in)                               0.01       4.01 f
U530/Y (INVX1)                           0.09       4.10 r
```

```
U528/Y (AND2X1)                          0.07       4.17 r
state_reg[1]/D (DFFPOSX1)                0.00       4.17 r
data arrival time                                   4.17

clock clk (rise edge)                   10.00      10.00
clock network delay (ideal)              0.00      10.00
clock reconvergence pessimism            0.00      10.00
state_reg[1]/CLK (DFFPOSX1)                        10.00 r
library setup time                      -0.19       9.81
data required time                                  9.81
-----------------------------------------------------------
data required time                                  9.81
data arrival time                                  -4.17
-----------------------------------------------------------
slack (MET)                                         5.64


Startpoint: reset (input port clocked by clk)
Endpoint: state_reg[0]
            (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Point                                    Incr       Path
-----------------------------------------------------------
clock clk (rise edge)                    0.00       0.00
clock network delay (ideal)              0.00       0.00
input external delay                     4.00       4.00 f
reset (in)                               0.01       4.01 f
U530/Y (INVX1)                           0.09       4.10 r
U529/Y (AND2X1)                          0.07       4.17 r
state_reg[0]/D (DFFPOSX1)                0.00       4.17 r
data arrival time                                   4.17

clock clk (rise edge)                   10.00      10.00
clock network delay (ideal)              0.00      10.00
clock reconvergence pessimism            0.00      10.00
state_reg[0]/CLK (DFFPOSX1)                        10.00 r
library setup time                      -0.19       9.81
data required time                                  9.81
-----------------------------------------------------------
data required time                                  9.81
data arrival time                                  -4.17
```

```
  -----------------------------------------------------------
  slack (MET)                                      5.64


1
```

Synthesised Netlists

Alarm

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version   : O-2018.06-SP3
// Date      : Fri Nov  8 20:11:22 2019
////////////////////////////////////////////////////////////


module TIME_STATE_MACHINE ( TIME_BUTTON, HOURS_BUTTON, MINUTES_BUTTON, CLK,
        SECS, HOURS, MINS );
  input TIME_BUTTON, HOURS_BUTTON, MINUTES_BUTTON, CLK;
  output SECS, HOURS, MINS;
  wire   n1, n2, n7, n8, n9, n3, n4, n5, n6;
  wire   [1:0] CURRENT_STATE;
  wire   [1:0] NEXT_STATE;

  FD1 \CURRENT_STATE_reg[0]  ( .D(NEXT_STATE[0]), .CP(CLK), .Q(
        CURRENT_STATE[0]), .QN(n2) );
  FD1 \CURRENT_STATE_reg[1]  ( .D(NEXT_STATE[1]), .CP(CLK), .Q(
        CURRENT_STATE[1]), .QN(n1) );
  NR2I U4 ( .A(CURRENT_STATE[0]), .B(n5), .Z(n9) );
  AN3 U12 ( .A(n2), .B(n1), .C(n7), .Z(MINS) );
  AN3 U13 ( .A(TIME_BUTTON), .B(n6), .C(MINUTES_BUTTON), .Z(n7) );
  AN3 U14 ( .A(n2), .B(n1), .C(n8), .Z(HOURS) );
  AO4 U3 ( .A(n7), .B(n1), .C(n8), .D(n9), .Z(SECS) );
  AO6 U5 ( .A(CURRENT_STATE[0]), .B(n1), .C(n5), .Z(NEXT_STATE[1]) );
  AO6 U6 ( .A(CURRENT_STATE[1]), .B(n2), .C(n3), .Z(NEXT_STATE[0]) );
  NR3 U7 ( .A(n6), .B(MINUTES_BUTTON), .C(n4), .Z(n8) );
  IVI U8 ( .A(n8), .Z(n3) );
  IVI U9 ( .A(TIME_BUTTON), .Z(n4) );
```

```
  IVI U10 ( .A(n7), .Z(n5) );
  IVI U11 ( .A(HOURS_BUTTON), .Z(n6) );
endmodule


module TIME_COUNTER ( HOURS, MINS, SECS, CLK, HOURS_OUT, MINUTES_OUT,
        AM_PM_OUT );
  output [3:0] HOURS_OUT;
  output [5:0] MINUTES_OUT;
  input HOURS, MINS, SECS, CLK;
  output AM_PM_OUT;
  wire   N28, N29, N30, N33, N34, N35, N36, N37, N50, N51, N52, N53, N54, N65,
         N66, N67, N84, N85, N86, n11, n12, n20, n27, n32, n33, n34, n35, n36,
         n37, n38, n39, n40, n41, n42, n43, n44, n45, n46, n47, n48, n49, n50,
         n51, n52, n53, n54, n55, n56, n57, n58, n59, n60, n61, n62, n63, n64,
         n65, n66, n67, n68, n69, n70, n71, n72, n73, n74, n75, n76, n77, n78,
         n79, n80, N168, N164, N160, \mult_add_34_aco/PROD_not[2] ,
         \mult_add_34_aco/B[0] , \mult_add_54_aco/PROD_not[2] ,
         \mult_add_68_aco/PROD_not[2] , n1, n2, n3, n4, n5, n6, n7, n8, n9,
         n10, n13, n14, n15, n16, n17, n18, n19, n21, n22, n23, n24, n25, n26,
         n28, n29, n30, n31, n81, n82, n83, n84, n85, n86, n87, n88, n89, n90,
         n91, n92, n93, n94, n95, n96, n97, n98, n99, n100, n101, n102, n103;
  wire   [5:0] CURRENT_SECS;

  FD1 \CURRENT_SECS_reg[0]  ( .D(n23), .CP(CLK), .Q(CURRENT_SECS[0]), .QN(n7)
        );
  FD1 \MINUTES_OUT_reg[4]  ( .D(n25), .CP(CLK), .Q(MINUTES_OUT[4]), .QN(n4) );
  FD1 \MINUTES_OUT_reg[0]  ( .D(n26), .CP(CLK), .Q(MINUTES_OUT[0]), .QN(n6) );
  FD1 \MINUTES_OUT_reg[1]  ( .D(n28), .CP(CLK), .Q(MINUTES_OUT[1]) );
  FD1 \MINUTES_OUT_reg[2]  ( .D(n29), .CP(CLK), .Q(MINUTES_OUT[2]), .QN(n20)
        );
  FD1 \MINUTES_OUT_reg[3]  ( .D(n30), .CP(CLK), .Q(MINUTES_OUT[3]) );
  FD1 \MINUTES_OUT_reg[5]  ( .D(n31), .CP(CLK), .Q(MINUTES_OUT[5]) );
  FD1 \HOURS_OUT_reg[3]  ( .D(n84), .CP(CLK), .Q(HOURS_OUT[3]), .QN(n11) );
  FD1 \HOURS_OUT_reg[2]  ( .D(n83), .CP(CLK), .Q(HOURS_OUT[2]), .QN(n12) );
  FD1 \HOURS_OUT_reg[1]  ( .D(n82), .CP(CLK), .Q(HOURS_OUT[1]) );
  FD1 \HOURS_OUT_reg[0]  ( .D(n81), .CP(CLK), .Q(HOURS_OUT[0]) );
  FD1 \CURRENT_SECS_reg[1]  ( .D(n24), .CP(CLK), .Q(CURRENT_SECS[1]) );
  FD1 \CURRENT_SECS_reg[2]  ( .D(n22), .CP(CLK), .Q(CURRENT_SECS[2]), .QN(n27)
        );
  FD1 \CURRENT_SECS_reg[3]  ( .D(n18), .CP(CLK), .Q(CURRENT_SECS[3]) );
  FD1 \CURRENT_SECS_reg[4]  ( .D(n19), .CP(CLK), .Q(CURRENT_SECS[4]), .QN(n5)
        );
```

```
FD1 \CURRENT_SECS_reg[5]  ( .D(n21), .CP(CLK), .Q(CURRENT_SECS[5]) );
FD1 AM_PM_OUT_reg ( .D(n85), .CP(CLK), .Q(AM_PM_OUT) );
AO1P U4 ( .A(n35), .B(n36), .C(n86), .D(AM_PM_OUT), .Z(n34) );
ND2I U14 ( .A(n54), .B(n55), .Z(n53) );
ND2I U15 ( .A(N67), .B(n38), .Z(n55) );
ND2I U18 ( .A(n58), .B(n59), .Z(n57) );
ND2I U19 ( .A(N66), .B(n38), .Z(n59) );
ND2I U22 ( .A(n62), .B(n63), .Z(n61) );
ND2I U23 ( .A(N65), .B(n38), .Z(n63) );
ND2I U26 ( .A(n66), .B(n67), .Z(n65) );
ND2I U27 ( .A(n91), .B(n38), .Z(n67) );
NR2I U29 ( .A(n41), .B(n68), .Z(n39) );
ND2I U31 ( .A(n42), .B(n36), .Z(n52) );
NR2I U39 ( .A(n71), .B(n87), .Z(n72) );
NR2I U43 ( .A(n41), .B(n69), .Z(n45) );
AN3 U70 ( .A(MINUTES_OUT[4]), .B(MINUTES_OUT[3]), .C(MINUTES_OUT[5]), .Z(n78) );
AN4 U71 ( .A(CURRENT_SECS[0]), .B(n27), .C(CURRENT_SECS[1]), .D(n80), .Z(n69) );
AN3 U72 ( .A(CURRENT_SECS[4]), .B(CURRENT_SECS[3]), .C(CURRENT_SECS[5]), .Z(
      n80) );
OR3 U73 ( .A(MINS), .B(SECS), .C(n96), .Z(n36) );
OR3 U74 ( .A(HOURS), .B(MINS), .C(n97), .Z(n41) );
AN3 U75 ( .A(n96), .B(n97), .C(MINS), .Z(n38) );
AN2I U3 ( .A(HOURS_OUT[0]), .B(\mult_add_34_aco/B[0] ), .Z(n1) );
AN2I U5 ( .A(HOURS_OUT[1]), .B(\mult_add_34_aco/B[0] ), .Z(n2) );
AN2I U6 ( .A(\mult_add_34_aco/B[0] ), .B(HOURS_OUT[3]), .Z(n3) );
NR3 U7 ( .A(n38), .B(n95), .C(n94), .Z(n44) );
AO7 U8 ( .A(n87), .B(n41), .C(n52), .Z(n51) );
AO7 U9 ( .A(\mult_add_34_aco/B[0] ), .B(n36), .C(n42), .Z(n37) );
AO2 U10 ( .A(N30), .B(n39), .C(N86), .D(n94), .Z(n54) );
AO6 U11 ( .A(n69), .B(n95), .C(n38), .Z(n71) );
AO2 U12 ( .A(n69), .B(n95), .C(n87), .D(n38), .Z(n42) );
AO2 U13 ( .A(N29), .B(n39), .C(N85), .D(n94), .Z(n58) );
AO7 U16 ( .A(n38), .B(n39), .C(n40), .Z(n35) );
AO2 U17 ( .A(N28), .B(n39), .C(N84), .D(n94), .Z(n62) );
AO2 U20 ( .A(n93), .B(n39), .C(n89), .D(n94), .Z(n66) );
NR4 U21 ( .A(n11), .B(n12), .C(HOURS_OUT[0]), .D(HOURS_OUT[1]), .Z(n40) );
AO6 U24 ( .A(n33), .B(AM_PM_OUT), .C(n34), .Z(n32) );
AO3 U25 ( .A(n41), .B(n87), .C(n40), .D(n37), .Z(n33) );
AO2 U28 ( .A(CURRENT_SECS[5]), .B(n44), .C(N54), .D(n45), .Z(n43) );
AO2 U30 ( .A(CURRENT_SECS[4]), .B(n44), .C(N53), .D(n45), .Z(n46) );
AO2 U32 ( .A(CURRENT_SECS[3]), .B(n44), .C(N52), .D(n45), .Z(n47) );
AO2 U33 ( .A(n44), .B(CURRENT_SECS[2]), .C(N51), .D(n45), .Z(n48) );
AO2 U34 ( .A(CURRENT_SECS[1]), .B(n44), .C(N50), .D(n45), .Z(n49) );
```

```
AO2 U35 ( .A(n51), .B(HOURS_OUT[0]), .C(n52), .D(n65), .Z(n64) );
AO2 U36 ( .A(n51), .B(HOURS_OUT[1]), .C(n52), .D(n61), .Z(n60) );
AO2 U37 ( .A(n51), .B(HOURS_OUT[2]), .C(n52), .D(n57), .Z(n56) );
AO2 U38 ( .A(MINUTES_OUT[5]), .B(n71), .C(N37), .D(n72), .Z(n70) );
AO2 U40 ( .A(n71), .B(MINUTES_OUT[2]), .C(N34), .D(n72), .Z(n74) );
AO2 U41 ( .A(MINUTES_OUT[0]), .B(n71), .C(n6), .D(n72), .Z(n76) );
AO2 U42 ( .A(MINUTES_OUT[4]), .B(n71), .C(N36), .D(n72), .Z(n77) );
AO2 U44 ( .A(CURRENT_SECS[0]), .B(n44), .C(n7), .D(n45), .Z(n79) );
AO2 U45 ( .A(MINUTES_OUT[1]), .B(n71), .C(N33), .D(n72), .Z(n75) );
ND4 U46 ( .A(MINUTES_OUT[0]), .B(n20), .C(MINUTES_OUT[1]), .D(n78), .Z(n68)
      );
AO2 U47 ( .A(MINUTES_OUT[3]), .B(n71), .C(N35), .D(n72), .Z(n73) );
AO2 U48 ( .A(n51), .B(HOURS_OUT[3]), .C(n52), .D(n53), .Z(n50) );
ND2 U49 ( .A(HOURS_OUT[2]), .B(\mult_add_34_aco/B[0] ), .Z(
      \mult_add_68_aco/PROD_not[2] ) );
IVI U50 ( .A(\mult_add_68_aco/PROD_not[2] ), .Z(N160) );
ND2 U51 ( .A(HOURS_OUT[2]), .B(\mult_add_34_aco/B[0] ), .Z(
      \mult_add_54_aco/PROD_not[2] ) );
IVI U52 ( .A(\mult_add_54_aco/PROD_not[2] ), .Z(N164) );
ND2 U53 ( .A(HOURS_OUT[2]), .B(\mult_add_34_aco/B[0] ), .Z(
      \mult_add_34_aco/PROD_not[2] ) );
IVI U54 ( .A(\mult_add_34_aco/PROD_not[2] ), .Z(N168) );
EN U55 ( .A(MINUTES_OUT[1]), .B(n6), .Z(N33) );
ND2 U56 ( .A(MINUTES_OUT[1]), .B(MINUTES_OUT[0]), .Z(n8) );
EN U57 ( .A(MINUTES_OUT[2]), .B(n8), .Z(N34) );
AN3 U58 ( .A(MINUTES_OUT[1]), .B(MINUTES_OUT[0]), .C(MINUTES_OUT[2]), .Z(n9)
      );
EO U59 ( .A(MINUTES_OUT[3]), .B(n9), .Z(N35) );
ND2 U60 ( .A(MINUTES_OUT[3]), .B(n9), .Z(n10) );
EN U61 ( .A(MINUTES_OUT[4]), .B(n10), .Z(N36) );
NR2 U62 ( .A(n10), .B(n4), .Z(n13) );
EO U63 ( .A(MINUTES_OUT[5]), .B(n13), .Z(N37) );
EN U64 ( .A(CURRENT_SECS[1]), .B(n7), .Z(N50) );
ND2 U65 ( .A(CURRENT_SECS[1]), .B(CURRENT_SECS[0]), .Z(n14) );
EN U66 ( .A(CURRENT_SECS[2]), .B(n14), .Z(N51) );
AN3 U67 ( .A(CURRENT_SECS[1]), .B(CURRENT_SECS[0]), .C(CURRENT_SECS[2]), .Z(
      n15) );
EO U68 ( .A(CURRENT_SECS[3]), .B(n15), .Z(N52) );
ND2 U69 ( .A(CURRENT_SECS[3]), .B(n15), .Z(n16) );
EN U76 ( .A(CURRENT_SECS[4]), .B(n16), .Z(N53) );
NR2 U77 ( .A(n16), .B(n5), .Z(n17) );
EO U78 ( .A(CURRENT_SECS[5]), .B(n17), .Z(N54) );
EOI U79 ( .A(n3), .B(n98), .Z(N86) );
```

```
NR2I U80 ( .A(n99), .B(n88), .Z(n98) );
EOI U81 ( .A(n88), .B(n99), .Z(N85) );
ND2I U82 ( .A(n2), .B(n1), .Z(n99) );
EOI U83 ( .A(n2), .B(n1), .Z(N84) );
EOI U84 ( .A(n3), .B(n100), .Z(N67) );
NR2I U85 ( .A(n101), .B(n90), .Z(n100) );
EOI U86 ( .A(n90), .B(n101), .Z(N66) );
ND2I U87 ( .A(n2), .B(n1), .Z(n101) );
EOI U88 ( .A(n2), .B(n1), .Z(N65) );
EOI U89 ( .A(n3), .B(n102), .Z(N30) );
NR2I U90 ( .A(n103), .B(n92), .Z(n102) );
EOI U91 ( .A(n92), .B(n103), .Z(N29) );
ND2I U92 ( .A(n2), .B(n1), .Z(n103) );
EOI U93 ( .A(n2), .B(n1), .Z(N28) );
IVI U94 ( .A(n47), .Z(n18) );
IVI U95 ( .A(n46), .Z(n19) );
IVI U96 ( .A(n43), .Z(n21) );
IVI U97 ( .A(n48), .Z(n22) );
IVI U98 ( .A(n79), .Z(n23) );
IVI U99 ( .A(n49), .Z(n24) );
IVI U100 ( .A(n77), .Z(n25) );
IVI U101 ( .A(n76), .Z(n26) );
IVI U102 ( .A(n75), .Z(n28) );
IVI U103 ( .A(n74), .Z(n29) );
IVI U104 ( .A(n73), .Z(n30) );
IVI U105 ( .A(n70), .Z(n31) );
IVI U106 ( .A(n64), .Z(n81) );
IVI U107 ( .A(n60), .Z(n82) );
IVI U108 ( .A(n56), .Z(n83) );
IVI U109 ( .A(n50), .Z(n84) );
IVI U110 ( .A(n32), .Z(n85) );
IVI U111 ( .A(n37), .Z(n86) );
IVI U112 ( .A(n68), .Z(n87) );
IVI U113 ( .A(N160), .Z(n88) );
IVI U114 ( .A(n1), .Z(n89) );
IVI U115 ( .A(N164), .Z(n90) );
IVI U116 ( .A(n1), .Z(n91) );
IVI U117 ( .A(N168), .Z(n92) );
IVI U118 ( .A(n1), .Z(n93) );
IVI U119 ( .A(n40), .Z(\mult_add_34_aco/B[0] ) );
IVI U120 ( .A(n36), .Z(n94) );
IVI U121 ( .A(n41), .Z(n95) );
IVI U122 ( .A(HOURS), .Z(n96) );
```

```
  IVI U123 ( .A(SECS), .Z(n97) );
endmodule


module TIME_BLOCK ( SET_TIME, HRS, MINS, CLK, CONNECT6, CONNECT7, CONNECT8 );
  output [3:0] CONNECT6;
  output [5:0] CONNECT7;
  input SET_TIME, HRS, MINS, CLK;
  output CONNECT8;
  wire   CONNECT5, CONNECT3, CONNECT4;

  TIME_STATE_MACHINE U1 ( .TIME_BUTTON(SET_TIME), .HOURS_BUTTON(HRS),
        .MINUTES_BUTTON(MINS), .CLK(CLK), .SECS(CONNECT5), .HOURS(CONNECT3),
        .MINS(CONNECT4) );
  TIME_COUNTER U2 ( .HOURS(CONNECT3), .MINS(CONNECT4), .SECS(CONNECT5), .CLK(
        CLK), .HOURS_OUT(CONNECT6), .MINUTES_OUT(CONNECT7), .AM_PM_OUT(
        CONNECT8) );
endmodule


module ALARM_STATE_MACHINE ( ALARM_BUTTON, HOURS_BUTTON, MINUTES_BUTTON, CLK,
        HOURS, MINS );
  input ALARM_BUTTON, HOURS_BUTTON, MINUTES_BUTTON, CLK;
  output HOURS, MINS;
  wire   n1, n2, n5, n6, n3, n4;
  wire   [1:0] CURRENT_STATE;
  wire   [1:0] NEXT_STATE;

  FD1 \CURRENT_STATE_reg[0]  ( .D(NEXT_STATE[0]), .CP(CLK), .Q(
        CURRENT_STATE[0]), .QN(n2) );
  FD1 \CURRENT_STATE_reg[1]  ( .D(NEXT_STATE[1]), .CP(CLK), .Q(
        CURRENT_STATE[1]), .QN(n1) );
  AO1P U4 ( .A(CURRENT_STATE[1]), .B(n2), .C(n3), .D(n6), .Z(NEXT_STATE[0]) );
  ND2I U8 ( .A(n4), .B(ALARM_BUTTON), .Z(n6) );
  NR3 U3 ( .A(CURRENT_STATE[0]), .B(CURRENT_STATE[1]), .C(n5), .Z(MINS) );
  NR4 U5 ( .A(CURRENT_STATE[1]), .B(CURRENT_STATE[0]), .C(n6), .D(n3), .Z(
        HOURS) );
  AO6 U6 ( .A(CURRENT_STATE[0]), .B(n1), .C(n5), .Z(NEXT_STATE[1]) );
  ND3 U7 ( .A(n6), .B(n3), .C(ALARM_BUTTON), .Z(n5) );
  IVI U9 ( .A(HOURS_BUTTON), .Z(n3) );
  IVI U10 ( .A(MINUTES_BUTTON), .Z(n4) );
endmodule
```

```
module ALARM_COUNTER ( HOURS, MINS, CLK, HOURS_OUT, MINUTES_OUT, AM_PM_OUT );
  output [3:0] HOURS_OUT;
  output [5:0] MINUTES_OUT;
  input HOURS, MINS, CLK;
  output AM_PM_OUT;
  wire   N15, N16, N17, N34, N35, N36, N57, n3, n5, n11, n12, n14, n15, n16,
         n18, n19, n20, n21, n22, n23, n24, n25, n30, n31, n32, n33, n34, n35,
         n36, n37, n38, n39, n40, n41, n42, n43, n44, n45, n46, n47, n48, n49,
         n50, n51, n52, N76, N72, \mult_add_42_aco/PROD_not[2] ,
         \mult_add_29_aco/PROD_not[2] , \mult_add_29_aco/PROD_not[3] , n1, n2,
         n4, n6, n7, n8, n9, n10, n13, n17, n26, n27, n28, n29, n53, n54, n55,
         n56, n57, n58, n59, n60, n61, n62;

  FD1 \MINUTES_OUT_reg[0]  ( .D(n42), .CP(CLK), .Q(MINUTES_OUT[0]), .QN(n36)
        );
  FD1 \HOURS_OUT_reg[2]  ( .D(n49), .CP(CLK), .Q(HOURS_OUT[2]), .QN(n44) );
  FD1 \HOURS_OUT_reg[1]  ( .D(n50), .CP(CLK), .Q(HOURS_OUT[1]), .QN(n45) );
  FD1 \HOURS_OUT_reg[0]  ( .D(n51), .CP(CLK), .Q(HOURS_OUT[0]), .QN(n46) );
  FD1 \HOURS_OUT_reg[3]  ( .D(n52), .CP(CLK), .Q(HOURS_OUT[3]), .QN(n43) );
  FD1 \MINUTES_OUT_reg[1]  ( .D(n41), .CP(CLK), .Q(MINUTES_OUT[1]), .QN(n35)
        );
  FD1 \MINUTES_OUT_reg[2]  ( .D(n40), .CP(CLK), .Q(MINUTES_OUT[2]), .QN(n31)
        );
  FD1 \MINUTES_OUT_reg[3]  ( .D(n39), .CP(CLK), .Q(MINUTES_OUT[3]), .QN(n33)
        );
  FD1 \MINUTES_OUT_reg[4]  ( .D(n38), .CP(CLK), .Q(MINUTES_OUT[4]), .QN(n34)
        );
  FD1 \MINUTES_OUT_reg[5]  ( .D(n37), .CP(CLK), .Q(MINUTES_OUT[5]), .QN(n32)
        );
  FD1 AM_PM_OUT_reg ( .D(n48), .CP(CLK), .Q(AM_PM_OUT), .QN(n47) );
  ND2I U14 ( .A(n3), .B(n11), .Z(n5) );
  ENI U16 ( .A(n12), .B(n47), .Z(n48) );
  NR2I U17 ( .A(n28), .B(N57), .Z(n12) );
  ND2I U19 ( .A(N35), .B(n56), .Z(n16) );
  ND2I U20 ( .A(N16), .B(n18), .Z(n15) );
  ND2I U22 ( .A(N34), .B(n56), .Z(n20) );
  ND2I U23 ( .A(N15), .B(n18), .Z(n19) );
  ND2I U25 ( .A(n54), .B(n56), .Z(n22) );
  ND2I U26 ( .A(n55), .B(n18), .Z(n21) );
  ND2I U28 ( .A(N36), .B(n56), .Z(n24) );
  ND2I U30 ( .A(N17), .B(n18), .Z(n23) );
  NR2I U31 ( .A(n57), .B(n28), .Z(n18) );
```

```
ND2I U34 ( .A(HOURS), .B(n58), .Z(n25) );
NR2I U40 ( .A(n58), .B(HOURS), .Z(n3) );
AO4 U3 ( .A(n35), .B(n3), .C(n10), .D(n5), .Z(n41) );
AN2I U4 ( .A(HOURS_OUT[0]), .B(N57), .Z(n1) );
AN2I U5 ( .A(HOURS_OUT[1]), .B(N57), .Z(n2) );
ENI U6 ( .A(MINUTES_OUT[5]), .B(n27), .Z(n4) );
AN2I U7 ( .A(N57), .B(HOURS_OUT[3]), .Z(n6) );
EOI U8 ( .A(MINUTES_OUT[4]), .B(n26), .Z(n7) );
ENI U9 ( .A(MINUTES_OUT[3]), .B(n17), .Z(n8) );
EOI U10 ( .A(MINUTES_OUT[2]), .B(n13), .Z(n9) );
AO7 U11 ( .A(n57), .B(n11), .C(n25), .Z(n14) );
AO3 U12 ( .A(n46), .B(n14), .C(n21), .D(n22), .Z(n51) );
AO3 U13 ( .A(n45), .B(n14), .C(n19), .D(n20), .Z(n50) );
AO3 U15 ( .A(n44), .B(n14), .C(n15), .D(n16), .Z(n49) );
ND4 U18 ( .A(MINUTES_OUT[0]), .B(MINUTES_OUT[1]), .C(n31), .D(n30), .Z(n11)
      );
NR3 U21 ( .A(n34), .B(n32), .C(n33), .Z(n30) );
AO4 U24 ( .A(n32), .B(n3), .C(n4), .D(n5), .Z(n37) );
AO4 U27 ( .A(n34), .B(n3), .C(n7), .D(n5), .Z(n38) );
AO4 U29 ( .A(n33), .B(n3), .C(n8), .D(n5), .Z(n39) );
AO4 U32 ( .A(n31), .B(n3), .C(n9), .D(n5), .Z(n40) );
EOI U33 ( .A(n35), .B(MINUTES_OUT[0]), .Z(n10) );
AO4 U35 ( .A(n36), .B(n3), .C(MINUTES_OUT[0]), .D(n5), .Z(n42) );
ENI U36 ( .A(\mult_add_29_aco/PROD_not[3] ), .B(n61), .Z(N17) );
AO3 U37 ( .A(n43), .B(n14), .C(n23), .D(n24), .Z(n52) );
ND4 U38 ( .A(n45), .B(n46), .C(HOURS_OUT[3]), .D(HOURS_OUT[2]), .Z(N57) );
ND2 U39 ( .A(HOURS_OUT[2]), .B(N57), .Z(\mult_add_29_aco/PROD_not[2] ) );
ND2 U41 ( .A(N57), .B(HOURS_OUT[3]), .Z(\mult_add_29_aco/PROD_not[3] ) );
IVI U42 ( .A(\mult_add_29_aco/PROD_not[2] ), .Z(N76) );
ND2 U43 ( .A(HOURS_OUT[2]), .B(N57), .Z(\mult_add_42_aco/PROD_not[2] ) );
IVI U44 ( .A(\mult_add_42_aco/PROD_not[2] ), .Z(N72) );
ND2 U45 ( .A(MINUTES_OUT[1]), .B(MINUTES_OUT[0]), .Z(n13) );
AN3 U46 ( .A(MINUTES_OUT[1]), .B(MINUTES_OUT[0]), .C(MINUTES_OUT[2]), .Z(n17) );
ND2 U47 ( .A(MINUTES_OUT[3]), .B(n17), .Z(n26) );
NR2 U48 ( .A(n26), .B(n34), .Z(n27) );
EOI U49 ( .A(n6), .B(n59), .Z(N36) );
NR2I U50 ( .A(n60), .B(n29), .Z(n59) );
EOI U51 ( .A(n29), .B(n60), .Z(N35) );
ND2I U52 ( .A(n2), .B(n1), .Z(n60) );
EOI U53 ( .A(n2), .B(n1), .Z(N34) );
NR2I U54 ( .A(n62), .B(n53), .Z(n61) );
EOI U55 ( .A(n53), .B(n62), .Z(N16) );
ND2I U56 ( .A(n2), .B(n1), .Z(n62) );
```

```
  EOI U57 ( .A(n2), .B(n1), .Z(N15) );
  IVI U58 ( .A(n14), .Z(n28) );
  IVI U59 ( .A(N72), .Z(n29) );
  IVI U60 ( .A(N76), .Z(n53) );
  IVI U61 ( .A(n1), .Z(n54) );
  IVI U62 ( .A(n1), .Z(n55) );
  IVI U63 ( .A(n25), .Z(n56) );
  IVI U64 ( .A(n3), .Z(n57) );
  IVI U65 ( .A(MINS), .Z(n58) );
endmodule


module ALARM_BLOCK ( ALARM, HRS, MINS, CLK, CONNECT9, CONNECT10, CONNECT11 );
  output [3:0] CONNECT9;
  output [5:0] CONNECT10;
  input ALARM, HRS, MINS, CLK;
  output CONNECT11;
  wire   CONNECT1, CONNECT2;

  ALARM_STATE_MACHINE U0 ( .ALARM_BUTTON(ALARM), .HOURS_BUTTON(HRS),
        .MINUTES_BUTTON(MINS), .CLK(CLK), .HOURS(CONNECT1), .MINS(CONNECT2) );
  ALARM_COUNTER U3 ( .HOURS(CONNECT1), .MINS(CONNECT2), .CLK(CLK), .HOURS_OUT(
        CONNECT9), .MINUTES_OUT(CONNECT10), .AM_PM_OUT(CONNECT11) );
endmodule


module CONVERTOR_0 ( T0, T1, T2, T3, T4, T5, A0, B0, C0, D0, E0, F0, G0, A1,
        B1, C1, D1, E1, F1, G1 );
  input T0, T1, T2, T3, T4, T5;
  output A0, B0, C0, D0, E0, F0, G0, A1, B1, C1, D1, E1, F1, G1;
  wire   net_22, net_24, net_28, net_31, net_33, net_38, net_39, net_40,
         net_44, net_48, net_49, net_51, net_60, net_62, net_63, net_68,
         net_70, net_74, net_75, net_77, net_81, net_82, net_83, net_84,
         net_85, net_87, net_93, net_94, net_96, net_103, n1, n2, n3, n4, n5,
         n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20,
         n21;

  ND2I U9 ( .A(net_77), .B(net_40), .Z(net_39) );
  NR2I U16 ( .A(T5), .B(net_51), .Z(E1) );
  AO7P U21 ( .A(T3), .B(net_60), .C(n10), .Z(E0) );
  ND2I U23 ( .A(net_63), .B(n6), .Z(D1) );
  NR2I U30 ( .A(n11), .B(net_60), .Z(net_44) );
  ND2I U34 ( .A(T3), .B(net_70), .Z(net_75) );
```

```
ND2I U37 ( .A(T2), .B(n11), .Z(net_77) );
ND2I U46 ( .A(T3), .B(n13), .Z(net_85) );
ND2I U55 ( .A(net_49), .B(n17), .Z(B0) );
ND2I U60 ( .A(T3), .B(n15), .Z(net_28) );
NR2I U71 ( .A(T2), .B(T4), .Z(net_70) );
ND2I U83 ( .A(n12), .B(n11), .Z(net_103) );
ND2I U85 ( .A(T2), .B(n13), .Z(net_60) );
AO7P U86 ( .A(T2), .B(n11), .C(n14), .Z(net_62) );
ND2I U88 ( .A(T4), .B(n17), .Z(net_93) );
OR2 U111 ( .A(n11), .B(T5), .Z(net_24) );
OR3 U115 ( .A(n9), .B(n5), .C(net_74), .Z(C1) );
AN3 U117 ( .A(n16), .B(n13), .C(n11), .Z(net_49) );
AN2I U2 ( .A(n11), .B(T2), .Z(n1) );
AN2I U3 ( .A(n11), .B(n13), .Z(n2) );
AN2I U4 ( .A(n12), .B(T3), .Z(n3) );
AN2I U5 ( .A(T5), .B(T4), .Z(n4) );
AN2I U6 ( .A(T3), .B(T4), .Z(n5) );
EO1 U7 ( .A(net_68), .B(n16), .C(n16), .D(net_22), .Z(net_63) );
ND3 U8 ( .A(net_22), .B(net_24), .C(n19), .Z(G1) );
NR3 U10 ( .A(n1), .B(n15), .C(n3), .Z(net_51) );
AO6 U11 ( .A(net_83), .B(n16), .C(net_84), .Z(net_82) );
AO2 U12 ( .A(T3), .B(net_38), .C(net_39), .D(n16), .Z(net_33) );
AO2 U13 ( .A(n1), .B(n16), .C(T5), .D(net_96), .Z(net_94) );
ND3 U14 ( .A(n17), .B(net_60), .C(net_85), .Z(net_83) );
AO1 U15 ( .A(T3), .B(net_93), .C(net_44), .D(n12), .Z(net_22) );
AO7 U17 ( .A(n13), .B(T3), .C(net_93), .Z(net_68) );
EON1 U18 ( .A(T2), .B(n4), .C(T4), .D(T2), .Z(net_38) );
ND3 U19 ( .A(T2), .B(T4), .C(T3), .Z(net_40) );
IVI U20 ( .A(net_84), .Z(n6) );
IVI U22 ( .A(net_49), .Z(n7) );
IVI U24 ( .A(B0), .Z(n8) );
IVI U25 ( .A(net_77), .Z(n9) );
IVI U26 ( .A(net_62), .Z(n10) );
IVI U27 ( .A(T3), .Z(n11) );
IVI U28 ( .A(net_60), .Z(n12) );
IVI U29 ( .A(T4), .Z(n13) );
IVI U31 ( .A(n20), .Z(n14) );
IVI U32 ( .A(net_93), .Z(n15) );
IVI U33 ( .A(T5), .Z(n16) );
IVI U35 ( .A(T2), .Z(n17) );
ND2I U36 ( .A(n6), .B(net_94), .Z(A1) );
ND2I U38 ( .A(net_31), .B(net_33), .Z(F1) );
ND2I U39 ( .A(net_75), .B(n16), .Z(net_74) );
```

```
  ND2I U40 ( .A(net_48), .B(n18), .Z(F0) );
  ND2I U41 ( .A(n2), .B(T2), .Z(n18) );
  ND2I U42 ( .A(n11), .B(net_28), .Z(net_96) );
  ND2I U43 ( .A(n10), .B(net_103), .Z(A0) );
  ND2I U44 ( .A(net_93), .B(T3), .Z(net_87) );
  ND2I U45 ( .A(n15), .B(n11), .Z(n19) );
  ND2I U47 ( .A(n21), .B(net_93), .Z(n20) );
  ND2I U48 ( .A(n17), .B(T5), .Z(n21) );
  ND2I U49 ( .A(net_81), .B(net_82), .Z(B1) );
  ND2I U50 ( .A(net_40), .B(n8), .Z(C0) );
  ND2I U51 ( .A(n12), .B(n11), .Z(net_31) );
  ND2I U52 ( .A(T5), .B(net_87), .Z(net_81) );
  ND2I U53 ( .A(net_60), .B(n19), .Z(net_84) );
  ND2I U54 ( .A(n7), .B(n17), .Z(net_48) );
endmodule


module CONVERTOR_1 ( T0, T1, T2, T3, T4, T5, A0, B0, C0, D0, E0, F0, G0, A1,
        B1, C1, D1, E1, F1, G1 );
  input T0, T1, T2, T3, T4, T5;
  output A0, B0, C0, D0, E0, F0, G0, A1, B1, C1, D1, E1, F1, G1;
  wire   A0, n1, n2, n3, n4, n5, n6, n7, n9, n10, n11, n12, n13, n14, n15, n16,
         n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30,
         n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41, n42, n43, n44,
         n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58,
         n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72,
         n73, n74, n75, n76, n77, n78, n79, n80, n81, n82, n83, n84, n85, n86,
         n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n100,
         n101, n102, n103, n104, n105, n106, n107, n108, n109, n110;
  assign D0 = A0;

  NR2I U7 ( .A(n97), .B(n96), .Z(n98) );
  ND2I U9 ( .A(n15), .B(n92), .Z(n93) );
  ND2I U15 ( .A(n7), .B(n82), .Z(n85) );
  NR2I U16 ( .A(T5), .B(n81), .Z(E1) );
  ND2I U23 ( .A(n69), .B(n68), .Z(D1) );
  NR2I U30 ( .A(n21), .B(n72), .Z(n88) );
  NR2I U31 ( .A(n18), .B(n7), .Z(n61) );
  NR2I U32 ( .A(n60), .B(T1), .Z(n97) );
  ND2I U34 ( .A(n56), .B(n62), .Z(n57) );
  ND2I U37 ( .A(n89), .B(n21), .Z(n55) );
  NR2I U38 ( .A(n26), .B(n7), .Z(n89) );
  ND2I U40 ( .A(n4), .B(n92), .Z(n52) );
```

```
ND2I U45 ( .A(n18), .B(n7), .Z(n60) );
ND2I U46 ( .A(n56), .B(n16), .Z(n47) );
AO1P U49 ( .A(n22), .B(n26), .C(n65), .D(n42), .Z(n43) );
AN2I U50 ( .A(n4), .B(n41), .Z(n42) );
ND2I U52 ( .A(n41), .B(n16), .Z(n86) );
ND2I U55 ( .A(n83), .B(n26), .Z(n53) );
NR2I U56 ( .A(n77), .B(T2), .Z(n65) );
ND2I U60 ( .A(T3), .B(n18), .Z(n104) );
ND2I U61 ( .A(n14), .B(T1), .Z(n35) );
ND2I U64 ( .A(n41), .B(T2), .Z(n73) );
NR2I U65 ( .A(n5), .B(T3), .Z(n41) );
ND2I U66 ( .A(n5), .B(n21), .Z(n77) );
AO1P U67 ( .A(n102), .B(n33), .C(n32), .D(n48), .Z(n68) );
ND2I U70 ( .A(n7), .B(n62), .Z(n30) );
NR2I U71 ( .A(T2), .B(T4), .Z(n62) );
NR2I U73 ( .A(n39), .B(T0), .Z(n106) );
NR2I U74 ( .A(n21), .B(n7), .Z(n56) );
NR2I U75 ( .A(n4), .B(T4), .Z(n33) );
NR2I U76 ( .A(T3), .B(T2), .Z(n102) );
ND2I U77 ( .A(n23), .B(n7), .Z(n75) );
ND2I U78 ( .A(T3), .B(n1), .Z(n87) );
ND2I U80 ( .A(n25), .B(n6), .Z(n34) );
NR2I U82 ( .A(n16), .B(n3), .Z(n95) );
ND2I U83 ( .A(n13), .B(n21), .Z(n29) );
ND2I U84 ( .A(n14), .B(n4), .Z(n78) );
ND2I U85 ( .A(T2), .B(n16), .Z(n72) );
ND2I U88 ( .A(T4), .B(n26), .Z(n39) );
ND2I U89 ( .A(n1), .B(n26), .Z(n82) );
OR2 U111 ( .A(n107), .B(T5), .Z(n108) );
AN3 U112 ( .A(n95), .B(n24), .C(T1), .Z(n96) );
AN3 U113 ( .A(n16), .B(n10), .C(T1), .Z(n103) );
OR3 U114 ( .A(n83), .B(n7), .C(n82), .Z(n84) );
OR3 U115 ( .A(n91), .B(n59), .C(n58), .Z(C1) );
OR3 U116 ( .A(n95), .B(T1), .C(n26), .Z(n46) );
AN3 U117 ( .A(n10), .B(n16), .C(n21), .Z(n83) );
AN3 U118 ( .A(T1), .B(n10), .C(n56), .Z(n32) );
AO6 U2 ( .A(n49), .B(n10), .C(n48), .Z(n50) );
IVI U3 ( .A(n87), .Z(n23) );
ND4 U4 ( .A(n47), .B(n46), .C(n60), .D(n78), .Z(n49) );
AO3 U5 ( .A(n4), .B(n86), .C(n51), .D(n50), .Z(B1) );
AO2 U6 ( .A(n37), .B(n10), .C(T5), .D(n36), .Z(n38) );
AO3 U8 ( .A(n39), .B(n75), .C(n68), .D(n38), .Z(A1) );
IVI U10 ( .A(T4), .Z(n16) );
```

```
AO1P U11 ( .A(T4), .B(n22), .C(n65), .D(n64), .Z(n66) );
AO4 U12 ( .A(T3), .B(n44), .C(n1), .D(n30), .Z(n31) );
AO1 U13 ( .A(n23), .B(n61), .C(n88), .D(n13), .Z(n110) );
AO1 U14 ( .A(n1), .B(n53), .C(n65), .D(n52), .Z(n54) );
IVDA U17 ( .A(T0), .Y(n3), .Z(n7) );
OR2I U18 ( .A(n5), .B(n78), .Z(n2) );
ND2I U19 ( .A(n2), .B(n12), .Z(n48) );
AO6 U20 ( .A(n4), .B(n20), .C(n23), .Z(n107) );
AO3 U21 ( .A(n63), .B(n1), .C(n44), .D(n43), .Z(n45) );
AO2 U22 ( .A(n20), .B(n19), .C(n13), .D(n87), .Z(n101) );
ND4 U24 ( .A(n101), .B(n100), .C(n99), .D(n98), .Z(F1) );
AO2 U25 ( .A(n88), .B(n6), .C(n103), .D(n4), .Z(n100) );
AO2 U26 ( .A(n23), .B(n94), .C(n93), .D(n10), .Z(n99) );
AO7 U27 ( .A(n23), .B(n78), .C(n71), .Z(E0) );
AO2 U28 ( .A(n89), .B(n20), .C(n70), .D(n4), .Z(n71) );
AO3 U29 ( .A(n4), .B(n72), .C(n55), .D(n77), .Z(n91) );
AO4 U33 ( .A(n1), .B(n26), .C(n16), .D(n87), .Z(n59) );
ND4 U35 ( .A(n57), .B(n60), .C(n75), .D(n10), .Z(n58) );
AO3 U36 ( .A(n26), .B(n86), .C(n85), .D(n84), .Z(F0) );
AO7 U39 ( .A(n5), .B(n3), .C(n40), .Z(B0) );
AO2 U41 ( .A(n65), .B(n16), .C(n53), .D(n4), .Z(n40) );
ND3 U42 ( .A(n72), .B(n4), .C(n5), .Z(n74) );
AO7 U43 ( .A(n102), .B(n1), .C(n3), .Z(G0) );
AO6 U44 ( .A(n95), .B(T2), .C(n106), .Z(n44) );
AO2 U47 ( .A(n56), .B(n5), .C(T5), .D(n45), .Z(n51) );
AO7 U48 ( .A(n39), .B(n77), .C(n73), .Z(n37) );
ND4 U51 ( .A(n107), .B(n35), .C(n104), .D(n34), .Z(n36) );
IVDA U53 ( .A(T0), .Y(n4), .Z(n6) );
AO7 U54 ( .A(n63), .B(T3), .C(n76), .Z(n64) );
AO3 U57 ( .A(n5), .B(n24), .C(n104), .D(n9), .Z(n105) );
AO2 U58 ( .A(n25), .B(T5), .C(n18), .D(n1), .Z(n27) );
AO6 U59 ( .A(n4), .B(n16), .C(n10), .Z(n90) );
NR3 U62 ( .A(n80), .B(n19), .C(n79), .Z(n81) );
AO3 U63 ( .A(n16), .B(n75), .C(n74), .D(n73), .Z(n80) );
AO4 U68 ( .A(n21), .B(n78), .C(n4), .D(n77), .Z(n79) );
ND4 U69 ( .A(n11), .B(n29), .C(n28), .D(n34), .Z(A0) );
IVDA U72 ( .A(T1), .Y(n1), .Z(n5) );
ND4 U79 ( .A(n12), .B(n110), .C(n109), .D(n108), .Z(G1) );
EO1 U81 ( .A(T5), .B(n67), .C(n66), .D(T5), .Z(n69) );
AO3 U86 ( .A(n1), .B(n104), .C(n17), .D(n110), .Z(n67) );
AO7 U87 ( .A(n95), .B(T2), .C(n5), .Z(n28) );
AO2 U90 ( .A(n106), .B(T1), .C(n6), .D(n105), .Z(n109) );
AO7 U91 ( .A(T2), .B(n21), .C(n27), .Z(n70) );
```

```
  EON1 U92 ( .A(T2), .B(n90), .C(T4), .D(n89), .Z(n94) );
  ND3 U93 ( .A(T2), .B(T4), .C(T3), .Z(n92) );
  AO6 U94 ( .A(n62), .B(n6), .C(n106), .Z(n76) );
  AO2 U95 ( .A(n6), .B(T2), .C(n4), .D(T4), .Z(n63) );
  IVI U96 ( .A(n54), .Z(C0) );
  IVI U97 ( .A(n103), .Z(n9) );
  IVI U98 ( .A(T5), .Z(n10) );
  IVI U99 ( .A(n70), .Z(n11) );
  IVI U100 ( .A(n31), .Z(n12) );
  IVI U101 ( .A(n78), .Z(n13) );
  IVI U102 ( .A(n72), .Z(n14) );
  IVI U103 ( .A(n91), .Z(n15) );
  IVI U104 ( .A(n97), .Z(n17) );
  IVI U105 ( .A(n39), .Z(n18) );
  IVI U106 ( .A(n76), .Z(n19) );
  IVI U107 ( .A(n77), .Z(n20) );
  IVI U108 ( .A(T3), .Z(n21) );
  IVI U109 ( .A(n75), .Z(n22) );
  IVI U110 ( .A(n102), .Z(n24) );
  IVI U119 ( .A(n82), .Z(n25) );
  IVI U120 ( .A(T2), .Z(n26) );
endmodule


module HOURS_FILTER ( TENS_DIGIT_HOURS_IN, TENS_DIGIT_HOURS_OUT );
  input [6:0] TENS_DIGIT_HOURS_IN;
  output [6:0] TENS_DIGIT_HOURS_OUT;
  wire   n7, n8, n1, n2, n3, n4, n5, n6;
  assign TENS_DIGIT_HOURS_OUT[6] = TENS_DIGIT_HOURS_OUT[3];

  NR2I U3 ( .A(n7), .B(n2), .Z(TENS_DIGIT_HOURS_OUT[5]) );
  NR2I U4 ( .A(n7), .B(n3), .Z(TENS_DIGIT_HOURS_OUT[4]) );
  NR2I U5 ( .A(n7), .B(n4), .Z(TENS_DIGIT_HOURS_OUT[3]) );
  NR2I U6 ( .A(n7), .B(n5), .Z(TENS_DIGIT_HOURS_OUT[2]) );
  NR2I U7 ( .A(n7), .B(n1), .Z(TENS_DIGIT_HOURS_OUT[1]) );
  NR4P U8 ( .A(n2), .B(n3), .C(n4), .D(n8), .Z(n7) );
  AN2I U2 ( .A(TENS_DIGIT_HOURS_IN[1]), .B(TENS_DIGIT_HOURS_IN[2]), .Z(n6) );
  IVI U9 ( .A(TENS_DIGIT_HOURS_IN[1]), .Z(n1) );
  IVI U10 ( .A(TENS_DIGIT_HOURS_IN[5]), .Z(n2) );
  IVI U11 ( .A(TENS_DIGIT_HOURS_IN[4]), .Z(n3) );
  IVI U12 ( .A(TENS_DIGIT_HOURS_IN[6]), .Z(n4) );
  IVI U13 ( .A(TENS_DIGIT_HOURS_IN[2]), .Z(n5) );
  ND2I U14 ( .A(n6), .B(TENS_DIGIT_HOURS_IN[6]), .Z(n8) );
```

```
endmodule


module CONVERTOR_CKT ( connect13, disp1, disp2 );
  input [9:0] connect13;
  output [13:0] disp1;
  output [13:0] disp2;
  wire   net2689, net2690;
  wire   [6:0] connect14;
  wire   SYNOPSYS_UNCONNECTED__0;

  CONVERTOR_0 U7 ( .T0(1'b0), .T1(1'b0), .T2(connect13[9]), .T3(connect13[8]),
        .T4(connect13[7]), .T5(connect13[6]), .A0(connect14[6]), .B0(
        connect14[5]), .C0(connect14[4]), .E0(connect14[2]), .F0(connect14[1]),
        .A1(disp1[6]), .B1(disp1[5]), .C1(disp1[4]), .D1(disp1[3]), .E1(
        disp1[2]), .F1(disp1[1]), .G1(disp1[0]) );
  CONVERTOR_1 U8 ( .T0(connect13[5]), .T1(connect13[4]), .T2(connect13[3]),
        .T3(connect13[2]), .T4(connect13[1]), .T5(connect13[0]), .A0(disp2[13]), .B0(disp2[12]), .C0(disp2[11]),
.D0(disp2[10]), .E0(disp2[9]), .F0(disp2[8]),
        .G0(disp2[7]), .A1(disp2[6]), .B1(disp2[5]), .C1(disp2[4]), .D1(
        disp2[3]), .E1(disp2[2]), .F1(disp2[1]), .G1(disp2[0]) );
  HOURS_FILTER U9 ( .TENS_DIGIT_HOURS_IN({connect14[6:4], net2689,
        connect14[2:1], net2690}), .TENS_DIGIT_HOURS_OUT({disp1[13:8],
        SYNOPSYS_UNCONNECTED__0}) );
endmodule


module COMPARATOR ( ALARM_HRS, CLOCK_HRS, ALARM_MINS, CLOCK_MINS, ALARM_AM_PM,
        CLOCK_AM_PM, RINGER );
  input [3:0] ALARM_HRS;
  input [3:0] CLOCK_HRS;
  input [5:0] ALARM_MINS;
  input [5:0] CLOCK_MINS;
  input ALARM_AM_PM, CLOCK_AM_PM;
  output RINGER;
  wire   N2, N4, n2, n3, n4, n5, n6, n7, n1, n8, n9, n10, n11, n12, n13, n14,
         n15, n16, n17, n18;
  assign RINGER = N4;

  EOI U3 ( .A(CLOCK_HRS[2]), .B(ALARM_HRS[2]), .Z(n5) );
  EOI U4 ( .A(CLOCK_AM_PM), .B(ALARM_AM_PM), .Z(n4) );
  EOI U5 ( .A(CLOCK_HRS[1]), .B(ALARM_HRS[1]), .Z(n3) );
  ENI U7 ( .A(CLOCK_HRS[0]), .B(ALARM_HRS[0]), .Z(n7) );
```

```
  ENI U8 ( .A(CLOCK_HRS[3]), .B(ALARM_HRS[3]), .Z(n6) );
  ND3 U1 ( .A(n6), .B(N2), .C(n7), .Z(n2) );
  NR4 U2 ( .A(n2), .B(n3), .C(n4), .D(n5), .Z(N4) );
  ND2 U6 ( .A(ALARM_MINS[0]), .B(n16), .Z(n1) );
  AO2 U9 ( .A(n17), .B(n1), .C(n1), .D(CLOCK_MINS[1]), .Z(n8) );
  NR2 U10 ( .A(n16), .B(ALARM_MINS[0]), .Z(n9) );
  AO4 U11 ( .A(n9), .B(n17), .C(CLOCK_MINS[1]), .D(n9), .Z(n11) );
  EN U12 ( .A(CLOCK_MINS[5]), .B(ALARM_MINS[5]), .Z(n10) );
  ND3 U13 ( .A(n18), .B(n11), .C(n10), .Z(n15) );
  EO U14 ( .A(CLOCK_MINS[4]), .B(ALARM_MINS[4]), .Z(n14) );
  EO U15 ( .A(CLOCK_MINS[2]), .B(ALARM_MINS[2]), .Z(n13) );
  EO U16 ( .A(CLOCK_MINS[3]), .B(ALARM_MINS[3]), .Z(n12) );
  NR4 U17 ( .A(n15), .B(n14), .C(n13), .D(n12), .Z(N2) );
  IVI U18 ( .A(CLOCK_MINS[0]), .Z(n16) );
  IVI U19 ( .A(ALARM_MINS[1]), .Z(n17) );
  IVI U20 ( .A(n8), .Z(n18) );
endmodule


module ALARM_SM_2 ( COMPARE_IN, TOGGLE_ON, CLOCK, RING );
  input COMPARE_IN, TOGGLE_ON, CLOCK;
  output RING;
  wire   CURRENT_STATE, n2, n1;
  assign RING = CURRENT_STATE;

  FD1 CURRENT_STATE_reg ( .D(n1), .CP(CLOCK), .Q(CURRENT_STATE) );
  AO7 U3 ( .A(COMPARE_IN), .B(CURRENT_STATE), .C(TOGGLE_ON), .Z(n2) );
  IVI U4 ( .A(n2), .Z(n1) );
endmodule


module MUX ( ALARM_HRS, ALARM_MINS, ALARM_AM_PM, TIME_HRS, TIME_MINS,
        TIME_AM_PM, ALARM_SET, OUTBUS );
  input [3:0] ALARM_HRS;
  input [5:0] ALARM_MINS;
  input [3:0] TIME_HRS;
  input [5:0] TIME_MINS;
  output [10:0] OUTBUS;
  input ALARM_AM_PM, TIME_AM_PM, ALARM_SET;
  wire   n13, n14, n15, n16, n17, n18, n19, n20, n21, n22, n23, n12;

  AO2P U10 ( .A(ALARM_HRS[3]), .B(ALARM_SET), .C(TIME_HRS[3]), .D(n12), .Z(n22) );
  AO2 U1 ( .A(ALARM_MINS[5]), .B(ALARM_SET), .C(TIME_MINS[5]), .D(n12), .Z(n16) );
```

```
  AO2 U2 ( .A(ALARM_MINS[3]), .B(ALARM_SET), .C(TIME_MINS[3]), .D(n12), .Z(n18) );
  AO2 U3 ( .A(ALARM_MINS[4]), .B(ALARM_SET), .C(TIME_MINS[4]), .D(n12), .Z(n17) );
  AO2 U4 ( .A(ALARM_SET), .B(ALARM_HRS[2]), .C(TIME_HRS[2]), .D(n12), .Z(n13)
        );
  AO2 U5 ( .A(ALARM_HRS[0]), .B(ALARM_SET), .C(TIME_HRS[0]), .D(n12), .Z(n15)
        );
  IVI U6 ( .A(n22), .Z(OUTBUS[10]) );
  AO2 U7 ( .A(ALARM_HRS[1]), .B(ALARM_SET), .C(TIME_HRS[1]), .D(n12), .Z(n14)
        );
  AO2 U8 ( .A(ALARM_MINS[2]), .B(ALARM_SET), .C(TIME_MINS[2]), .D(n12), .Z(n19) );
  AO2 U9 ( .A(ALARM_MINS[0]), .B(ALARM_SET), .C(TIME_MINS[0]), .D(n12), .Z(n21) );
  AO2 U11 ( .A(ALARM_AM_PM), .B(ALARM_SET), .C(TIME_AM_PM), .D(n12), .Z(n23)
        );
  IVI U12 ( .A(ALARM_SET), .Z(n12) );
  AO2 U13 ( .A(ALARM_MINS[1]), .B(ALARM_SET), .C(TIME_MINS[1]), .D(n12), .Z(
       n20) );
  IVI U14 ( .A(n21), .Z(OUTBUS[1]) );
  IVI U15 ( .A(n13), .Z(OUTBUS[9]) );
  IVI U16 ( .A(n14), .Z(OUTBUS[8]) );
  IVI U17 ( .A(n15), .Z(OUTBUS[7]) );
  IVI U18 ( .A(n20), .Z(OUTBUS[2]) );
  IVI U19 ( .A(n19), .Z(OUTBUS[3]) );
  IVI U20 ( .A(n18), .Z(OUTBUS[4]) );
  IVI U21 ( .A(n17), .Z(OUTBUS[5]) );
  IVI U22 ( .A(n16), .Z(OUTBUS[6]) );
  IVI U23 ( .A(n23), .Z(OUTBUS[0]) );
endmodule


module TOP ( SET_TIME, ALARM, HRS, MINS, TOGGLE_SWITCH, CLK, SPEAKER_OUT,
        DISP1, DISP2, AM_PM_DISPLAY );
  output [13:0] DISP1;
  output [13:0] DISP2;
  input SET_TIME, ALARM, HRS, MINS, TOGGLE_SWITCH, CLK;
  output SPEAKER_OUT, AM_PM_DISPLAY;
  wire   KONNECT8, KONNECT11, KONNECT12;
  wire   [3:0] KONNECT6;
  wire   [5:0] KONNECT7;
  wire   [3:0] KONNECT9;
  wire   [5:0] KONNECT10;
  wire   [9:0] KONNECT13;
  wire   SYNOPSYS_UNCONNECTED__0;
  assign DISP1[7] = 1'b0;
```

```
  TIME_BLOCK U1 ( .SET_TIME(SET_TIME), .HRS(HRS), .MINS(MINS), .CLK(CLK),
        .CONNECT6(KONNECT6), .CONNECT7(KONNECT7), .CONNECT8(KONNECT8) );
  ALARM_BLOCK U2 ( .ALARM(ALARM), .HRS(HRS), .MINS(MINS), .CLK(CLK),
        .CONNECT9(KONNECT9), .CONNECT10(KONNECT10), .CONNECT11(KONNECT11) );
  CONVERTOR_CKT U3 ( .connect13(KONNECT13), .disp1({DISP1[13:8],
        SYNOPSYS_UNCONNECTED__0, DISP1[6:0]}), .disp2(DISP2) );
  COMPARATOR U4 ( .ALARM_HRS(KONNECT9), .CLOCK_HRS(KONNECT6), .ALARM_MINS(
        KONNECT10), .CLOCK_MINS(KONNECT7), .ALARM_AM_PM(KONNECT11),
        .CLOCK_AM_PM(KONNECT8), .RINGER(KONNECT12) );
  ALARM_SM_2 U5 ( .COMPARE_IN(KONNECT12), .TOGGLE_ON(TOGGLE_SWITCH), .CLOCK(
        CLK), .RING(SPEAKER_OUT) );
  MUX U6 ( .ALARM_HRS(KONNECT9), .ALARM_MINS(KONNECT10), .ALARM_AM_PM(
        KONNECT11), .TIME_HRS(KONNECT6), .TIME_MINS(KONNECT7), .TIME_AM_PM(
        KONNECT8), .ALARM_SET(ALARM), .OUTBUS({KONNECT13, AM_PM_DISPLAY}) );
endmodule
```

Cruise Control

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version   : O-2018.06-SP3
// Date      : Mon Nov 11 12:35:05 2019
////////////////////////////////////////////////////////////


module cruise_control_DW01_inc_0 ( A, SUM );
  input [7:0] A;
  output [7:0] SUM;

  wire   [7:2] carry;

  HAX1 U1_1_6 ( .A(A[6]), .B(carry[6]), .YC(carry[7]), .YS(SUM[6]) );
  HAX1 U1_1_5 ( .A(A[5]), .B(carry[5]), .YC(carry[6]), .YS(SUM[5]) );
  HAX1 U1_1_4 ( .A(A[4]), .B(carry[4]), .YC(carry[5]), .YS(SUM[4]) );
  HAX1 U1_1_3 ( .A(A[3]), .B(carry[3]), .YC(carry[4]), .YS(SUM[3]) );
  HAX1 U1_1_2 ( .A(A[2]), .B(carry[2]), .YC(carry[3]), .YS(SUM[2]) );
  HAX1 U1_1_1 ( .A(A[1]), .B(A[0]), .YC(carry[2]), .YS(SUM[1]) );
  INVX1 U1 ( .A(A[0]), .Y(SUM[0]) );
  XOR2X1 U2 ( .A(carry[7]), .B(A[7]), .Y(SUM[7]) );
endmodule
```

```
module cruise_control_DW01_inc_1 ( A, SUM );
  input [7:0] A;
  output [7:0] SUM;

  wire   [7:2] carry;

  HAX1 U1_1_6 ( .A(A[6]), .B(carry[6]), .YC(carry[7]), .YS(SUM[6]) );
  HAX1 U1_1_5 ( .A(A[5]), .B(carry[5]), .YC(carry[6]), .YS(SUM[5]) );
  HAX1 U1_1_4 ( .A(A[4]), .B(carry[4]), .YC(carry[5]), .YS(SUM[4]) );
  HAX1 U1_1_3 ( .A(A[3]), .B(carry[3]), .YC(carry[4]), .YS(SUM[3]) );
  HAX1 U1_1_2 ( .A(A[2]), .B(carry[2]), .YC(carry[3]), .YS(SUM[2]) );
  HAX1 U1_1_1 ( .A(A[1]), .B(A[0]), .YC(carry[2]), .YS(SUM[1]) );
  INVX2 U1 ( .A(A[0]), .Y(SUM[0]) );
  XOR2X1 U2 ( .A(carry[7]), .B(A[7]), .Y(SUM[7]) );
endmodule


module cruise_control ( clk, reset, throttle, set_0011, accel, coast, cancel,
        resume, brake, speed, cruisespeed, cruisecontrol );
  output [7:0] speed;
  output [7:0] cruisespeed;
  input clk, reset, throttle, set_0011, accel, coast, cancel, resume, brake;
  output cruisecontrol;
  wire   n419, n420, n421, n422, n423, n424, n425, n426, n427, n428, n429,
         n430, n431, n432, n433, n434, N17, N18, N19, N45, N47, N48, N49, N50,
         N51, N52, N63, N64, N65, N66, N67, N68, N69, N79, N80, N93, N94, N95,
         N96, N97, N98, N99, N100, N128, N129, N130, N131, N132, N133, N134,
         N135, N146, N149, N150, N151, N152, N153, N154, N155, N183, N185,
         N186, N187, N188, N189, N190, N207, N208, N209, N210, N211, N212,
         N217, N218, N219, N220, N221, N222, N223, N224, N225, N226, N227,
         N228, N229, N230, N231, N232, N233, N234, N235, N236, N237, N238,
         N239, N240, \gt_69/B[0] , \gt_69/B[1] , \gt_69/B[2] , \gt_69/B[3] ,
         \gt_69/B[4] , \gt_69/B[5] , \gt_69/B[6] , \gt_69/B[7] , \sub_78/A[1] ,
         \sub_78/A[2] , \sub_78/A[3] , \sub_78/A[4] , \sub_78/A[5] ,
         \sub_78/A[6] , \sub_78/A[7] , n180, n184, n187, n189, n191, n195,
         n197, n199, n201, n203, n205, n207, n209, n210, n211, n212, n213,
         n214, n215, n216, n217, n218, n219, n220, n221, n222, n223, n224,
         n225, n226, n227, n228, n229, n230, n231, n232, n233, n234, n235,
         n236, n237, n238, n239, n240, n241, n242, n243, n244, n245, n246,
         n247, n248, n249, n250, n251, n252, n253, n254, n255, n256, n257,
         n258, n259, n260, n261, n262, n263, n264, n265, n266, n267, n268,
         n269, n270, n271, n272, n273, n274, n275, n276, n277, n278, n279,
         n280, n281, n282, n283, n284, n285, n286, n287, n288, n289, n290,
```

```
       n291, n292, n293, n294, n295, n296, n297, n298, n299, n300, n301,
       n302, n303, n304, n305, n306, n307, n308, n309, n310, n311, n312,
       n313, n314, n315, n316, n317, n318, n319, n320, n321, n322, n323,
       n324, n325, n326, n327, n328, n329, n330, n331, n332, n333, n334,
       n335, n336, n337, n338, n339, n340, n341, n342, n343, n344, n345,
       n346, n347, n348, n349, n350, n351, n352, n353, n354, n355, n356,
       n357, n358, n359, n360, n361, n362, n363, n364, n365, n366, n367,
       n368, n369, n370, n371, n372, n373, n374, n375, n376, n377, n378,
       n379, n380, n381, n382, n383, n384, n385, n386, n387, n388, n389,
       n390, n391, n392, n393, n394, n395, n396, n397, n398, n399, n400,
       n401, n402, n403, n404, n405, n406, n407, n408, n409, n410, n411,
       n412, n413, n414, n415, n416, n417, n418;
wire   [2:0] state;
wire   [2:0] nextstate;
wire   [8:0] \sub_85/carry ;
wire   [8:0] \sub_78/carry ;
wire   [7:1] \r108/carry ;

LATCH \cruisespeed_reg[0]  ( .CLK(N230), .D(N231), .Q(n433) );
LATCH \cruisespeed_reg[7]  ( .CLK(N230), .D(N238), .Q(n426) );
LATCH \cruisespeed_reg[6]  ( .CLK(N230), .D(N237), .Q(n427) );
LATCH \cruisespeed_reg[5]  ( .CLK(N230), .D(N236), .Q(n428) );
LATCH \speed_reg[7]  ( .CLK(N221), .D(N229), .Q(n419) );
LATCH \nextstate_reg[2]  ( .CLK(N217), .D(N220), .Q(nextstate[2]) );
DFFPOSX1 \state_reg[2]  ( .D(N19), .CLK(clk), .Q(state[2]) );
LATCH \speed_reg[6]  ( .CLK(N221), .D(N228), .Q(n420) );
LATCH \speed_reg[5]  ( .CLK(N221), .D(N227), .Q(n421) );
LATCH \speed_reg[4]  ( .CLK(N221), .D(N226), .Q(n422) );
LATCH \cruisespeed_reg[4]  ( .CLK(N230), .D(N235), .Q(n429) );
LATCH \speed_reg[3]  ( .CLK(N221), .D(N225), .Q(n423) );
LATCH \speed_reg[2]  ( .CLK(N221), .D(N224), .Q(n424) );
LATCH \speed_reg[1]  ( .CLK(N221), .D(N223), .Q(n425) );
LATCH \nextstate_reg[1]  ( .CLK(N217), .D(N219), .Q(nextstate[1]) );
DFFPOSX1 \state_reg[1]  ( .D(N18), .CLK(clk), .Q(state[1]) );
LATCH \nextstate_reg[0]  ( .CLK(N217), .D(N218), .Q(nextstate[0]) );
DFFPOSX1 \state_reg[0]  ( .D(N17), .CLK(clk), .Q(state[0]) );
LATCH \speed_reg[0]  ( .CLK(N221), .D(N222), .Q(N45) );
LATCH cruisecontrol_reg ( .CLK(N239), .D(N240), .Q(n434) );
LATCH \cruisespeed_reg[1]  ( .CLK(N230), .D(N232), .Q(n432) );
LATCH \cruisespeed_reg[2]  ( .CLK(N230), .D(N233), .Q(n431) );
LATCH \cruisespeed_reg[3]  ( .CLK(N230), .D(N234), .Q(n430) );
cruise_control_DW01_inc_0 add_66 ( .A({cruisespeed[7], n427, n428,
      cruisespeed[4], n430, cruisespeed[2:1], n433}), .SUM({N135, N134, N133,
```

```
      N132, N131, N130, N129, N128}) );
cruise_control_DW01_inc_1 add_61 ( .A({n419, n420, n421, n422, n423, n424,
      n425, N45}), .SUM({N100, N99, N98, N97, N96, N95, N94, N93}) );
INVX1 U199 ( .A(n434), .Y(n180) );
INVX8 U200 ( .A(n180), .Y(cruisecontrol) );
INVX8 U201 ( .A(n259), .Y(cruisespeed[6]) );
INVX1 U202 ( .A(n427), .Y(n259) );
INVX8 U203 ( .A(n255), .Y(cruisespeed[3]) );
INVX1 U204 ( .A(n430), .Y(n255) );
INVX2 U205 ( .A(n432), .Y(n184) );
INVX8 U206 ( .A(n184), .Y(cruisespeed[1]) );
INVX8 U207 ( .A(n258), .Y(cruisespeed[5]) );
INVX1 U208 ( .A(n428), .Y(n258) );
INVX2 U209 ( .A(n431), .Y(n187) );
INVX8 U210 ( .A(n187), .Y(cruisespeed[2]) );
OR2X2 U211 ( .A(n197), .B(cruisespeed[2]), .Y(n227) );
INVX2 U212 ( .A(n429), .Y(n189) );
INVX8 U213 ( .A(n189), .Y(cruisespeed[4]) );
OR2X2 U214 ( .A(n201), .B(cruisespeed[4]), .Y(n230) );
INVX2 U215 ( .A(n426), .Y(n191) );
INVX8 U216 ( .A(n191), .Y(cruisespeed[7]) );
INVX8 U217 ( .A(n263), .Y(cruisespeed[0]) );
INVX1 U218 ( .A(n433), .Y(n263) );
INVX2 U219 ( .A(n297), .Y(n307) );
INVX2 U220 ( .A(n380), .Y(n333) );
INVX2 U221 ( .A(n381), .Y(n332) );
INVX2 U222 ( .A(n385), .Y(n340) );
INVX2 U223 ( .A(n421), .Y(n203) );
INVX2 U224 ( .A(n423), .Y(n199) );
INVX2 U225 ( .A(n424), .Y(n197) );
INVX2 U226 ( .A(n425), .Y(n195) );
INVX2 U227 ( .A(n422), .Y(n201) );
INVX2 U228 ( .A(n420), .Y(n205) );
INVX2 U229 ( .A(n419), .Y(n207) );
INVX2 U230 ( .A(throttle), .Y(n338) );
INVX8 U231 ( .A(n374), .Y(speed[0]) );
INVX8 U232 ( .A(n195), .Y(speed[1]) );
INVX8 U233 ( .A(n197), .Y(speed[2]) );
INVX8 U234 ( .A(n199), .Y(speed[3]) );
INVX8 U235 ( .A(n201), .Y(speed[4]) );
INVX8 U236 ( .A(n203), .Y(speed[5]) );
INVX8 U237 ( .A(n205), .Y(speed[6]) );
INVX8 U238 ( .A(n207), .Y(speed[7]) );
```

```
XNOR2X1 U239 ( .A(n419), .B(\sub_85/carry [7]), .Y(N212) );
OR2X1 U240 ( .A(\sub_85/carry [6]), .B(n420), .Y(\sub_85/carry [7]) );
XNOR2X1 U241 ( .A(n420), .B(\sub_85/carry [6]), .Y(N211) );
OR2X1 U242 ( .A(\sub_85/carry [5]), .B(n421), .Y(\sub_85/carry [6]) );
XNOR2X1 U243 ( .A(n421), .B(\sub_85/carry [5]), .Y(N210) );
OR2X1 U244 ( .A(\sub_85/carry [4]), .B(n422), .Y(\sub_85/carry [5]) );
XNOR2X1 U245 ( .A(n422), .B(\sub_85/carry [4]), .Y(N209) );
OR2X1 U246 ( .A(\sub_85/carry [3]), .B(n423), .Y(\sub_85/carry [4]) );
XNOR2X1 U247 ( .A(n423), .B(\sub_85/carry [3]), .Y(N208) );
OR2X1 U248 ( .A(n425), .B(n424), .Y(\sub_85/carry [3]) );
XNOR2X1 U249 ( .A(n424), .B(n425), .Y(N207) );
XNOR2X1 U250 ( .A(\sub_78/A[7] ), .B(\sub_78/carry [7]), .Y(N190) );
OR2X1 U251 ( .A(\sub_78/carry [6]), .B(\sub_78/A[6] ), .Y(\sub_78/carry [7])
       );
XNOR2X1 U252 ( .A(\sub_78/A[6] ), .B(\sub_78/carry [6]), .Y(N189) );
OR2X1 U253 ( .A(\sub_78/carry [5]), .B(\sub_78/A[5] ), .Y(\sub_78/carry [6])
       );
XNOR2X1 U254 ( .A(\sub_78/A[5] ), .B(\sub_78/carry [5]), .Y(N188) );
OR2X1 U255 ( .A(\sub_78/carry [4]), .B(\sub_78/A[4] ), .Y(\sub_78/carry [5])
       );
XNOR2X1 U256 ( .A(\sub_78/A[4] ), .B(\sub_78/carry [4]), .Y(N187) );
OR2X1 U257 ( .A(\sub_78/carry [3]), .B(\sub_78/A[3] ), .Y(\sub_78/carry [4])
       );
XNOR2X1 U258 ( .A(\sub_78/A[3] ), .B(\sub_78/carry [3]), .Y(N186) );
OR2X1 U259 ( .A(\sub_78/A[1] ), .B(\sub_78/A[2] ), .Y(\sub_78/carry [3]) );
XNOR2X1 U260 ( .A(\sub_78/A[2] ), .B(\sub_78/A[1] ), .Y(N185) );
XOR2X1 U261 ( .A(n419), .B(\r108/carry [7]), .Y(N52) );
AND2X1 U262 ( .A(n420), .B(\r108/carry [6]), .Y(\r108/carry [7]) );
XOR2X1 U263 ( .A(\r108/carry [6]), .B(n420), .Y(N51) );
AND2X1 U264 ( .A(n421), .B(\r108/carry [5]), .Y(\r108/carry [6]) );
XOR2X1 U265 ( .A(\r108/carry [5]), .B(n421), .Y(N50) );
AND2X1 U266 ( .A(n422), .B(\r108/carry [4]), .Y(\r108/carry [5]) );
XOR2X1 U267 ( .A(\r108/carry [4]), .B(n422), .Y(N49) );
AND2X1 U268 ( .A(n423), .B(\r108/carry [3]), .Y(\r108/carry [4]) );
XOR2X1 U269 ( .A(\r108/carry [3]), .B(n423), .Y(N48) );
AND2X1 U270 ( .A(n424), .B(n425), .Y(\r108/carry [3]) );
XOR2X1 U271 ( .A(n425), .B(n424), .Y(N47) );
NAND2X1 U272 ( .A(n397), .B(n374), .Y(n209) );
OAI21X1 U273 ( .A(n374), .B(n397), .C(n209), .Y(N63) );
NOR2X1 U274 ( .A(n209), .B(n424), .Y(n211) );
AOI21X1 U275 ( .A(n209), .B(n424), .C(n211), .Y(n210) );
NAND2X1 U276 ( .A(n211), .B(n199), .Y(n212) );
OAI21X1 U277 ( .A(n211), .B(n199), .C(n212), .Y(N65) );
```

```
NOR2X1 U278 ( .A(n212), .B(n422), .Y(n214) );
AOI21X1 U279 ( .A(n212), .B(n422), .C(n214), .Y(n213) );
NAND2X1 U280 ( .A(n214), .B(n203), .Y(n215) );
OAI21X1 U281 ( .A(n214), .B(n203), .C(n215), .Y(N67) );
XNOR2X1 U282 ( .A(n420), .B(n215), .Y(N68) );
NOR2X1 U283 ( .A(n420), .B(n215), .Y(n216) );
XOR2X1 U284 ( .A(n419), .B(n216), .Y(N69) );
INVX2 U285 ( .A(n210), .Y(N64) );
INVX2 U286 ( .A(n213), .Y(N66) );
NAND2X1 U287 ( .A(n295), .B(n296), .Y(n217) );
OAI21X1 U288 ( .A(n296), .B(n295), .C(n217), .Y(N149) );
NOR2X1 U289 ( .A(n217), .B(\gt_69/B[2] ), .Y(n219) );
AOI21X1 U290 ( .A(n217), .B(\gt_69/B[2] ), .C(n219), .Y(n218) );
NAND2X1 U291 ( .A(n219), .B(n293), .Y(n220) );
OAI21X1 U292 ( .A(n219), .B(n293), .C(n220), .Y(N151) );
NOR2X1 U293 ( .A(n220), .B(\gt_69/B[4] ), .Y(n222) );
AOI21X1 U294 ( .A(n220), .B(\gt_69/B[4] ), .C(n222), .Y(n221) );
NAND2X1 U295 ( .A(n222), .B(n291), .Y(n223) );
OAI21X1 U296 ( .A(n222), .B(n291), .C(n223), .Y(N153) );
XNOR2X1 U297 ( .A(\gt_69/B[6] ), .B(n223), .Y(N154) );
NOR2X1 U298 ( .A(\gt_69/B[6] ), .B(n223), .Y(n224) );
XOR2X1 U299 ( .A(\gt_69/B[7] ), .B(n224), .Y(N155) );
INVX2 U300 ( .A(n218), .Y(N150) );
INVX2 U301 ( .A(n221), .Y(N152) );
NAND2X1 U302 ( .A(cruisespeed[7]), .B(n207), .Y(n253) );
NAND2X1 U303 ( .A(n421), .B(n258), .Y(n248) );
AND2X1 U304 ( .A(n230), .B(n248), .Y(n233) );
NAND2X1 U305 ( .A(cruisespeed[2]), .B(n197), .Y(n239) );
NAND2X1 U306 ( .A(n227), .B(n239), .Y(n241) );
NAND2X1 U307 ( .A(N45), .B(n263), .Y(n225) );
OAI21X1 U308 ( .A(n195), .B(n225), .C(cruisespeed[1]), .Y(n226) );
OAI21X1 U309 ( .A(n425), .B(n266), .C(n226), .Y(n229) );
NAND2X1 U310 ( .A(n423), .B(n255), .Y(n242) );
AND2X1 U311 ( .A(n227), .B(n242), .Y(n228) );
OAI21X1 U312 ( .A(n241), .B(n229), .C(n228), .Y(n231) );
NOR2X1 U313 ( .A(n255), .B(n423), .Y(n244) );
NAND2X1 U314 ( .A(cruisespeed[4]), .B(n201), .Y(n245) );
NAND2X1 U315 ( .A(n230), .B(n245), .Y(n247) );
NAND3X1 U316 ( .A(n231), .B(n256), .C(n257), .Y(n232) );
NOR2X1 U317 ( .A(n258), .B(n421), .Y(n250) );
AOI21X1 U318 ( .A(n233), .B(n232), .C(n250), .Y(n234) );
XOR2X1 U319 ( .A(n420), .B(n259), .Y(n236) );
AOI22X1 U320 ( .A(n420), .B(n259), .C(n234), .D(n236), .Y(n235) );
```

```
NOR2X1 U321 ( .A(n207), .B(cruisespeed[7]), .Y(n254) );
OAI21X1 U322 ( .A(n261), .B(n235), .C(n262), .Y(N80) );
NOR2X1 U323 ( .A(n263), .B(N45), .Y(n238) );
AOI21X1 U324 ( .A(n397), .B(n238), .C(cruisespeed[1]), .Y(n237) );
OAI21X1 U325 ( .A(n238), .B(n195), .C(n264), .Y(n240) );
OAI21X1 U326 ( .A(n241), .B(n240), .C(n239), .Y(n243) );
OAI21X1 U327 ( .A(n244), .B(n243), .C(n242), .Y(n246) );
OAI21X1 U328 ( .A(n247), .B(n246), .C(n245), .Y(n249) );
OAI21X1 U329 ( .A(n250), .B(n249), .C(n248), .Y(n251) );
OAI22X1 U330 ( .A(n260), .B(n251), .C(n420), .D(n259), .Y(n252) );
OAI21X1 U331 ( .A(n254), .B(n265), .C(n253), .Y(N79) );
INVX2 U332 ( .A(n244), .Y(n256) );
INVX2 U333 ( .A(n247), .Y(n257) );
INVX2 U334 ( .A(n236), .Y(n260) );
INVX2 U335 ( .A(n253), .Y(n261) );
INVX2 U336 ( .A(n254), .Y(n262) );
INVX2 U337 ( .A(n237), .Y(n264) );
INVX2 U338 ( .A(n252), .Y(n265) );
INVX2 U339 ( .A(n225), .Y(n266) );
NOR2X1 U340 ( .A(\gt_69/B[7] ), .B(\gt_69/B[6] ), .Y(n270) );
OR2X1 U341 ( .A(\gt_69/B[2] ), .B(\gt_69/B[1] ), .Y(n267) );
OAI21X1 U342 ( .A(\gt_69/B[0] ), .B(n267), .C(\gt_69/B[3] ), .Y(n268) );
OAI21X1 U343 ( .A(n271), .B(\gt_69/B[4] ), .C(\gt_69/B[5] ), .Y(n269) );
NAND2X1 U344 ( .A(n270), .B(n269), .Y(N146) );
INVX2 U345 ( .A(n268), .Y(n271) );
INVX1 U346 ( .A(n272), .Y(\sub_78/A[7] ) );
MUX2X1 U347 ( .B(n419), .A(n273), .S(n274), .Y(n272) );
INVX1 U348 ( .A(n275), .Y(\sub_78/A[6] ) );
MUX2X1 U349 ( .B(n420), .A(n276), .S(n274), .Y(n275) );
INVX1 U350 ( .A(n277), .Y(\sub_78/A[5] ) );
MUX2X1 U351 ( .B(n421), .A(n278), .S(n274), .Y(n277) );
INVX1 U352 ( .A(n279), .Y(\sub_78/A[4] ) );
MUX2X1 U353 ( .B(n422), .A(n280), .S(n274), .Y(n279) );
INVX1 U354 ( .A(n281), .Y(\sub_78/A[3] ) );
MUX2X1 U355 ( .B(n423), .A(n282), .S(n274), .Y(n281) );
INVX1 U356 ( .A(n283), .Y(\sub_78/A[2] ) );
MUX2X1 U357 ( .B(n424), .A(n284), .S(n274), .Y(n283) );
INVX1 U358 ( .A(n285), .Y(\sub_78/A[1] ) );
MUX2X1 U359 ( .B(n425), .A(n286), .S(n274), .Y(n285) );
INVX1 U360 ( .A(n287), .Y(N183) );
MUX2X1 U361 ( .B(N45), .A(n288), .S(n274), .Y(n287) );
INVX1 U362 ( .A(n289), .Y(\gt_69/B[7] ) );
MUX2X1 U363 ( .B(cruisespeed[7]), .A(N135), .S(accel), .Y(n289) );
```

```
INVX1 U364 ( .A(n290), .Y(\gt_69/B[6] ) );
MUX2X1 U365 ( .B(n427), .A(N134), .S(accel), .Y(n290) );
INVX1 U366 ( .A(n291), .Y(\gt_69/B[5] ) );
MUX2X1 U367 ( .B(n428), .A(N133), .S(accel), .Y(n291) );
INVX1 U368 ( .A(n292), .Y(\gt_69/B[4] ) );
MUX2X1 U369 ( .B(cruisespeed[4]), .A(N132), .S(accel), .Y(n292) );
INVX1 U370 ( .A(n293), .Y(\gt_69/B[3] ) );
MUX2X1 U371 ( .B(n430), .A(N131), .S(accel), .Y(n293) );
INVX1 U372 ( .A(n294), .Y(\gt_69/B[2] ) );
MUX2X1 U373 ( .B(cruisespeed[2]), .A(N130), .S(accel), .Y(n294) );
INVX1 U374 ( .A(n295), .Y(\gt_69/B[1] ) );
MUX2X1 U375 ( .B(cruisespeed[1]), .A(N129), .S(accel), .Y(n295) );
INVX1 U376 ( .A(n296), .Y(\gt_69/B[0] ) );
MUX2X1 U377 ( .B(n433), .A(N128), .S(accel), .Y(n296) );
NAND2X1 U378 ( .A(n297), .B(n298), .Y(N240) );
NAND3X1 U379 ( .A(n299), .B(n300), .C(n301), .Y(N239) );
NAND2X1 U380 ( .A(n302), .B(n303), .Y(N238) );
AOI22X1 U381 ( .A(n304), .B(cruisespeed[7]), .C(n305), .D(N135), .Y(n303) );
AOI22X1 U382 ( .A(N155), .B(n306), .C(n419), .D(n307), .Y(n302) );
NAND2X1 U383 ( .A(n308), .B(n309), .Y(N237) );
AOI22X1 U384 ( .A(n304), .B(n427), .C(n305), .D(N134), .Y(n309) );
AOI22X1 U385 ( .A(N154), .B(n306), .C(n420), .D(n307), .Y(n308) );
NAND2X1 U386 ( .A(n310), .B(n311), .Y(N236) );
AOI22X1 U387 ( .A(n304), .B(n428), .C(n305), .D(N133), .Y(n311) );
AOI22X1 U388 ( .A(N153), .B(n306), .C(n421), .D(n307), .Y(n310) );
NAND2X1 U389 ( .A(n312), .B(n313), .Y(N235) );
AOI22X1 U390 ( .A(n304), .B(cruisespeed[4]), .C(n305), .D(N132), .Y(n313) );
AOI22X1 U391 ( .A(N152), .B(n306), .C(n422), .D(n307), .Y(n312) );
NAND2X1 U392 ( .A(n314), .B(n315), .Y(N234) );
AOI22X1 U393 ( .A(n304), .B(n430), .C(n305), .D(N131), .Y(n315) );
AOI22X1 U394 ( .A(N151), .B(n306), .C(n423), .D(n307), .Y(n314) );
NAND2X1 U395 ( .A(n316), .B(n317), .Y(N233) );
AOI22X1 U396 ( .A(n304), .B(cruisespeed[2]), .C(n305), .D(N130), .Y(n317) );
AOI22X1 U397 ( .A(N150), .B(n306), .C(n424), .D(n307), .Y(n316) );
NAND2X1 U398 ( .A(n318), .B(n319), .Y(N232) );
AOI22X1 U399 ( .A(n304), .B(cruisespeed[1]), .C(n305), .D(N129), .Y(n319) );
AOI22X1 U400 ( .A(N149), .B(n306), .C(n425), .D(n307), .Y(n318) );
NAND2X1 U401 ( .A(n320), .B(n321), .Y(N231) );
AOI22X1 U402 ( .A(n304), .B(n433), .C(n305), .D(N128), .Y(n321) );
AOI21X1 U403 ( .A(n322), .B(coast), .C(n323), .Y(n305) );
NAND2X1 U404 ( .A(n324), .B(accel), .Y(n322) );
NOR3X1 U405 ( .A(N146), .B(accel), .C(n325), .Y(n304) );
AOI22X1 U406 ( .A(n296), .B(n306), .C(N45), .D(n307), .Y(n320) );
```

```
NOR2X1 U407 ( .A(n324), .B(n325), .Y(n306) );
INVX1 U408 ( .A(N146), .Y(n324) );
NAND3X1 U409 ( .A(n299), .B(n325), .C(n326), .Y(N230) );
AOI21X1 U410 ( .A(n327), .B(accel), .C(n328), .Y(n326) );
NAND2X1 U411 ( .A(coast), .B(n327), .Y(n325) );
NAND3X1 U412 ( .A(n329), .B(n330), .C(n331), .Y(N229) );
AOI22X1 U413 ( .A(n332), .B(N69), .C(n333), .D(N52), .Y(n331) );
NAND2X1 U414 ( .A(N212), .B(n334), .Y(n330) );
AOI22X1 U415 ( .A(n335), .B(n273), .C(N190), .D(n336), .Y(n329) );
OAI21X1 U416 ( .A(n337), .B(n338), .C(n339), .Y(n273) );
AOI22X1 U417 ( .A(N69), .B(n340), .C(N100), .D(n341), .Y(n339) );
INVX1 U418 ( .A(N52), .Y(n337) );
NAND3X1 U419 ( .A(n342), .B(n343), .C(n344), .Y(N228) );
AOI22X1 U420 ( .A(n332), .B(N68), .C(n333), .D(N51), .Y(n344) );
NAND2X1 U421 ( .A(N211), .B(n334), .Y(n343) );
AOI22X1 U422 ( .A(n335), .B(n276), .C(N189), .D(n336), .Y(n342) );
OAI21X1 U423 ( .A(n338), .B(n345), .C(n346), .Y(n276) );
AOI22X1 U424 ( .A(N68), .B(n340), .C(N99), .D(n341), .Y(n346) );
INVX1 U425 ( .A(N51), .Y(n345) );
NAND3X1 U426 ( .A(n347), .B(n348), .C(n349), .Y(N227) );
AOI22X1 U427 ( .A(n332), .B(N67), .C(n333), .D(N50), .Y(n349) );
NAND2X1 U428 ( .A(N210), .B(n334), .Y(n348) );
AOI22X1 U429 ( .A(n335), .B(n278), .C(N188), .D(n336), .Y(n347) );
OAI21X1 U430 ( .A(n338), .B(n350), .C(n351), .Y(n278) );
AOI22X1 U431 ( .A(N67), .B(n340), .C(N98), .D(n341), .Y(n351) );
INVX1 U432 ( .A(N50), .Y(n350) );
NAND3X1 U433 ( .A(n352), .B(n353), .C(n354), .Y(N226) );
AOI22X1 U434 ( .A(n332), .B(N66), .C(n333), .D(N49), .Y(n354) );
NAND2X1 U435 ( .A(N209), .B(n334), .Y(n353) );
AOI22X1 U436 ( .A(n335), .B(n280), .C(N187), .D(n336), .Y(n352) );
OAI21X1 U437 ( .A(n338), .B(n355), .C(n356), .Y(n280) );
AOI22X1 U438 ( .A(N66), .B(n340), .C(N97), .D(n341), .Y(n356) );
INVX1 U439 ( .A(N49), .Y(n355) );
NAND3X1 U440 ( .A(n357), .B(n358), .C(n359), .Y(N225) );
AOI22X1 U441 ( .A(n332), .B(N65), .C(n333), .D(N48), .Y(n359) );
NAND2X1 U442 ( .A(N208), .B(n334), .Y(n358) );
AOI22X1 U443 ( .A(n335), .B(n282), .C(N186), .D(n336), .Y(n357) );
OAI21X1 U444 ( .A(n338), .B(n360), .C(n361), .Y(n282) );
AOI22X1 U445 ( .A(N65), .B(n340), .C(N96), .D(n341), .Y(n361) );
INVX1 U446 ( .A(N48), .Y(n360) );
NAND3X1 U447 ( .A(n362), .B(n363), .C(n364), .Y(N224) );
AOI22X1 U448 ( .A(n332), .B(N64), .C(n333), .D(N47), .Y(n364) );
NAND2X1 U449 ( .A(N207), .B(n334), .Y(n363) );
```

```
AOI22X1 U450 ( .A(n335), .B(n284), .C(N185), .D(n336), .Y(n362) );
OAI21X1 U451 ( .A(n338), .B(n365), .C(n366), .Y(n284) );
AOI22X1 U452 ( .A(N64), .B(n340), .C(N95), .D(n341), .Y(n366) );
INVX1 U453 ( .A(N47), .Y(n365) );
NAND3X1 U454 ( .A(n367), .B(n368), .C(n369), .Y(N223) );
AOI22X1 U455 ( .A(n332), .B(N63), .C(n397), .D(n333), .Y(n369) );
NAND2X1 U456 ( .A(n397), .B(n334), .Y(n368) );
AOI22X1 U457 ( .A(n335), .B(n286), .C(n285), .D(n336), .Y(n367) );
OAI21X1 U458 ( .A(n425), .B(n338), .C(n370), .Y(n286) );
AOI22X1 U459 ( .A(N63), .B(n340), .C(N94), .D(n341), .Y(n370) );
NAND3X1 U460 ( .A(n371), .B(n372), .C(n373), .Y(N222) );
AOI22X1 U461 ( .A(n332), .B(n374), .C(N45), .D(n333), .Y(n373) );
NAND2X1 U462 ( .A(N45), .B(n334), .Y(n372) );
AOI22X1 U463 ( .A(n335), .B(n288), .C(N183), .D(n336), .Y(n371) );
OAI21X1 U464 ( .A(n374), .B(n338), .C(n375), .Y(n288) );
AOI22X1 U465 ( .A(n374), .B(n340), .C(N93), .D(n341), .Y(n375) );
NAND3X1 U466 ( .A(n376), .B(n377), .C(n378), .Y(N221) );
AOI21X1 U467 ( .A(n327), .B(n274), .C(n379), .Y(n378) );
MUX2X1 U468 ( .B(n380), .A(n381), .S(n338), .Y(n379) );
NAND2X1 U469 ( .A(n382), .B(n338), .Y(n274) );
OAI22X1 U470 ( .A(n383), .B(n384), .C(n341), .D(n340), .Y(n382) );
NAND3X1 U471 ( .A(n386), .B(n338), .C(N80), .Y(n385) );
NOR2X1 U472 ( .A(n386), .B(throttle), .Y(n341) );
INVX1 U473 ( .A(N79), .Y(n386) );
NAND3X1 U474 ( .A(n387), .B(n388), .C(n389), .Y(n384) );
NOR2X1 U475 ( .A(n390), .B(n391), .Y(n389) );
XNOR2X1 U476 ( .A(n203), .B(n428), .Y(n391) );
XOR2X1 U477 ( .A(n422), .B(cruisespeed[4]), .Y(n390) );
XNOR2X1 U478 ( .A(n427), .B(n420), .Y(n388) );
XNOR2X1 U479 ( .A(cruisespeed[7]), .B(n419), .Y(n387) );
NAND3X1 U480 ( .A(n392), .B(n393), .C(n394), .Y(n383) );
NOR2X1 U481 ( .A(n395), .B(n396), .Y(n394) );
XNOR2X1 U482 ( .A(n397), .B(cruisespeed[1]), .Y(n396) );
XNOR2X1 U483 ( .A(n374), .B(n433), .Y(n395) );
XNOR2X1 U484 ( .A(cruisespeed[2]), .B(n424), .Y(n393) );
XNOR2X1 U485 ( .A(n430), .B(n423), .Y(n392) );
INVX1 U486 ( .A(N220), .Y(n377) );
OAI21X1 U487 ( .A(throttle), .B(n297), .C(n398), .Y(N219) );
NAND3X1 U488 ( .A(n399), .B(n400), .C(n398), .Y(N218) );
INVX1 U489 ( .A(n401), .Y(n398) );
OAI21X1 U490 ( .A(n402), .B(n298), .C(n299), .Y(n401) );
NAND3X1 U491 ( .A(set_0011), .B(n403), .C(n404), .Y(n299) );
MUX2X1 U492 ( .B(n381), .A(n380), .S(n338), .Y(n404) );
```

```
  OAI21X1 U493 ( .A(n203), .B(n405), .C(n406), .Y(n403) );
  OAI21X1 U494 ( .A(n422), .B(n423), .C(n407), .Y(n405) );
  INVX1 U495 ( .A(n335), .Y(n400) );
  NOR2X1 U496 ( .A(n323), .B(brake), .Y(n335) );
  OAI21X1 U497 ( .A(n328), .B(n307), .C(throttle), .Y(n399) );
  NAND3X1 U498 ( .A(n301), .B(n297), .C(n408), .Y(N217) );
  AOI21X1 U499 ( .A(state[2]), .B(n409), .C(N220), .Y(n408) );
  OAI21X1 U500 ( .A(n410), .B(n298), .C(n300), .Y(N220) );
  INVX1 U501 ( .A(n336), .Y(n300) );
  NOR2X1 U502 ( .A(n410), .B(n323), .Y(n336) );
  NOR2X1 U503 ( .A(n332), .B(n333), .Y(n297) );
  NAND3X1 U504 ( .A(n411), .B(n412), .C(state[0]), .Y(n380) );
  NAND3X1 U505 ( .A(n413), .B(n412), .C(state[1]), .Y(n381) );
  AOI21X1 U506 ( .A(n327), .B(cancel), .C(n414), .Y(n301) );
  OAI21X1 U507 ( .A(n298), .B(n402), .C(n376), .Y(n414) );
  INVX1 U508 ( .A(n328), .Y(n376) );
  NOR2X1 U509 ( .A(n409), .B(state[2]), .Y(n328) );
  NAND3X1 U510 ( .A(n415), .B(n410), .C(resume), .Y(n402) );
  INVX1 U511 ( .A(brake), .Y(n410) );
  NAND3X1 U512 ( .A(n406), .B(n203), .C(n416), .Y(n415) );
  NOR2X1 U513 ( .A(n423), .B(n407), .Y(n416) );
  NAND3X1 U514 ( .A(n374), .B(n397), .C(n417), .Y(n407) );
  NOR2X1 U515 ( .A(n422), .B(n424), .Y(n417) );
  INVX1 U516 ( .A(n425), .Y(n397) );
  INVX1 U517 ( .A(N45), .Y(n374) );
  NOR2X1 U518 ( .A(n419), .B(n420), .Y(n406) );
  INVX1 U519 ( .A(n334), .Y(n298) );
  NOR2X1 U520 ( .A(n412), .B(n409), .Y(n334) );
  NAND2X1 U521 ( .A(n411), .B(n413), .Y(n409) );
  INVX1 U522 ( .A(state[0]), .Y(n413) );
  INVX1 U523 ( .A(state[1]), .Y(n411) );
  INVX1 U524 ( .A(n323), .Y(n327) );
  NAND3X1 U525 ( .A(state[0]), .B(n412), .C(state[1]), .Y(n323) );
  INVX1 U526 ( .A(state[2]), .Y(n412) );
  AND2X1 U527 ( .A(nextstate[2]), .B(n418), .Y(N19) );
  AND2X1 U528 ( .A(nextstate[1]), .B(n418), .Y(N18) );
  AND2X1 U529 ( .A(nextstate[0]), .B(n418), .Y(N17) );
  INVX1 U530 ( .A(reset), .Y(n418) );
endmodule
```