# Tutorial 6 VHDL

Kartik Patel

13116036

## 8x1 enable Low Multiplexer

### Code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity MUX\_8x1 is

port (input7: in std\_logic;

input6: in std\_logic;

input5: in std\_logic;

input4: in std\_logic;

input3: in std\_logic;

input2: in std\_logic;

input1: in std\_logic;

input0: in std\_logic;

EN\_L: in std\_logic;

SEL: in BIT\_vector(2 downto 0);

output : out std\_logic := '0');

end MUX\_8x1;

architecture Behavioral of MUX\_8x1 is

begin

p1 : process(SEL, EN\_L, input0, input1, input2, input3, input4, input5, input6, input7)

begin

if EN\_L = '0' then

case SEL is

when "000" => output<=input0;

when "001" => output<=input1;

when "010" => output<=input2;

when "011" => output<=input3;

when "100" => output<=input4;

when "101" => output<=input5;

when "110" => output<=input6;

when "111" => output<=input7;

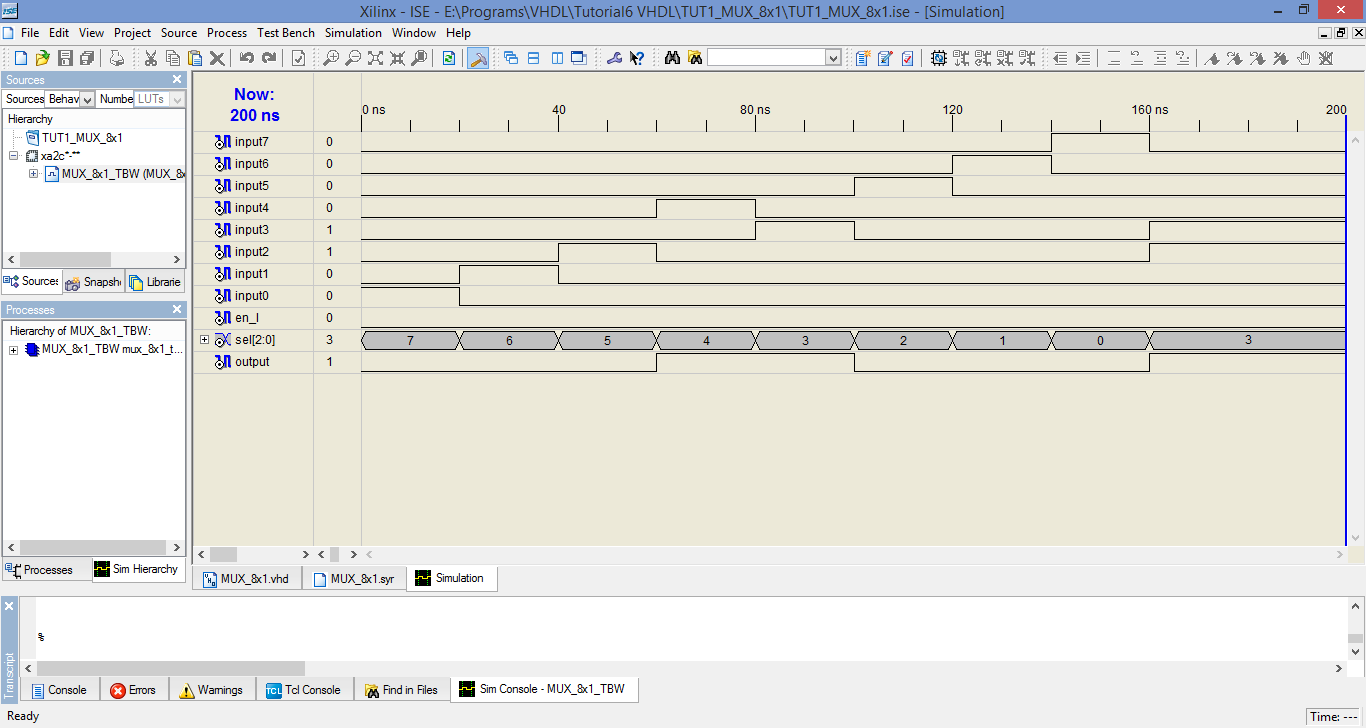
end case;

end if;

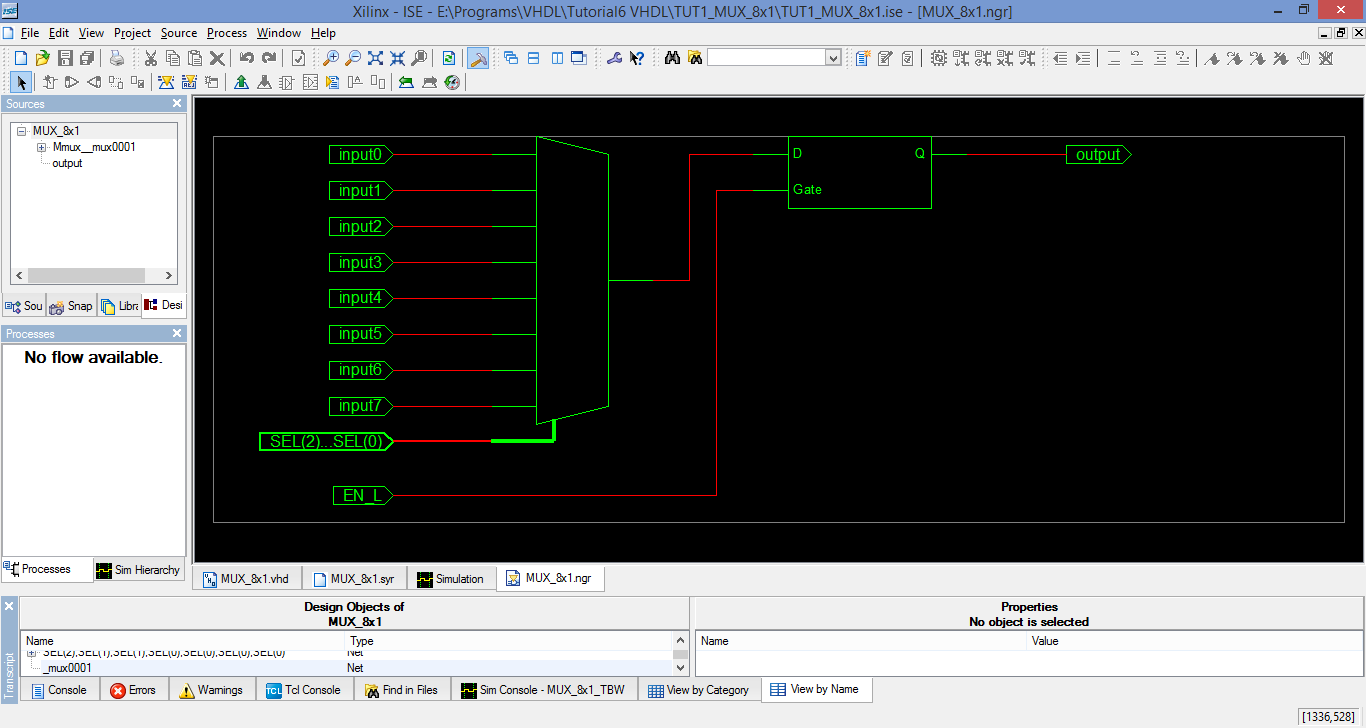
end process;

end Behavioral;

### Simulation:



### RTL Schematic:



## 4- Bit Combined Adder/Subtractor:

### VHDL Code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity FOUR\_BIT\_COMB\_ADDSUB is

port ( a0, a1, a2, a3 : in std\_logic;

b0, b1, b2, b3 : in std\_logic;

m : in std\_logic;

cout : out std\_logic;

sum0, sum1, sum2, sum3 : out std\_logic;

overflow : out std\_logic);

end FOUR\_BIT\_COMB\_ADDSUB;

architecture Behavioral of FOUR\_BIT\_COMB\_ADDSUB is

signal c0, c1, c2, c3, c4 : std\_logic;

begin

c0 <= m;

process(a0, a1, a2, a3, b0, b1, b2, b3, m, c0)

begin

if m='1' then

sum0 <= (a0 xor (not b0)) xor c0;

c1 <= (c0 and (a0 xor (not b0))) or (a0 and (not b0));

sum1 <= (a1 xor (not b1)) xor c1;

c2 <= (c1 and (a1 xor (not b1))) or (a1 and (not b1));

sum2 <= (a2 xor (not b2)) xor c2;

c3 <= (c2 and (a2 xor (not b2))) or (a2 and (not b2));

sum3 <= (a3 xor (not b3)) xor c3;

c4 <= (c3 and (a3 xor (not b3))) or (a3 and (not b3));

elsif m='0' then

sum0 <= (a0 xor b0) xor c0;

c1 <= (c0 and (a0 xor b0)) or (a0 and b0);

sum1 <= (a1 xor b1) xor c1;

c2 <= (c1 and (a1 xor b1)) or (a1 and b1);

sum2 <= (a2 xor b2) xor c2;

c3 <= (c2 and (a2 xor b2)) or (a2 and b2);

sum3 <= (a3 xor b3) xor c3;

c4 <= (c3 and (a3 xor b3)) or (a3 and b3);

end if;

end process;

cout <= c4;

overflow <= (c3 xor c4);

end Behavioral;

### Test bench Simulation:

### Synthesis Report:

Release 8.2i - xst I.31

Copyright (c) 1995-2006 Xilinx, Inc. All rights reserved.

--> Parameter TMPDIR set to ./xst/projnav.tmp

CPU : 0.00 / 0.27 s | Elapsed : 0.00 / 0.00 s

--> Parameter xsthdpdir set to ./xst

CPU : 0.00 / 0.27 s | Elapsed : 0.00 / 0.00 s

--> Reading design: FOUR\_BIT\_COMB\_ADDSUB.prj

TABLE OF CONTENTS

1) Synthesis Options Summary

2) HDL Compilation

3) Design Hierarchy Analysis

4) HDL Analysis

5) HDL Synthesis

5.1) HDL Synthesis Report

6) Advanced HDL Synthesis

6.1) Advanced HDL Synthesis Report

7) Low Level Synthesis

8) Partition Report

9) Final Report

=========================================================================

\* Synthesis Options Summary \*

=========================================================================

---- Source Parameters

Input File Name : "FOUR\_BIT\_COMB\_ADDSUB.prj"

Input Format : mixed

Ignore Synthesis Constraint File : NO

---- Target Parameters

Output File Name : "FOUR\_BIT\_COMB\_ADDSUB"

Output Format : NGC

Target Device : Automotive CoolRunner2

---- Source Options

Top Module Name : FOUR\_BIT\_COMB\_ADDSUB

Automatic FSM Extraction : YES

FSM Encoding Algorithm : Auto

Mux Extraction : YES

Resource Sharing : YES

---- Target Options

Add IO Buffers : YES

MACRO Preserve : YES

XOR Preserve : YES

Equivalent register Removal : YES

---- General Options

Optimization Goal : Speed

Optimization Effort : 1

Keep Hierarchy : YES

RTL Output : Yes

Hierarchy Separator : /

Bus Delimiter : <>

Case Specifier : maintain

---- Other Options

lso : FOUR\_BIT\_COMB\_ADDSUB.lso

verilog2001 : YES

safe\_implementation : No

Clock Enable : YES

wysiwyg : NO

=========================================================================

=========================================================================

\* HDL Compilation \*

=========================================================================

Compiling vhdl file "E:/Programs/VHDL/Tutorial6 VHDL/TUT2\_FOUR\_BIT\_COMB\_ADDSUB/FOUR\_BIT\_COMB\_ADDSUB.vhd" in Library work.

Entity <four\_bit\_comb\_addsub> compiled.

Entity <four\_bit\_comb\_addsub> (Architecture <behavioral>) compiled.

=========================================================================

\* Design Hierarchy Analysis \*

=========================================================================

Analyzing hierarchy for entity <FOUR\_BIT\_COMB\_ADDSUB> in library <work> (architecture <behavioral>).

Building hierarchy successfully finished.

=========================================================================

\* HDL Analysis \*

=========================================================================

Analyzing Entity <FOUR\_BIT\_COMB\_ADDSUB> in library <work> (Architecture <behavioral>).

Entity <FOUR\_BIT\_COMB\_ADDSUB> analyzed. Unit <FOUR\_BIT\_COMB\_ADDSUB> generated.

=========================================================================

\* HDL Synthesis \*

=========================================================================

Performing bidirectional port resolution...

Synthesizing Unit <FOUR\_BIT\_COMB\_ADDSUB>.

Related source file is "E:/Programs/VHDL/Tutorial6 VHDL/TUT2\_FOUR\_BIT\_COMB\_ADDSUB/FOUR\_BIT\_COMB\_ADDSUB.vhd".

Found 1-bit xor2 for signal <overflow>.

Found 1-bit xor2 for signal <$xor0000> created at line 20.

Found 1-bit xor2 for signal <$xor0001> created at line 29.

Found 1-bit xor2 for signal <$xor0002> created at line 22.

Found 1-bit xor2 for signal <$xor0003> created at line 31.

Found 1-bit xor2 for signal <$xor0004> created at line 24.

Found 1-bit xor2 for signal <$xor0005> created at line 33.

Found 1-bit xor2 for signal <$xor0006> created at line 26.

Found 1-bit xor2 for signal <$xor0007> created at line 35.

Found 1-bit xor2 for signal <$xor0032> created at line 21.

Found 1-bit xor2 for signal <$xor0033> created at line 30.

Found 1-bit xor2 for signal <$xor0034> created at line 23.

Found 1-bit xor2 for signal <$xor0035> created at line 32.

Found 1-bit xor2 for signal <$xor0036> created at line 25.

Found 1-bit xor2 for signal <$xor0037> created at line 34.

Found 1-bit xor2 for signal <$xor0038> created at line 27.

Found 1-bit xor2 for signal <$xor0039> created at line 36.

Found 1-bit xor2 for signal <$xor0040> created at line 20.

Found 1-bit xor2 for signal <$xor0041> created at line 22.

Found 1-bit xor2 for signal <$xor0042> created at line 24.

Found 1-bit xor2 for signal <$xor0043> created at line 26.

Summary:

inferred 21 Xor(s).

Unit <FOUR\_BIT\_COMB\_ADDSUB> synthesized.

=========================================================================

HDL Synthesis Report

Macro Statistics

# Xors : 21

1-bit xor2 : 21

=========================================================================

=========================================================================

\* Advanced HDL Synthesis \*

=========================================================================

=========================================================================

Advanced HDL Synthesis Report

Found no macro

=========================================================================

=========================================================================

\* Low Level Synthesis \*

=========================================================================

Optimizing unit <FOUR\_BIT\_COMB\_ADDSUB> ...

=========================================================================

\* Partition Report \*

=========================================================================

Partition Implementation Status

-------------------------------

No Partitions were found in this design.

-------------------------------

=========================================================================

\* Final Report \*

=========================================================================

Final Results

RTL Top Level Output File Name : FOUR\_BIT\_COMB\_ADDSUB.ngr

Top Level Output File Name : FOUR\_BIT\_COMB\_ADDSUB

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : YES

Target Technology : Automotive CoolRunner2

Macro Preserve : YES

XOR Preserve : YES

Clock Enable : YES

wysiwyg : NO

Design Statistics

# IOs : 15

Cell Usage :

# BELS : 68

# AND2 : 22

# INV : 11

# OR2 : 11

# XOR2 : 24

# IO Buffers : 15

# IBUF : 9

# OBUF : 6

=========================================================================

CPU : 6.11 / 6.39 s | Elapsed : 6.00 / 6.00 s

-->

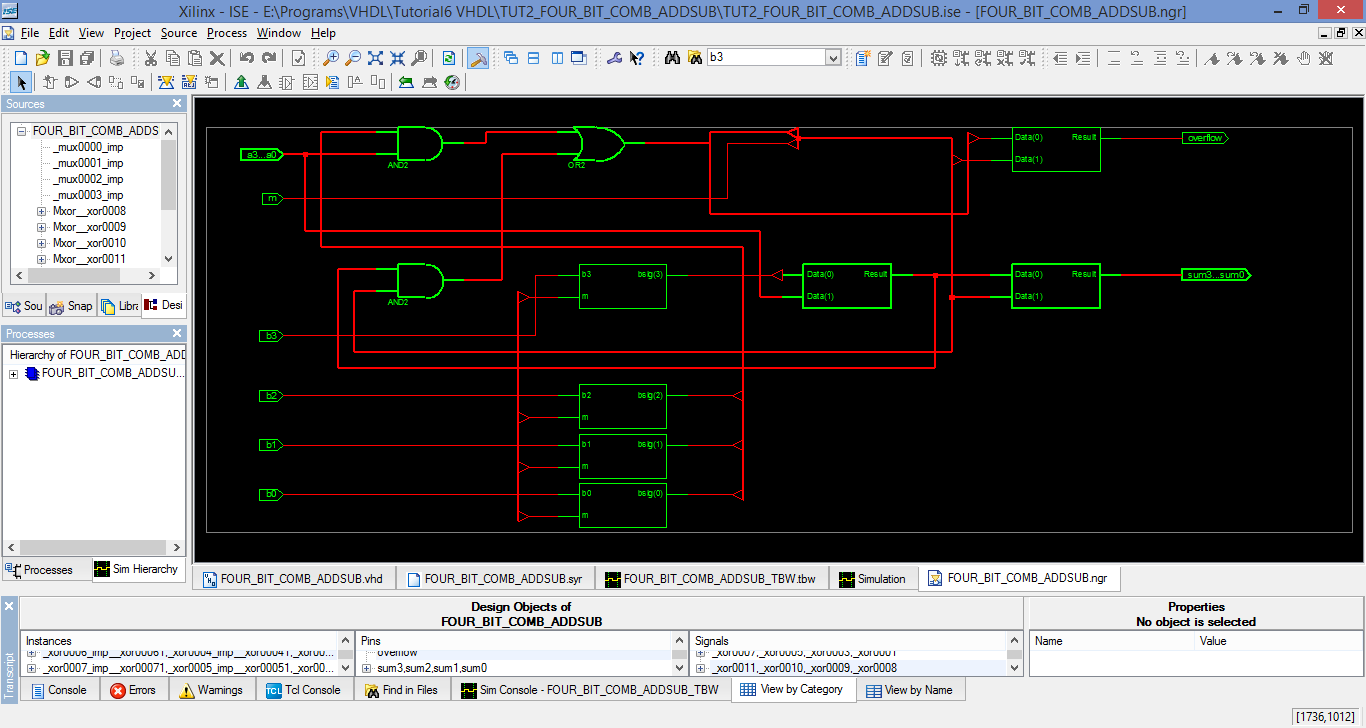
Total memory usage is 142148 kilobytes

Number of errors : 0 ( 0 filtered)

Number of warnings : 0 ( 0 filtered)

Number of infos : 0 ( 0 filtered)

### RTL Schematic:



## 8-bit Priority Encoder:

### VHDL Code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity EIGHT\_BIT\_PRI\_ENCODER is

port ( EI : in std\_logic;

I0,I1,I2,I3 : in std\_logic;

I4, I5, I6, I7 : in std\_logic;

A2, A1, A0 : out std\_logic := '0';

EO : out std\_logic := '1';

GS : out std\_logic := '0');

end EIGHT\_BIT\_PRI\_ENCODER;

architecture Behavioral of EIGHT\_BIT\_PRI\_ENCODER is

begin

p1: process(EI, I0, I1, I2, I3, I4, I5, I6, I7)

begin

if EI = '1' then

A2 <= '1';

A1 <= '1';

A0 <= '1';

GS <= '1';

EO <= '1';

elsif I7 = '0' then

A2 <= '0';

A1 <= '0';

A0 <= '0';

GS <= '0';

EO <= '1';

elsif I6 = '0' then

A2 <= '0';

A1 <= '0';

A0 <= '1';

GS <= '0';

EO <= '1';

elsif I5 = '0' then

A2 <= '0';

A1 <= '1';

A0 <= '0';

GS <= '0';

EO <= '1';

elsif I4 = '0' then

A2 <= '0';

A1 <= '1';

A0 <= '1';

GS <= '0';

EO <= '1';

elsif I3 = '0' then

A2 <= '1';

A1 <= '0';

A0 <= '0';

GS <= '0';

EO <= '1';

elsif I2 = '0' then

A2 <= '1';

A1 <= '0';

A0 <= '1';

GS <= '0';

EO <= '1';

elsif I1 = '0' then

A2 <= '1';

A1 <= '1';

A0 <= '0';

GS <= '0';

EO <= '1';

elsif I0 = '0' then

A2 <= '1';

A1 <= '1';

A0 <= '1';

GS <= '0';

EO <= '1';

else

A2 <= '1';

A1 <= '1';

A0 <= '1';

GS <= '1';

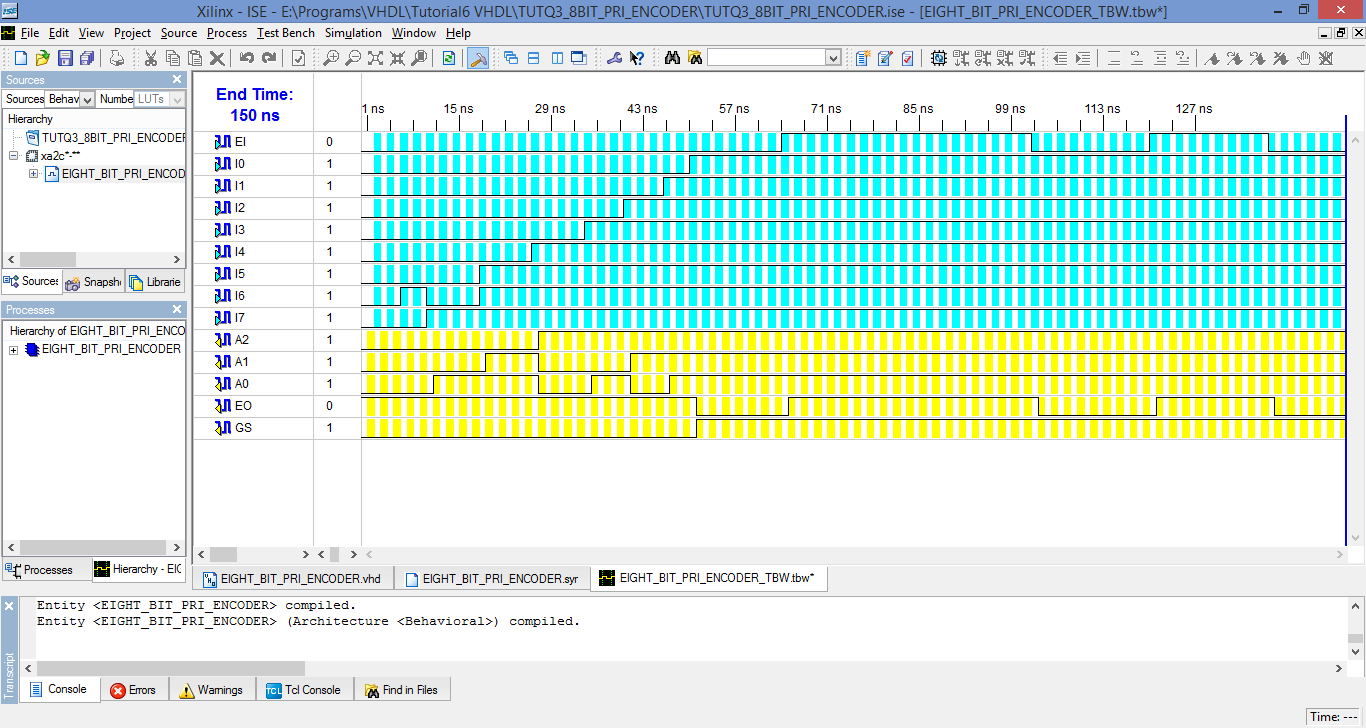
EO <= '0';

end if;

end process;

end Behavioral;

### Simulation:



### Synthesis Report:

Release 8.2i - xst I.31

Copyright (c) 1995-2006 Xilinx, Inc. All rights reserved.

--> Parameter TMPDIR set to ./xst/projnav.tmp

CPU : 0.00 / 0.28 s | Elapsed : 0.00 / 0.00 s

--> Parameter xsthdpdir set to ./xst

CPU : 0.00 / 0.28 s | Elapsed : 0.00 / 0.00 s

--> Reading design: EIGHT\_BIT\_PRI\_ENCODER.prj

TABLE OF CONTENTS

1) Synthesis Options Summary

2) HDL Compilation

3) Design Hierarchy Analysis

4) HDL Analysis

5) HDL Synthesis

5.1) HDL Synthesis Report

6) Advanced HDL Synthesis

6.1) Advanced HDL Synthesis Report

7) Low Level Synthesis

8) Partition Report

9) Final Report

=========================================================================

\* Synthesis Options Summary \*

=========================================================================

---- Source Parameters

Input File Name : "EIGHT\_BIT\_PRI\_ENCODER.prj"

Input Format : mixed

Ignore Synthesis Constraint File : NO

---- Target Parameters

Output File Name : "EIGHT\_BIT\_PRI\_ENCODER"

Output Format : NGC

Target Device : Automotive CoolRunner2

---- Source Options

Top Module Name : EIGHT\_BIT\_PRI\_ENCODER

Automatic FSM Extraction : YES

FSM Encoding Algorithm : Auto

Mux Extraction : YES

Resource Sharing : YES

---- Target Options

Add IO Buffers : YES

MACRO Preserve : YES

XOR Preserve : YES

Equivalent register Removal : YES

---- General Options

Optimization Goal : Speed

Optimization Effort : 1

Keep Hierarchy : YES

RTL Output : Yes

Hierarchy Separator : /

Bus Delimiter : <>

Case Specifier : maintain

---- Other Options

lso : EIGHT\_BIT\_PRI\_ENCODER.lso

verilog2001 : YES

safe\_implementation : No

Clock Enable : YES

wysiwyg : NO

=========================================================================

=========================================================================

\* HDL Compilation \*

=========================================================================

Compiling vhdl file "E:/Programs/VHDL/Tutorial6 VHDL/TUTQ3\_8BIT\_PRI\_ENCODER/EIGHT\_BIT\_PRI\_ENCODER.vhd" in Library work.

Entity <eight\_bit\_pri\_encoder> compiled.

Entity <eight\_bit\_pri\_encoder> (Architecture <behavioral>) compiled.

=========================================================================

\* Design Hierarchy Analysis \*

=========================================================================

Analyzing hierarchy for entity <EIGHT\_BIT\_PRI\_ENCODER> in library <work> (architecture <behavioral>).

Building hierarchy successfully finished.

=========================================================================

\* HDL Analysis \*

=========================================================================

Analyzing Entity <EIGHT\_BIT\_PRI\_ENCODER> in library <work> (Architecture <behavioral>).

Entity <EIGHT\_BIT\_PRI\_ENCODER> analyzed. Unit <EIGHT\_BIT\_PRI\_ENCODER> generated.

=========================================================================

\* HDL Synthesis \*

=========================================================================

Performing bidirectional port resolution...

Synthesizing Unit <EIGHT\_BIT\_PRI\_ENCODER>.

Related source file is "E:/Programs/VHDL/Tutorial6 VHDL/TUTQ3\_8BIT\_PRI\_ENCODER/EIGHT\_BIT\_PRI\_ENCODER.vhd".

Unit <EIGHT\_BIT\_PRI\_ENCODER> synthesized.

=========================================================================

HDL Synthesis Report

Found no macro

=========================================================================

=========================================================================

\* Advanced HDL Synthesis \*

=========================================================================

=========================================================================

Advanced HDL Synthesis Report

Found no macro

=========================================================================

=========================================================================

\* Low Level Synthesis \*

=========================================================================

Optimizing unit <EIGHT\_BIT\_PRI\_ENCODER> ...

=========================================================================

\* Partition Report \*

=========================================================================

Partition Implementation Status

-------------------------------

No Partitions were found in this design.

-------------------------------

=========================================================================

\* Final Report \*

=========================================================================

Final Results

RTL Top Level Output File Name : EIGHT\_BIT\_PRI\_ENCODER.ngr

Top Level Output File Name : EIGHT\_BIT\_PRI\_ENCODER

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : YES

Target Technology : Automotive CoolRunner2

Macro Preserve : YES

XOR Preserve : YES

Clock Enable : YES

wysiwyg : NO

Design Statistics

# IOs : 14

Cell Usage :

# BELS : 31

# AND2 : 6

# AND4 : 1

# AND8 : 1

# INV : 13

# OR2 : 8

# OR3 : 1

# OR8 : 1

# IO Buffers : 14

# IBUF : 9

# OBUF : 5

=========================================================================

CPU : 6.33 / 6.61 s | Elapsed : 6.00 / 6.00 s

-->

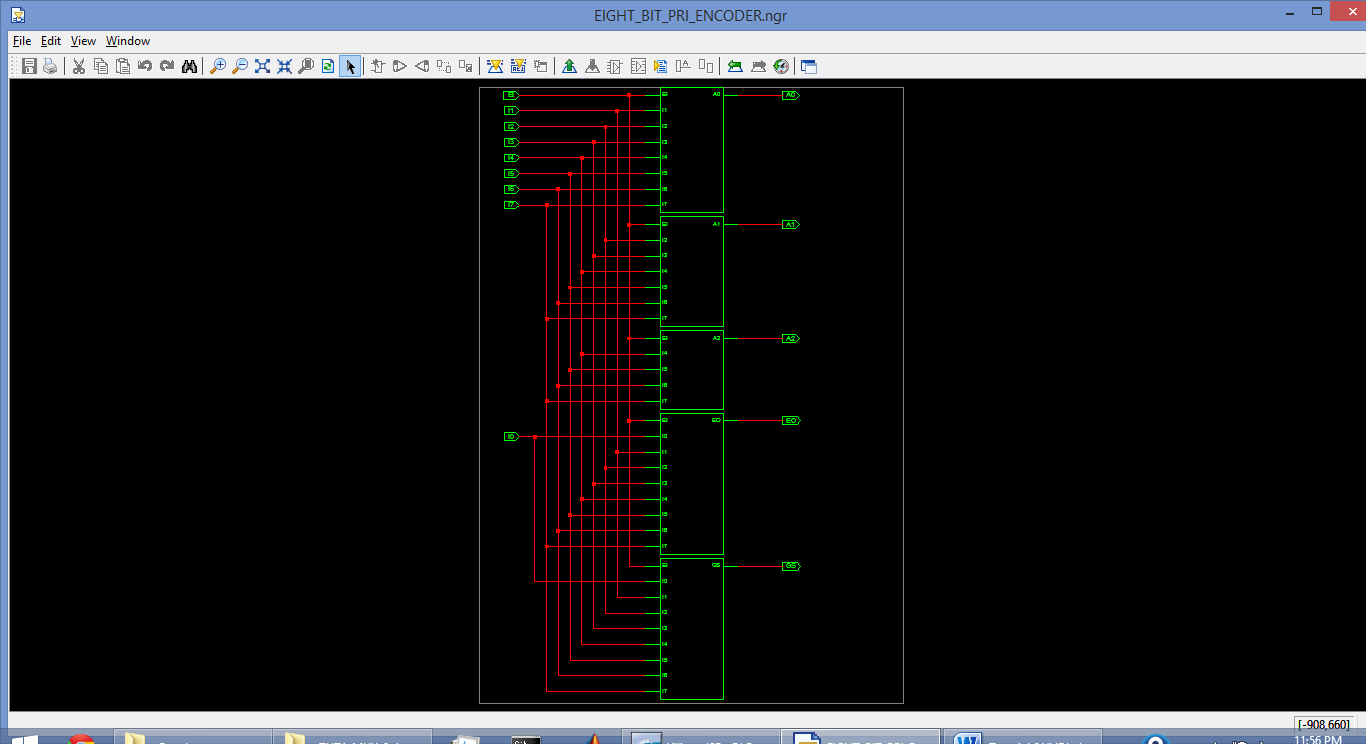
Total memory usage is 141956 kilobytes

Number of errors : 0 ( 0 filtered)

Number of warnings : 0 ( 0 filtered)

Number of infos : 0 ( 0 filtered)

### RTL Schematic:



## Schematic Solution:

### VHDL Code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity SCH\_SOL is

port (A, B, C, D, E, F: in std\_logic;

O1,O2,O3,O4,O5,O6,O7,O8: out std\_logic);

end SCH\_SOL;

architecture Behavioral of SCH\_SOL is

begin

O1 <= A or B or C or D;

O2 <= A or B or (not C) or D;

O3 <= A or B or C or (not D);

O4 <= A or B or (not(C and D));

O5 <= not(E and F) and (not (C or D));

O6 <= not(E and F) and C and (not D);

O7 <= not(E and F) and (not C) and D;

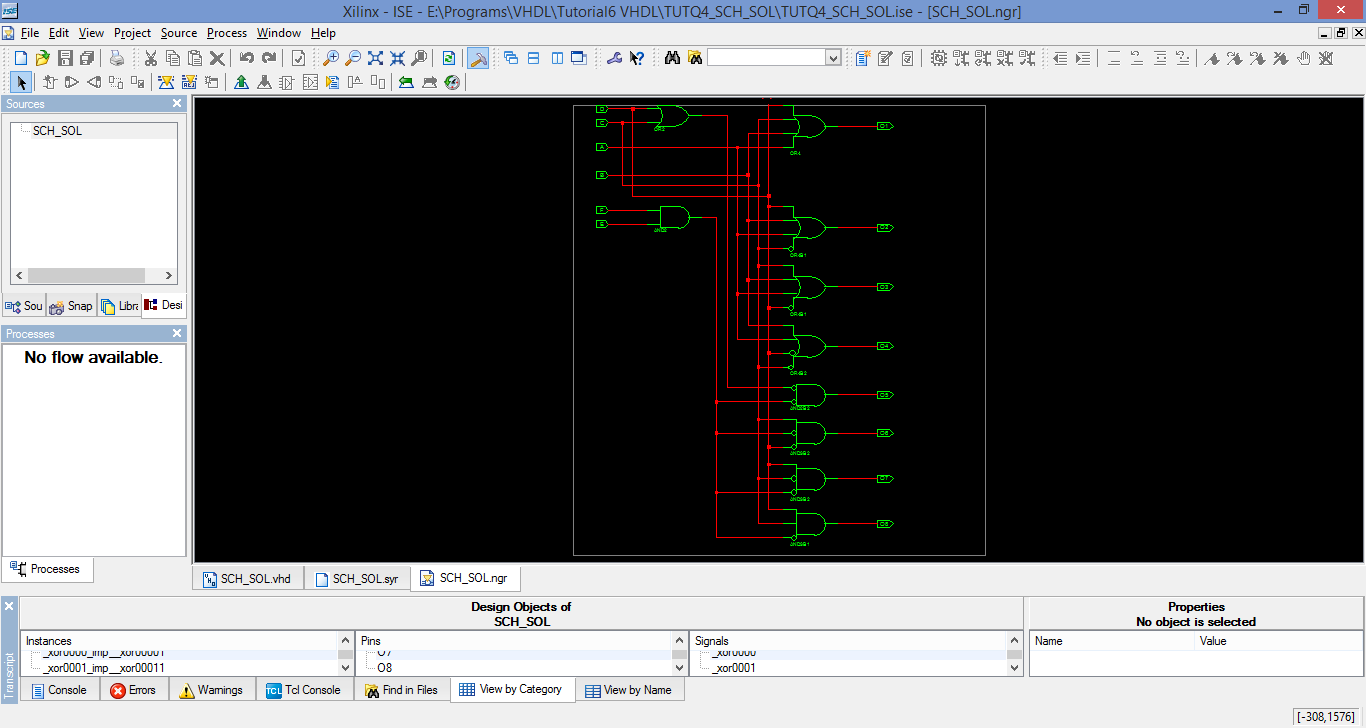
O8 <= not(E and F) and C and D;

end Behavioral;

### Test Bench Simulation:



### RTL Schematic:



## 4 Bit Comparator

### VHDL Code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity FOUR\_BIT\_COMP is

port (X3, X2, X1, X0 : in std\_logic;

Y3, Y2, Y1, Y0 : in std\_logic;

G : out std\_logic;

L : out std\_logic;

E : out std\_logic);

end FOUR\_BIT\_COMP;

architecture Behavioral of FOUR\_BIT\_COMP is

begin

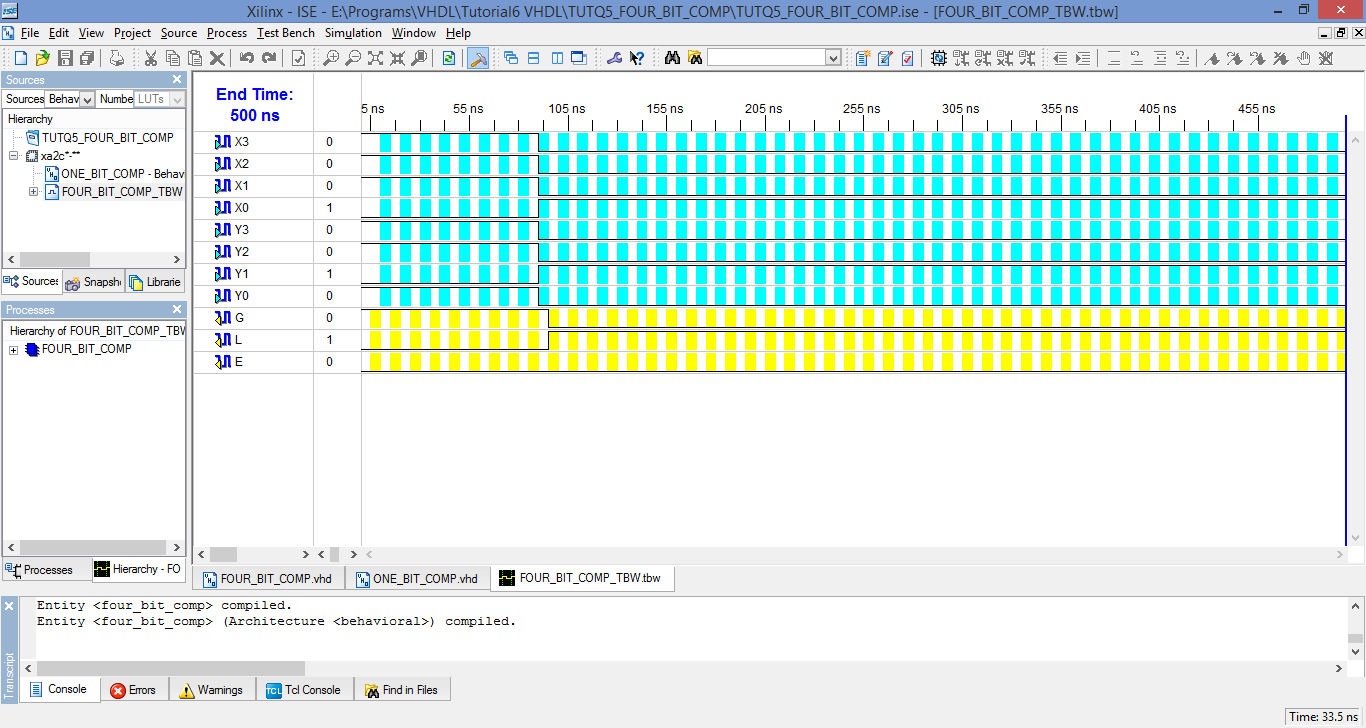
G <= (X3 and (not Y3)) or ((X3 xnor Y3) and (X2 and (not Y2))) or ((X3 xnor Y3) and (X2 xnor Y2) and (X1 and (not Y1))) or ((X3 xnor Y3) and (X2 xnor Y2) and (X1 xnor Y1) and (X0 and (not Y0)));

L <= (Y3 and (not X3)) or ((X3 xnor Y3) and (Y2 and (not X2))) or ((X3 xnor Y3) and (X2 xnor Y2) and (Y1 and (not X1))) or ((X3 xnor Y3) and (X2 xnor Y2) and (X1 xnor Y1) and (Y0 and (not X0)));

E <= (X3 xnor Y3) and (X2 xnor Y2) and (X1 xnor Y1) and (X0 xnor Y0);

end Behavioral;

### Test Bench Simulation:



### Synthesis Report:

Release 8.2i - xst I.31

Copyright (c) 1995-2006 Xilinx, Inc. All rights reserved.

--> Parameter TMPDIR set to ./xst/projnav.tmp

CPU : 0.00 / 0.27 s | Elapsed : 0.00 / 0.00 s

--> Parameter xsthdpdir set to ./xst

CPU : 0.00 / 0.27 s | Elapsed : 0.00 / 0.00 s

--> Reading design: FOUR\_BIT\_COMP.prj

TABLE OF CONTENTS

1) Synthesis Options Summary

2) HDL Compilation

3) Design Hierarchy Analysis

4) HDL Analysis

5) HDL Synthesis

5.1) HDL Synthesis Report

6) Advanced HDL Synthesis

6.1) Advanced HDL Synthesis Report

7) Low Level Synthesis

8) Partition Report

9) Final Report

=========================================================================

\* Synthesis Options Summary \*

=========================================================================

---- Source Parameters

Input File Name : "FOUR\_BIT\_COMP.prj"

Input Format : mixed

Ignore Synthesis Constraint File : NO

---- Target Parameters

Output File Name : "FOUR\_BIT\_COMP"

Output Format : NGC

Target Device : Automotive CoolRunner2

---- Source Options

Top Module Name : FOUR\_BIT\_COMP

Automatic FSM Extraction : YES

FSM Encoding Algorithm : Auto

Mux Extraction : YES

Resource Sharing : YES

---- Target Options

Add IO Buffers : YES

MACRO Preserve : YES

XOR Preserve : YES

Equivalent register Removal : YES

---- General Options

Optimization Goal : Speed

Optimization Effort : 1

Keep Hierarchy : YES

RTL Output : Yes

Hierarchy Separator : /

Bus Delimiter : <>

Case Specifier : maintain

---- Other Options

lso : FOUR\_BIT\_COMP.lso

verilog2001 : YES

safe\_implementation : No

Clock Enable : YES

wysiwyg : NO

=========================================================================

=========================================================================

\* HDL Compilation \*

=========================================================================

Compiling vhdl file "E:/Programs/VHDL/Tutorial6 VHDL/TUTQ5\_FOUR\_BIT\_COMP/FOUR\_BIT\_COMP.vhd" in Library work.

Architecture behavioral of Entity four\_bit\_comp is up to date.

=========================================================================

\* Design Hierarchy Analysis \*

=========================================================================

Analyzing hierarchy for entity <FOUR\_BIT\_COMP> in library <work> (architecture <behavioral>).

Building hierarchy successfully finished.

=========================================================================

\* HDL Analysis \*

=========================================================================

Analyzing Entity <FOUR\_BIT\_COMP> in library <work> (Architecture <behavioral>).

Entity <FOUR\_BIT\_COMP> analyzed. Unit <FOUR\_BIT\_COMP> generated.

=========================================================================

\* HDL Synthesis \*

=========================================================================

Performing bidirectional port resolution...

Synthesizing Unit <FOUR\_BIT\_COMP>.

Related source file is "E:/Programs/VHDL/Tutorial6 VHDL/TUTQ5\_FOUR\_BIT\_COMP/FOUR\_BIT\_COMP.vhd".

Found 1-bit xor2 for signal <$xor0000>.

Found 1-bit xor2 for signal <$xor0001>.

Found 1-bit xor2 for signal <$xor0002>.

Found 1-bit xor2 for signal <$xor0003>.

Summary:

inferred 4 Xor(s).

Unit <FOUR\_BIT\_COMP> synthesized.

=========================================================================

HDL Synthesis Report

Macro Statistics

# Xors : 4

1-bit xor2 : 4

=========================================================================

=========================================================================

\* Advanced HDL Synthesis \*

=========================================================================

=========================================================================

Advanced HDL Synthesis Report

Found no macro

=========================================================================

=========================================================================

\* Low Level Synthesis \*

=========================================================================

Optimizing unit <FOUR\_BIT\_COMP> ...

=========================================================================

\* Partition Report \*

=========================================================================

Partition Implementation Status

-------------------------------

No Partitions were found in this design.

-------------------------------

=========================================================================

\* Final Report \*

=========================================================================

Final Results

RTL Top Level Output File Name : FOUR\_BIT\_COMP.ngr

Top Level Output File Name : FOUR\_BIT\_COMP

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : YES

Target Technology : Automotive CoolRunner2

Macro Preserve : YES

XOR Preserve : YES

Clock Enable : YES

wysiwyg : NO

Design Statistics

# IOs : 11

Cell Usage :

# BELS : 39

# AND2 : 2

# AND3 : 2

# AND4 : 3

# AND5 : 2

# INV : 24

# OR4 : 2

# XOR2 : 4

# IO Buffers : 11

# IBUF : 8

# OBUF : 3

=========================================================================

CPU : 6.42 / 6.70 s | Elapsed : 6.00 / 6.00 s

-->

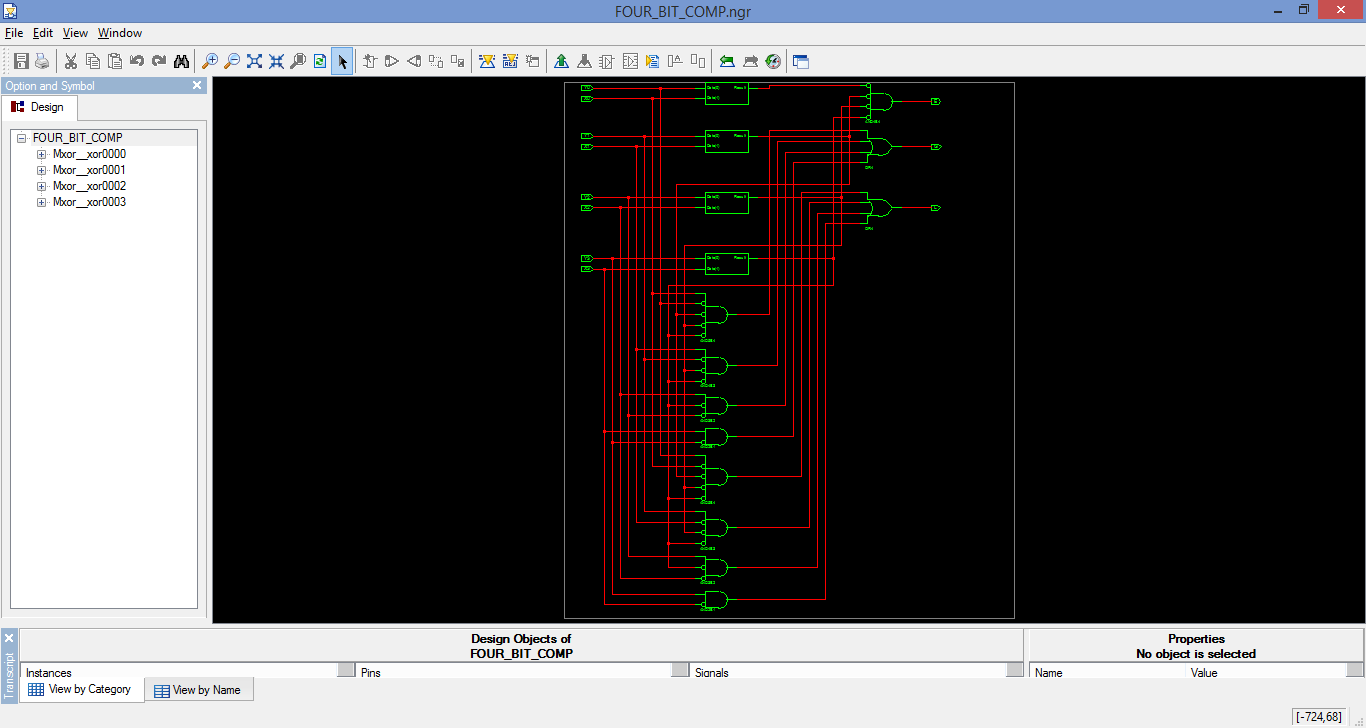
Total memory usage is 141316 kilobytes

Number of errors : 0 ( 0 filtered)

Number of warnings : 0 ( 0 filtered)

Number of infos : 0 ( 0 filtered)

### RTL Schematic:



## Digital Display:

### VHDL Code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity DISPLAY is

port( EN: in std\_logic;

X3, X2, X1, X0 : in std\_logic;

a, b, c, d, e, f, g : out std\_logic);

end DISPLAY;

architecture Behavioral of DISPLAY is

begin

a <= (not EN) and (((X3 xnor X1) and (X2 xnor X0)) or X3 or X1);

b <= (not EN) and ((not X2) or (X1 xnor X0));

c <= (not EN) and (X2 or (not X1) or X0);

d <= (not EN) and (X3 or (X1 and (not X0)) or ((not X0) and (not X2)) or ((not X3) and X0 and (X2 xor X1)));

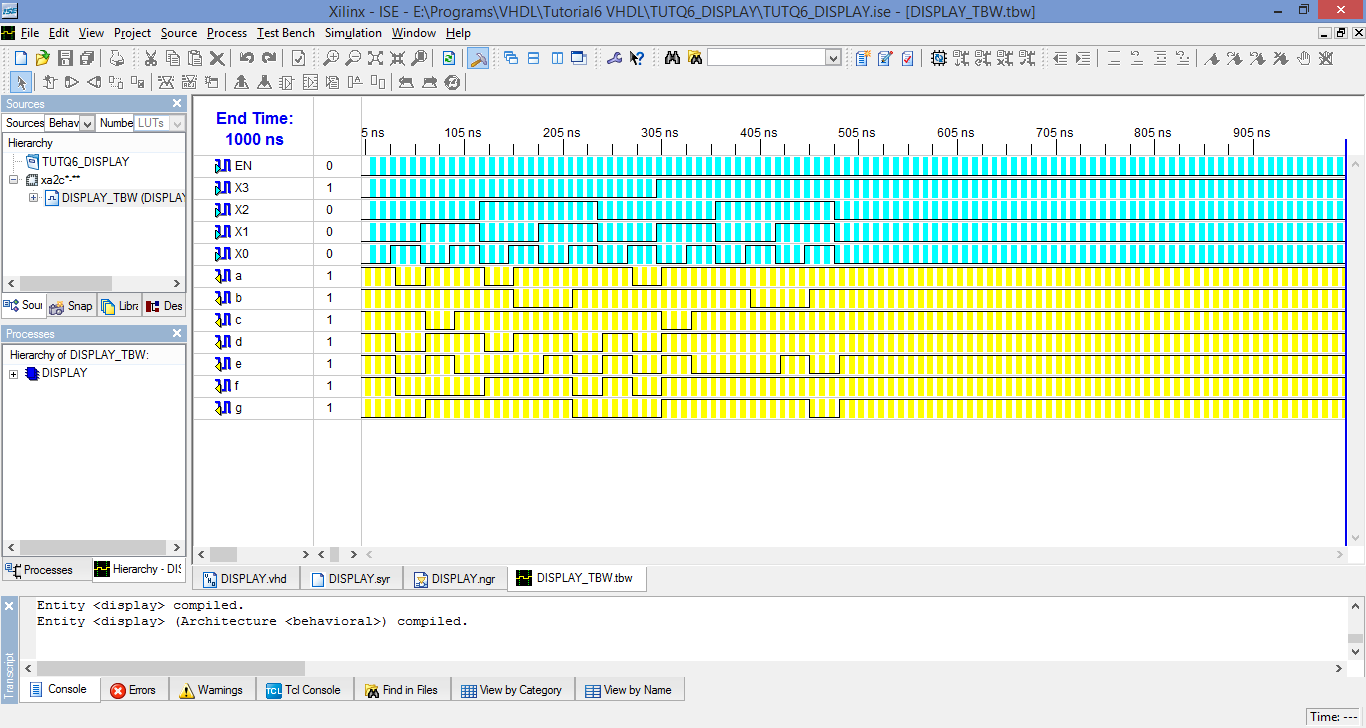
e <= (not EN) and ((not X0) and (X1 or (not X2)));

f <= (not EN) and (((not X1) and ((not X0) or X2)) or ((not X0) and X2) or X3);

g <= (not EN) and ((X3 and (not X2)) or (X1 and (not X0)) or (X1 xor X2));

end Behavioral;

### Test Bench Simulation:



### Synthesis Report:

Release 8.2i - xst I.31

Copyright (c) 1995-2006 Xilinx, Inc. All rights reserved.

--> Parameter TMPDIR set to ./xst/projnav.tmp

CPU : 0.00 / 0.28 s | Elapsed : 0.00 / 1.00 s

--> Parameter xsthdpdir set to ./xst

CPU : 0.00 / 0.28 s | Elapsed : 0.00 / 1.00 s

--> Reading design: DISPLAY.prj

TABLE OF CONTENTS

1) Synthesis Options Summary

2) HDL Compilation

3) Design Hierarchy Analysis

4) HDL Analysis

5) HDL Synthesis

5.1) HDL Synthesis Report

6) Advanced HDL Synthesis

6.1) Advanced HDL Synthesis Report

7) Low Level Synthesis

8) Partition Report

9) Final Report

=========================================================================

\* Synthesis Options Summary \*

=========================================================================

---- Source Parameters

Input File Name : "DISPLAY.prj"

Input Format : mixed

Ignore Synthesis Constraint File : NO

---- Target Parameters

Output File Name : "DISPLAY"

Output Format : NGC

Target Device : Automotive CoolRunner2

---- Source Options

Top Module Name : DISPLAY

Automatic FSM Extraction : YES

FSM Encoding Algorithm : Auto

Mux Extraction : YES

Resource Sharing : YES

---- Target Options

Add IO Buffers : YES

MACRO Preserve : YES

XOR Preserve : YES

Equivalent register Removal : YES

---- General Options

Optimization Goal : Speed

Optimization Effort : 1

Keep Hierarchy : YES

RTL Output : Yes

Hierarchy Separator : /

Bus Delimiter : <>

Case Specifier : maintain

---- Other Options

lso : DISPLAY.lso

verilog2001 : YES

safe\_implementation : No

Clock Enable : YES

wysiwyg : NO

=========================================================================

=========================================================================

\* HDL Compilation \*

=========================================================================

Compiling vhdl file "E:/Programs/VHDL/Tutorial6 VHDL/TUTQ6\_DISPLAY/DISPLAY.vhd" in Library work.

Entity <DISPLAY> compiled.

Entity <DISPLAY> (Architecture <Behavioral>) compiled.

=========================================================================

\* Design Hierarchy Analysis \*

=========================================================================

Analyzing hierarchy for entity <DISPLAY> in library <work> (architecture <Behavioral>).

Building hierarchy successfully finished.

=========================================================================

\* HDL Analysis \*

=========================================================================

Analyzing Entity <DISPLAY> in library <work> (Architecture <Behavioral>).

Entity <DISPLAY> analyzed. Unit <DISPLAY> generated.

=========================================================================

\* HDL Synthesis \*

=========================================================================

Performing bidirectional port resolution...

Synthesizing Unit <DISPLAY>.

Related source file is "E:/Programs/VHDL/Tutorial6 VHDL/TUTQ6\_DISPLAY/DISPLAY.vhd".

Found 1-bit xor2 for signal <$xor0014>.

Found 1-bit xor2 for signal <$xor0015>.

Found 1-bit xor2 for signal <$xor0016> created at line 15.

Summary:

inferred 3 Xor(s).

Unit <DISPLAY> synthesized.

=========================================================================

HDL Synthesis Report

Macro Statistics

# Xors : 3

1-bit xor2 : 3

=========================================================================

=========================================================================

\* Advanced HDL Synthesis \*

=========================================================================

=========================================================================

Advanced HDL Synthesis Report

Found no macro

=========================================================================

=========================================================================

\* Low Level Synthesis \*

=========================================================================

Optimizing unit <DISPLAY> ...

=========================================================================

\* Partition Report \*

=========================================================================

Partition Implementation Status

-------------------------------

No Partitions were found in this design.

-------------------------------

=========================================================================

\* Final Report \*

=========================================================================

Final Results

RTL Top Level Output File Name : DISPLAY.ngr

Top Level Output File Name : DISPLAY

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : YES

Target Technology : Automotive CoolRunner2

Macro Preserve : YES

XOR Preserve : YES

Clock Enable : YES

wysiwyg : NO

Design Statistics

# IOs : 12

Cell Usage :

# BELS : 49

# AND2 : 12

# AND3 : 1

# INV : 20

# OR2 : 12

# OR3 : 1

# XOR2 : 3

# IO Buffers : 12

# IBUF : 5

# OBUF : 7

=========================================================================

CPU : 6.38 / 6.67 s | Elapsed : 6.00 / 7.00 s

-->

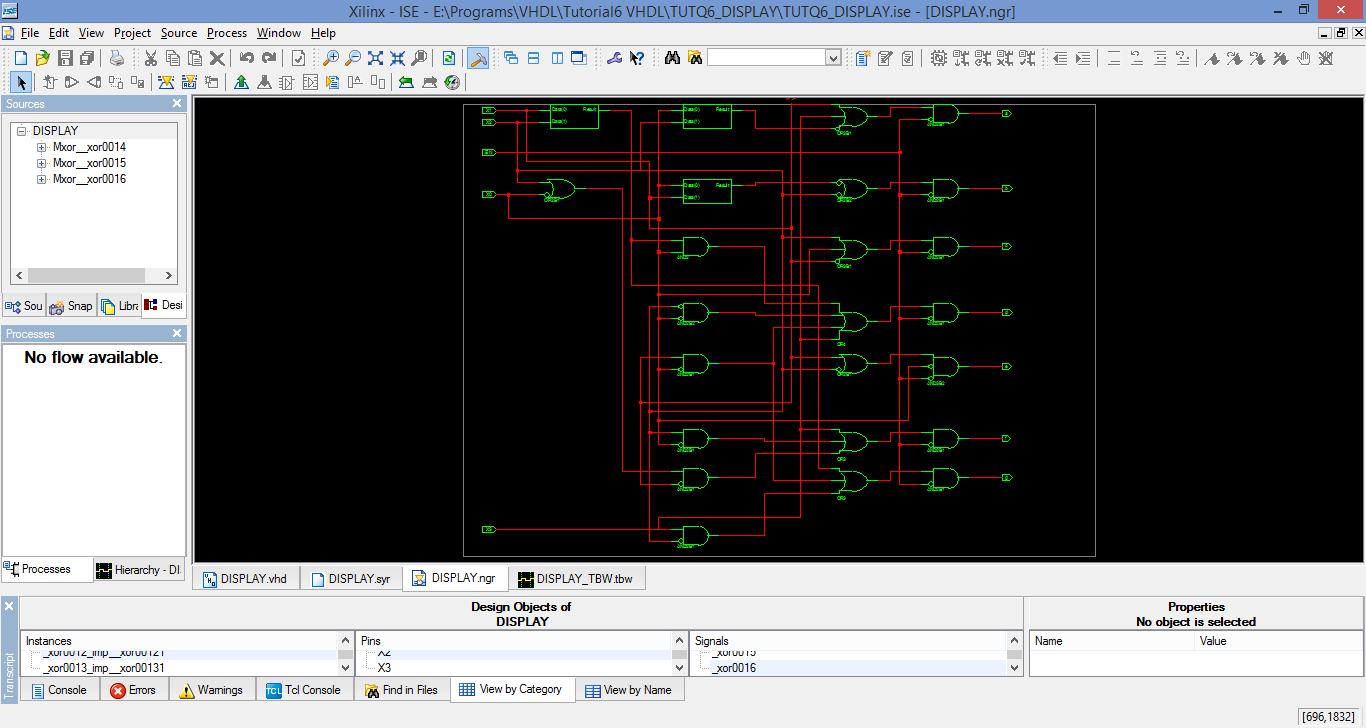
Total memory usage is 142148 kilobytes

Number of errors : 0 ( 0 filtered)

Number of warnings : 0 ( 0 filtered)

Number of infos : 0 ( 0 filtered)

### RTL Schematic:



## ALU:

### VHDL Code:

## library IEEE;

## use IEEE.STD\_LOGIC\_1164.ALL;

## entity ALU is

## port (SEL : in std\_logic\_vector(2 downto 0);

## Cin : in std\_logic;

## M : in std\_logic;

## A0, A1, A2, A3 : in std\_logic;

## B0, B1, B2, B3 : in std\_logic;

## OVR : out std\_logic;

## COUT : out std\_logic;

## F0, F1, F2, F3 : out std\_logic);

## end ALU;

## architecture Behavioral of ALU is

## signal c0, c1, c2, c3 : std\_logic;

## begin

## pmain : process(M, A0, A1, A2, A3, B0, B1, B2, B3, SEL, Cin)

## begin

## if SEL="000" and M='0' then

## if Cin='1' then

## F0 <= A0;

## F1 <= A1;

## F2 <= A2;

## F3 <= A3;

## OVR <= '0';

## COUT <= '0';

## elsif Cin='0' then

## F0 <= not A0;

## c0 <= A0;

## F1 <= A1 xnor A0;

## c1 <= A1 or A0;

## F2 <= A2 xnor (A1 or A0);

## c2 <= A2 or (A1 or A0);

## F3 <= A3 xnor (A2 or (A1 or A0));

## c3 <= A3 or (A2 or (A1 or A0));

## OVR <= (A3 or (A2 or (A1 or A0))) xor (A2 or (A1 or A0));

## COUT <= A3 or (A2 or (A1 or A0));

## end if;

## elsif SEL="000" and M='1' then

## F0 <= not A0;

## F1 <= not A1;

## F2 <= not A2;

## F3 <= not A3;

## COUT <= '0';

## OVR <= '0';

## elsif SEL="001" and M='0' then

## c0 <= (A0 and B0) or Cin;

## F0 <= (A0 and B0) xnor Cin;

## c1 <= (A1 and B1) or c0;

## F1 <= (A1 and B1) xnor c0;

## c2 <= (A2 and B2) or c1;

## F2 <= (A2 and B2) xnor c1;

## c3 <= (A3 and B3) or c2;

## F3 <= (A3 and B3) xnor c2;

## COUT <= c3;

## OVR <= '1';

## elsif SEL="001" and M='1' then

## F0 <= not (A0 and B0);

## F1 <= not (A1 and B1);

## F2 <= not (A2 and B2);

## F3 <= not (A3 and B3);

## COUT <= '0';

## OVR <= '0';

## elsif SEL="010" and M='0' then

## c0 <= (A0 and (not B0)) or Cin;

## F0 <= (A0 and (not B0)) xnor Cin;

## c1 <= (A1 and (not B1)) or c0;

## F1 <= (A1 and (not B1)) xnor c0;

## c2 <= (A2 and (not B2)) or c1;

## F2 <= (A2 and (not B2)) xnor c1;

## c3 <= (A3 and (not B3)) or c2;

## F3 <= (A3 and (not B3)) xnor c2;

## COUT <= c3;

## OVR <= c3 xor c2;

## elsif SEL="010" and M='1' then

## F0 <= not (A0 and (not B0));

## F1 <= not (A1 and (not B1));

## F2 <= not (A2 and (not B2));

## F3 <= not (A3 and (not B3));

## COUT <= '0';

## OVR <= '0';

## elsif SEL="011" and M='0' then

## if Cin = '0' then

## F0 <= '1';

## F1 <= '1';

## F2 <= '1';

## F3 <= '1';

## COUT <= '0';

## OVR <= '0';

## elsif Cin = '1' then

## F0 <= '0';

## F1 <= '0';

## F2 <= '0';

## F3 <= '0';

## COUT <= '1';

## OVR <= '1';

## end if;

## elsif SEL="011" and M='1' then

## F0 <= '1';

## F1 <= '1';

## F2 <= '1';

## F3 <= '1';

## COUT <= '0';

## OVR <= '0';

## elsif SEL="100" and M='0' then

## F0 <= (A0 xnor B0) xor Cin;

## c0 <= (Cin and (A0 xnor B0)) or (A0 and (not B0)) or A0;

## F1 <= (A1 xnor B1) xor c0;

## c1 <= (c0 and (A1 xnor B1)) or (A1 and (not B1)) or A0;

## F2 <= (A2 xnor B2) xor c1;

## c2 <= (c1 and (A2 xnor B2)) or (A2 and (not B2)) or A0;

## F3 <= (A3 xnor B3) xor c2;

## c3 <= (c2 and (A3 xnor B3)) or (A3 and (not B3)) or A0;

## COUT <= c3;

## OVR <= c3 xor c2;

## elsif SEL="100" and M='1' then

## F0 <= not (A0 or B0);

## F1 <= not (A1 or B1);

## F2 <= not (A2 or B2);

## F3 <= not (A3 or B3);

## COUT <= '0';

## OVR <= '0';

## elsif SEL="101" and M='0' then

## F0 <= (A0 or (not B0)) xor Cin;

## c0 <= (Cin and (A0 or (not B0))) or (A0 and B0);

## F1 <= (A1 or (not B1)) xor c0;

## c1 <= (c0 and (A1 or (not B1))) or (A1 and B1);

## F2 <= (A2 or (not B2)) xor c1;

## c2 <= (c1 and (A2 or (not B2))) or (A2 and B2);

## F3 <= (A3 or (not B3)) xor c2;

## c3 <= (c2 and (A3 or (not B3))) or (A3 and B3);

## COUT <= c3;

## OVR <= c3 xor c2;

## elsif SEL="101" and M='1' then

## F0 <= not B0;

## F1 <= not B1;

## F2 <= not B2;

## F3 <= not B3;

## COUT <= '0';

## OVR <= '0';

## elsif SEL="110" and M='1' then

## F0 <= A0 xor (not B0);

## F1 <= A1 xor (not B1);

## F2 <= A2 xor (not B2);

## F3 <= A3 xor (not B3);

## elsif SEL="110" and M='0' then

## F0 <= (A0 xor (not B0)) xor Cin;

## c0 <= (Cin and (A0 or (not B0))) or (A0 and B0);

## F1 <= (A1 or (not B1)) xor c0;

## c1 <= (c0 and (A1 or (not B1))) or (A1 and B1);

## F2 <= (A2 or (not B2)) xor c1;

## c2 <= (c1 and (A2 or (not B2))) or (A2 and B2);

## F3 <= (A3 or (not B3)) xor c2;

## c3 <= (c2 and (A3 or (not B3))) or (A3 and B3);

## COUT <= c3;

## OVR <= c3 xor c2;

## elsif SEL="111" and M='1' then

## F0 <= (A0 or (not B0));

## F1 <= A1 or (not B1);

## F2 <= A2 or (not B2);

## F3 <= A3 or (not B3);

## elsif SEL="111" and M='0' then

## F0 <= (A0 xor (not B0)) xor Cin;

## c0 <= (Cin and (A0 or (not B0))) or (A0 and B0);

## F1 <= (A1 or (not B1)) xor c0;

## c1 <= (c0 and (A1 or (not B1))) or (A1 and B1);

## F2 <= (A2 or (not B2)) xor c1;

## c2 <= (c1 and (A2 or (not B2))) or (A2 and B2);

## F3 <= (A3 or (not B3)) xor c2;

## c3 <= (c2 and (A3 or (not B3))) or (A3 and B3);

## COUT <= c3;

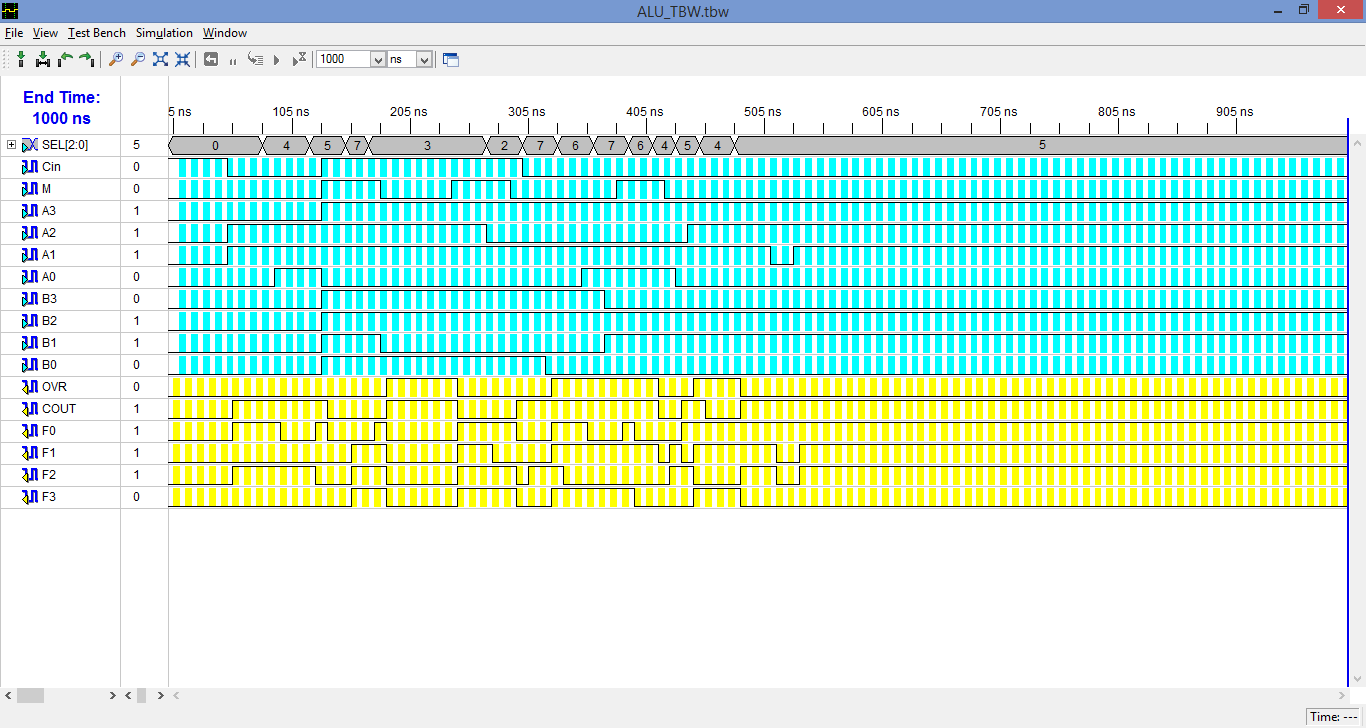
## OVR <= c3 xor c2;

## end if;

## end process;

## end Behavioral;

## Test Bench Simulation:



## Synthesis Report:

Release 8.2i - xst I.31

Copyright (c) 1995-2006 Xilinx, Inc. All rights reserved.

--> Parameter TMPDIR set to ./xst/projnav.tmp

CPU : 0.00 / 0.28 s | Elapsed : 0.00 / 0.00 s

--> Parameter xsthdpdir set to ./xst

CPU : 0.00 / 0.28 s | Elapsed : 0.00 / 0.00 s

--> Reading design: ALU.prj

TABLE OF CONTENTS

1) Synthesis Options Summary

2) HDL Compilation

3) Design Hierarchy Analysis

4) HDL Analysis

5) HDL Synthesis

5.1) HDL Synthesis Report

6) Advanced HDL Synthesis

6.1) Advanced HDL Synthesis Report

7) Low Level Synthesis

8) Partition Report

9) Final Report

=========================================================================

\* Synthesis Options Summary \*

=========================================================================

---- Source Parameters

Input File Name : "ALU.prj"

Input Format : mixed

Ignore Synthesis Constraint File : NO

---- Target Parameters

Output File Name : "ALU"

Output Format : NGC

Target Device : Automotive CoolRunner2

---- Source Options

Top Module Name : ALU

Automatic FSM Extraction : YES

FSM Encoding Algorithm : Auto

Mux Extraction : YES

Resource Sharing : YES

---- Target Options

Add IO Buffers : YES

MACRO Preserve : YES

XOR Preserve : YES

Equivalent register Removal : YES

---- General Options

Optimization Goal : Speed

Optimization Effort : 1

Keep Hierarchy : YES

RTL Output : Yes

Hierarchy Separator : /

Bus Delimiter : <>

Case Specifier : maintain

---- Other Options

lso : ALU.lso

verilog2001 : YES

safe\_implementation : No

Clock Enable : YES

wysiwyg : NO

=========================================================================

=========================================================================

\* HDL Compilation \*

=========================================================================

Compiling vhdl file "E:/Programs/VHDL/Tutorial6 VHDL/Q7\_ALU/ALU.vhd" in Library work.

Entity <alu> compiled.

Entity <alu> (Architecture <behavioral>) compiled.

=========================================================================

\* Design Hierarchy Analysis \*

=========================================================================

Analyzing hierarchy for entity <ALU> in library <work> (architecture <behavioral>).

Building hierarchy successfully finished.

=========================================================================

\* HDL Analysis \*

=========================================================================

Analyzing Entity <ALU> in library <work> (Architecture <behavioral>).

Entity <ALU> analyzed. Unit <ALU> generated.

=========================================================================

\* HDL Synthesis \*

=========================================================================

Performing bidirectional port resolution...

Synthesizing Unit <ALU>.

Related source file is "E:/Programs/VHDL/Tutorial6 VHDL/Q7\_ALU/ALU.vhd".

Found 1-bit xor2 for signal <$xor0021>.

Found 1-bit xor2 for signal <$xor0022>.

Found 1-bit xor2 for signal <$xor0023>.

Found 1-bit xor2 for signal <$xor0024> created at line 106.

Found 1-bit xor2 for signal <$xor0025> created at line 124.

Found 1-bit xor2 for signal <$xor0026> created at line 142.

Found 1-bit xor2 for signal <$xor0027> created at line 147.

Found 1-bit xor2 for signal <$xor0032>.

Found 1-bit xor2 for signal <$xor0033>.

Found 1-bit xor2 for signal <$xor0034>.

Found 1-bit xor2 for signal <$xor0035>.

Found 1-bit xor2 for signal <$xor0036> created at line 108.

Found 1-bit xor2 for signal <$xor0037> created at line 126.

Found 1-bit xor2 for signal <$xor0038> created at line 143.

Found 1-bit xor2 for signal <$xor0044>.

Found 1-bit xor2 for signal <$xor0045>.

Found 1-bit xor2 for signal <$xor0046>.

Found 1-bit xor2 for signal <$xor0047>.

Found 1-bit xor2 for signal <$xor0048> created at line 110.

Found 1-bit xor2 for signal <$xor0049> created at line 128.

Found 1-bit xor2 for signal <$xor0050> created at line 144.

Found 1-bit xor2 for signal <$xor0056>.

Found 1-bit xor2 for signal <$xor0057>.

Found 1-bit xor2 for signal <$xor0058>.

Found 1-bit xor2 for signal <$xor0059>.

Found 1-bit xor2 for signal <$xor0060> created at line 112.

Found 1-bit xor2 for signal <$xor0061> created at line 130.

Found 1-bit xor2 for signal <$xor0062> created at line 145.

Found 1-bit xor2 for signal <$xor0088> created at line 36.

Found 1-bit xor2 for signal <$xor0089> created at line 74.

Summary:

inferred 30 Xor(s).

Unit <ALU> synthesized.

=========================================================================

HDL Synthesis Report

Macro Statistics

# Latches : 10

1-bit latch : 10

# Xors : 30

1-bit xor2 : 30

=========================================================================

=========================================================================

\* Advanced HDL Synthesis \*

=========================================================================

=========================================================================

Advanced HDL Synthesis Report

Macro Statistics

# Latches : 10

1-bit latch : 10

=========================================================================

=========================================================================

\* Low Level Synthesis \*

=========================================================================

Optimizing unit <ALU> ...

WARNING:Xst:1294 - Latch <F3> is equivalent to a wire in block <ALU>.

WARNING:Xst:1294 - Latch <F2> is equivalent to a wire in block <ALU>.

WARNING:Xst:1294 - Latch <F1> is equivalent to a wire in block <ALU>.

WARNING:Xst:1294 - Latch <F0> is equivalent to a wire in block <ALU>.

WARNING:Xst:1294 - Latch <F3> is equivalent to a wire in block <ALU>.

WARNING:Xst:1294 - Latch <F2> is equivalent to a wire in block <ALU>.

WARNING:Xst:1294 - Latch <F1> is equivalent to a wire in block <ALU>.

WARNING:Xst:1294 - Latch <F0> is equivalent to a wire in block <ALU>.

=========================================================================

\* Partition Report \*

=========================================================================

Partition Implementation Status

-------------------------------

No Partitions were found in this design.

-------------------------------

=========================================================================

\* Final Report \*

=========================================================================

Final Results

RTL Top Level Output File Name : ALU.ngr

Top Level Output File Name : ALU

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : YES

Target Technology : Automotive CoolRunner2

Macro Preserve : YES

XOR Preserve : YES

Clock Enable : YES

wysiwyg : NO

Design Statistics

# IOs : 19

Cell Usage :

# BELS : 460

# AND2 : 155

# AND3 : 20

# AND4 : 4

# INV : 163

# OR2 : 62

# OR3 : 24

# OR4 : 1

# XOR2 : 31

# FlipFlops/Latches : 6

# LD : 6

# IO Buffers : 19

# IBUF : 13

# OBUF : 6

=========================================================================

CPU : 6.78 / 7.06 s | Elapsed : 7.00 / 7.00 s

-->

Total memory usage is 143236 kilobytes

Number of errors : 0 ( 0 filtered)

Number of warnings : 8 ( 0 filtered)

Number of infos : 0 ( 0 filtered)

### RTL Schematic:

