Department of Electrical and Computer Engineering

Digital Systems Tutorial and Exercise

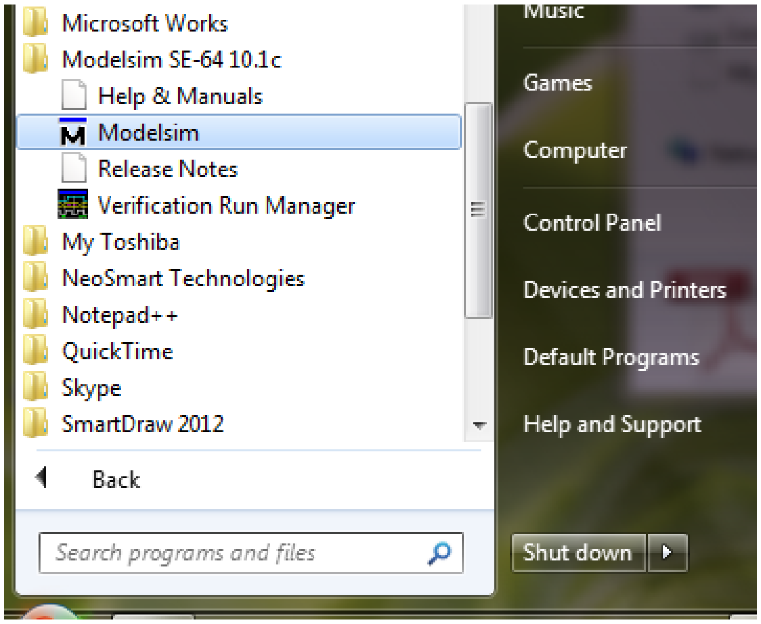
Tutorial: Modelsim – Introduction to VHDL Design Simulation and Environment

1. Start Modelsim Application

1.1 Create a new folder, for instance, named modelsim_projects, for your future digital design simulation projects.

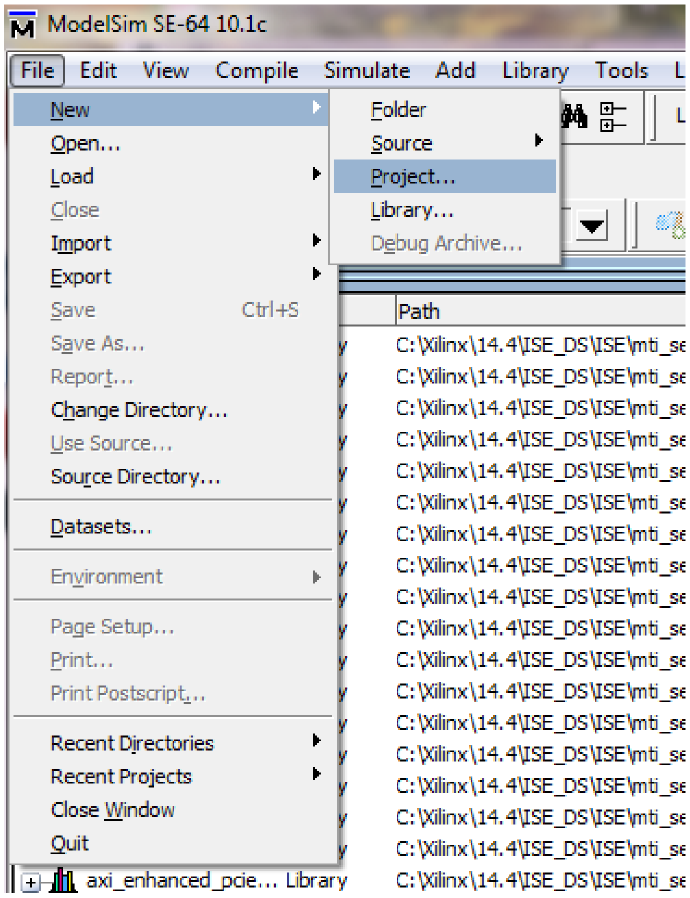
1.2 Start Modelsim

Go to All Programs > Modelsim SE-64 10.1c and launch Modelsim as shown below

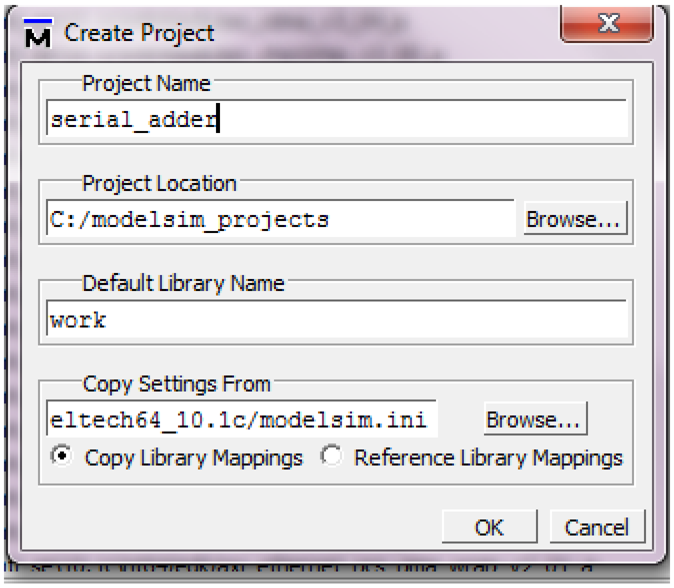


2. New Project

2.1 Start a new project: File > New > Project…



2.2 Project Name and Location



Project Name: serial_adder

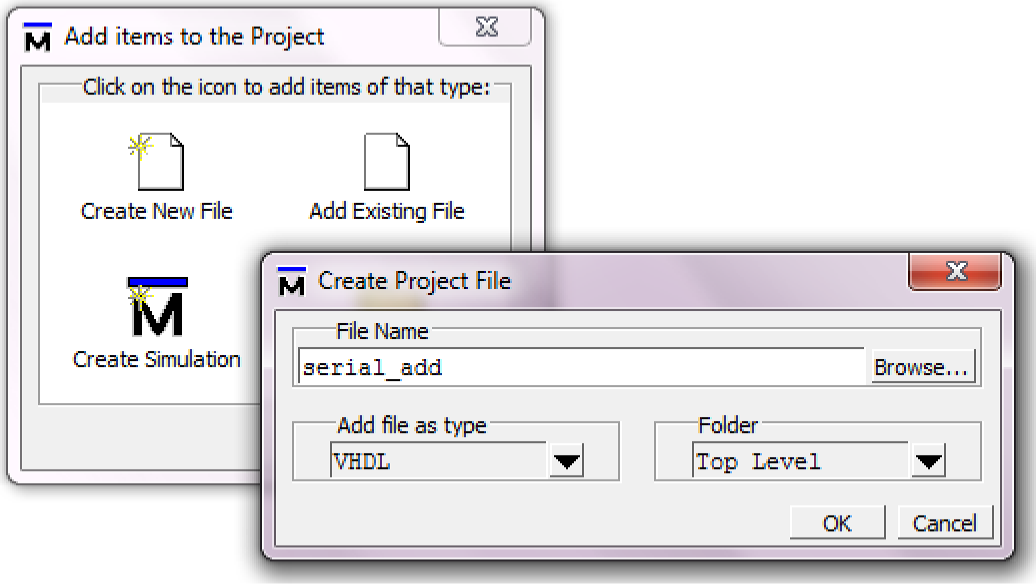
Project Location: C:/modelsim_projects

Default Library Name: work

Copy Settings From: eltech64_10.1c/modelsim.ini

[✓ Copy Library Mappings] Reference Library Mappings

2.3 Create New File

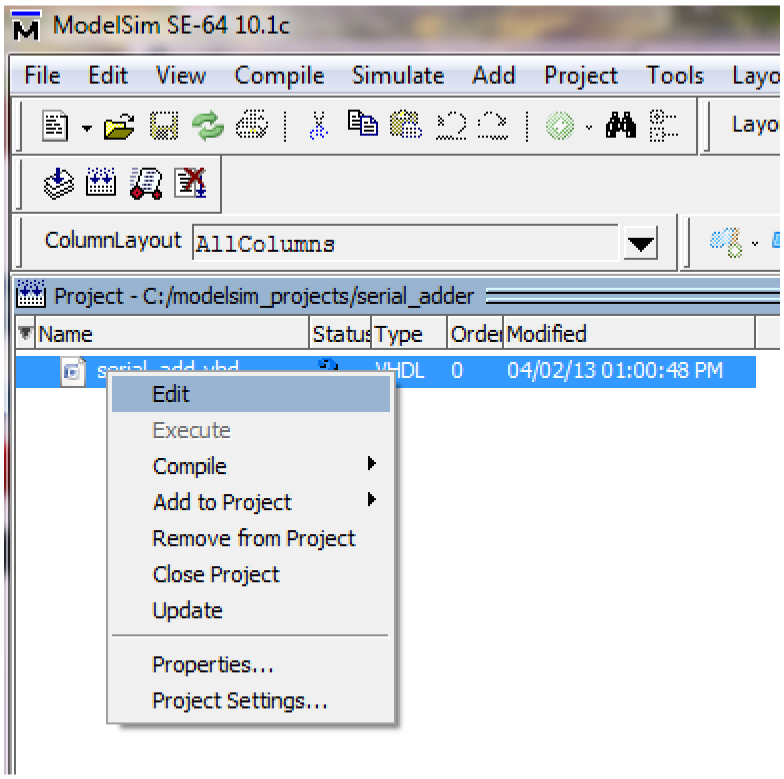


File Name: serial_add

Add file as type: VHDL

Folder: Top Level

2.3 Start edit session by right clicking on serial_add.vhd and select edit



ColumnLayout: AllColumns

Project - C:/modelsim_projects/serial_adder

Name	Status	Type	Order	Modified
serial_add.vhd	?	VHDL	0	04/02/13 01:00:48 PM

2.4 Tutorial VHDL Design

This modelsim project simulates a serial-input adder. Table 1 describes the ports.

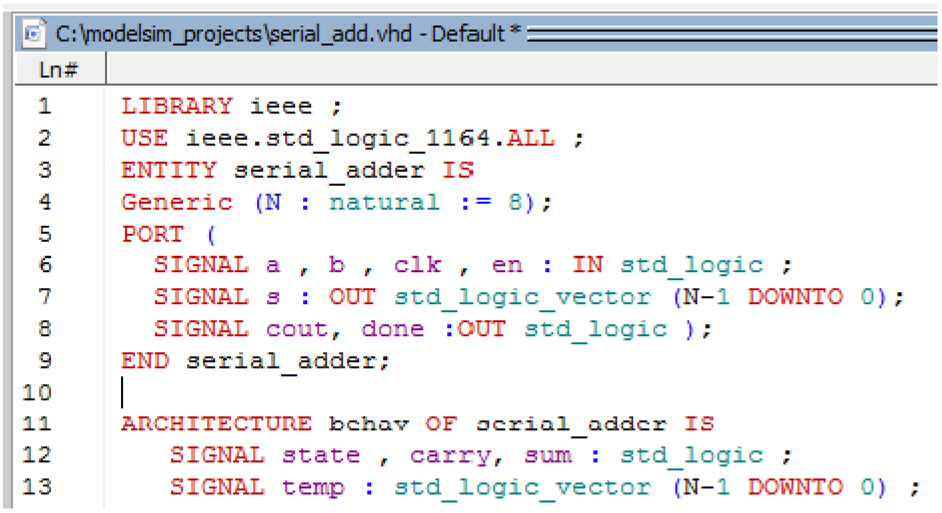
Table 1. Serial-Adder Ports

Port Name	Type	Length	Description
a	Input	1	input operand 1
b	Input	1	input operand 2
s	Output	N	sum
cout	Output	1	carry out
Clk	Input	1	system clock
En	Input	1	device enable
done	Output	1	done flag

Description: The device adds two n-bit unsigned vectors. After a reset (enable goes low one clock cycle and then high), apply the operand bits starting from the least significant bits at each clock cycles. When done is high, the result vector output at s and cout.

3. Edit VHDL Code

Edit and save the serial adder code below



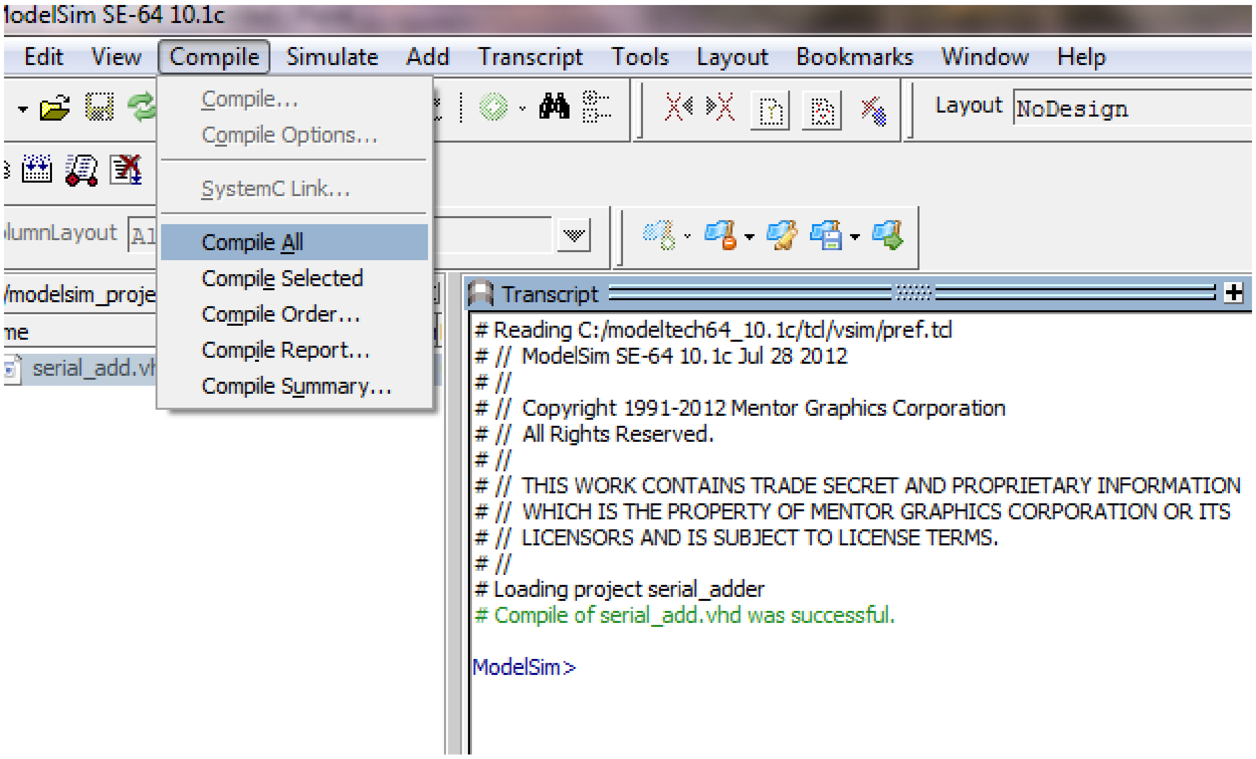
C:\modelsim_projects\serial_add.vhd - Default *

```
LIBRARY ieee ;
USE ieee.std_logic_1164.ALL ;
ENTITY serial_adder IS
Generic (N : natural := 8);
PORT  (
  SIGNAL a , b , clk , en : IN std_logic ;
  SIGNAL s : OUT std_logic_vector (N-1 DOWNTO 0);
  SIGNAL cout, done :OUT std_logic );
END serial_adder;

ARCHITECTURE behav OF serial_adder IS
   SIGNAL state , carry, sum : std_logic ;
   SIGNAL temp : std_logic_vector (N-1 DOWNTO 0) ;
```

```
BEGIN
trans_and_count :PROCESS ( clk , en )
   VARIABLE counter : INTEGER := 0;
   BEGIN
        IF (en = '0') THEN  -- reset.
            state <= '0';
            counter := 0;
            temp <= (others => '0');
            done <= '0';
        ELSIF clk = '1' and clk'event THEN
            IF (counter < N) THEN -- move to next bit
                done <= '0';
                state <= carry;
                counter := counter + 1;
                temp(N-1) <= sum;
                FOR i IN N-2 DOWNTO 0 LOOP
                    temp(i) <= temp(i+1);
                END LOOP;
            ELSE                -- the addition is done.
                done <= '1';
            END IF;
        END IF ;
   END PROCESS trans_and_count;
   -- wire state (storage for carry synch with ck) to Port cout
   cout <= state;

    output : PROCESS ( a , b , state )
    BEGIN
         sum <= a XOR b XOR state;
         carry <= (a AND b) OR (a AND state) OR (b AND state);
    END PROCESS output;


    s <= temp; --wire to output

END behav;
```

4. Compile

On Compile pull-down menu select Compile All. Error or successful message displays in Transcript window.

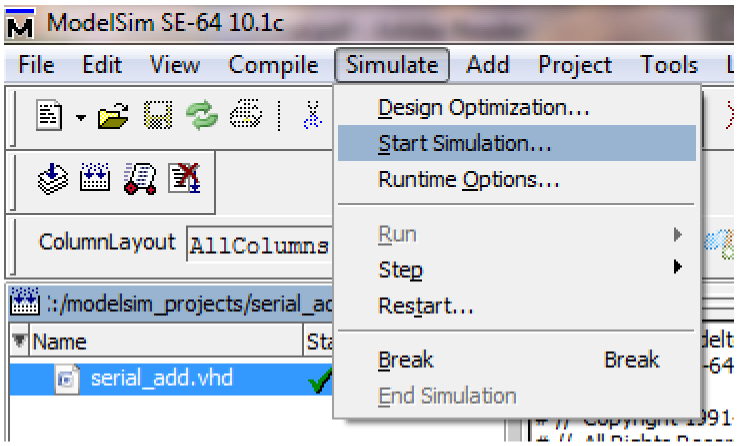


```
# Reading C:/modeltech64_10.1c/tcl/vsim/pref.tcl
# // ModelSim SE-64 10.1c Jul 28 2012
# //
# // Copyright 1991-2012 Mentor Graphics Corporation
# // All Rights Reserved.
# //
# // THIS WORK CONTAINS TRADE SECRET AND PROPRIETARY INFORMATION
# // WHICH IS THE PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS
# // LICENSORS AND IS SUBJECT TO LICENSE TERMS.
# //
# Loading project serial_adder
# Compile of serial_add.vhd was successful.

ModelSim>
```

5. Simulation

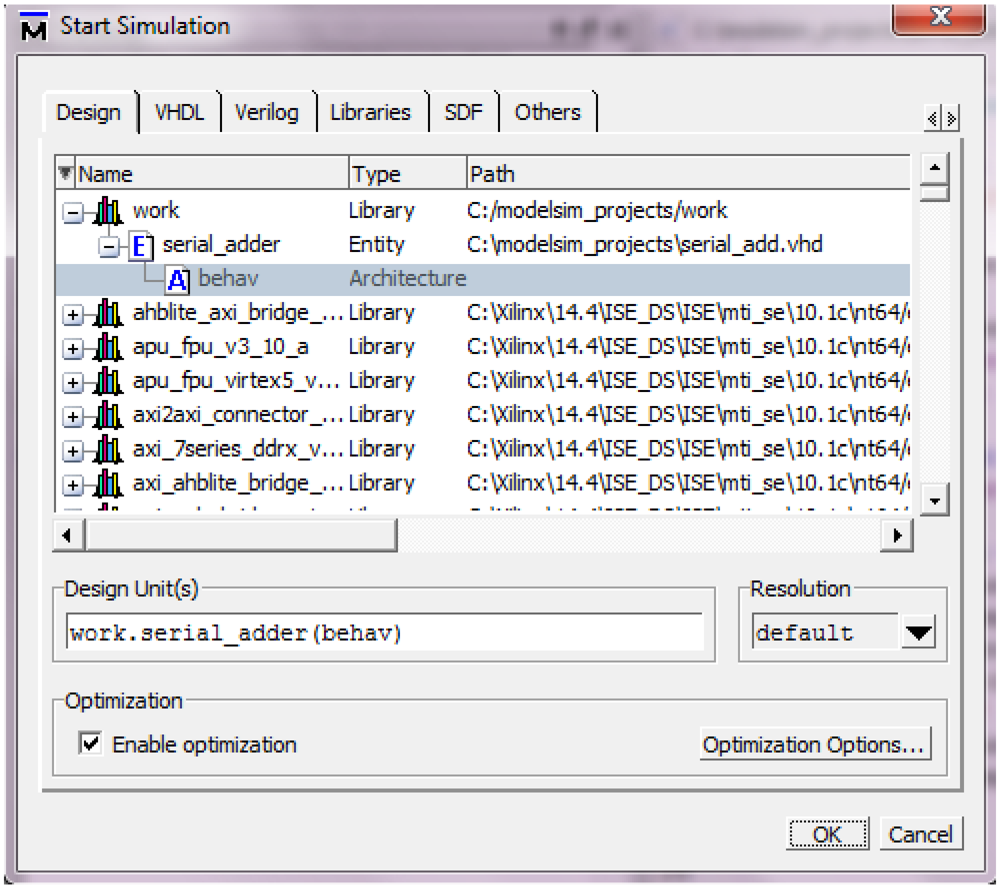
5.1 Start Simulation

On Simulate pull-down menu select Start Simulation...



5.2 Select Design Unit to Simulate

Expand work library (containing compiled vhdl codes) and select behave architecture.



Name	Type	Path
work	Library	C:/modelsim_projects/work
serial_adder	Entity	C:\modelsim_projects\serial_add.vhd
behav	Architecture	
ahblite_axi_bridge_...	Library	C:\Xilinx\14.4\ISE_DS\ISE\mti_se\10.1c\nt64/
apu_fpu_v3_10_a	Library	C:\Xilinx\14.4\ISE_DS\ISE\mti_se\10.1c\nt64/
apu_fpu_virtex5_v...	Library	C:\Xilinx\14.4\ISE_DS\ISE\mti_se\10.1c\nt64/
axi2axi_connector_...	Library	C:\Xilinx\14.4\ISE_DS\ISE\mti_se\10.1c\nt64/
axi_7series_ddrx_v...	Library	C:\Xilinx\14.4\ISE_DS\ISE\mti_se\10.1c\nt64/
axi_ahblite_bridge_...	Library	C:\Xilinx\14.4\ISE_DS\ISE\mti_se\10.1c\nt64/

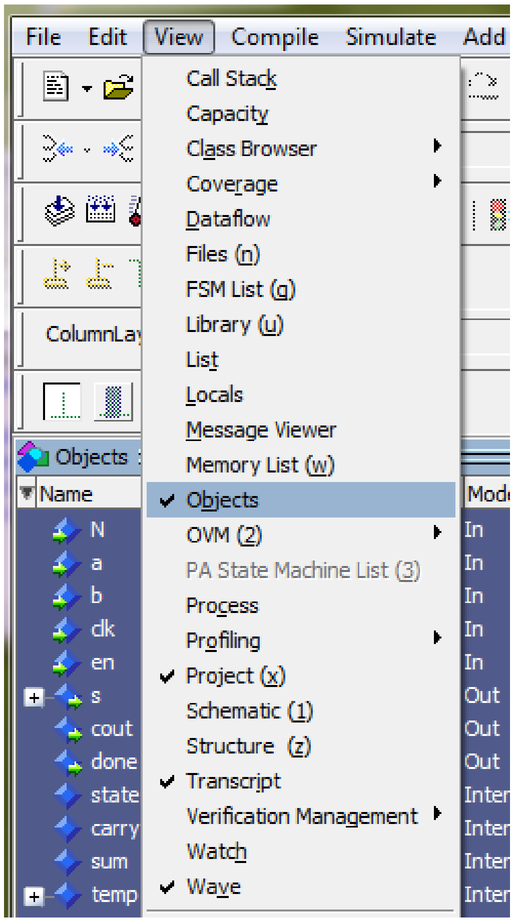
Design Unit(s): `work.serial_adder(behav)`

Resolution: `default`

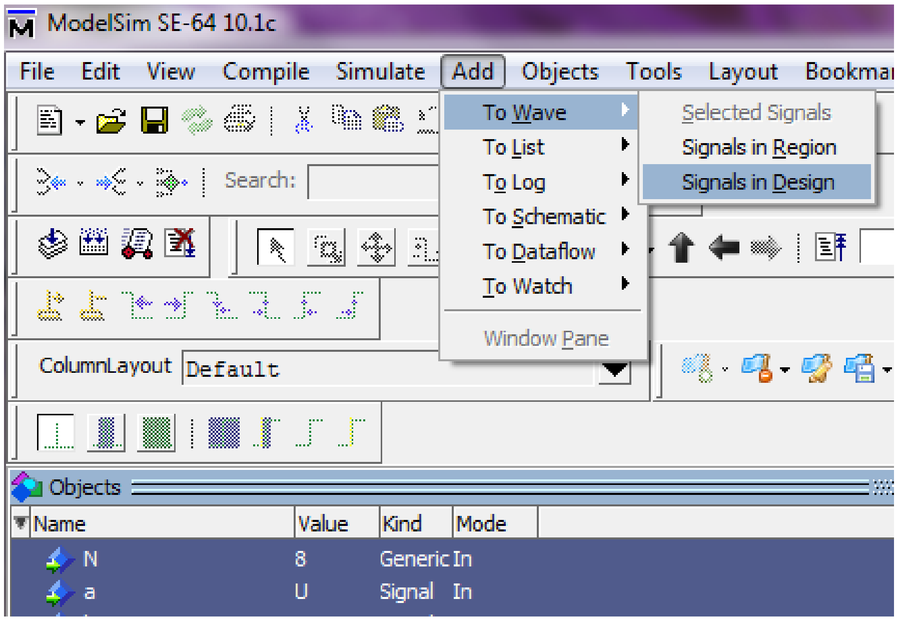
Optimization

[✓] Enable optimization

5.3 On View check Objects, Project, Transcript and Wave



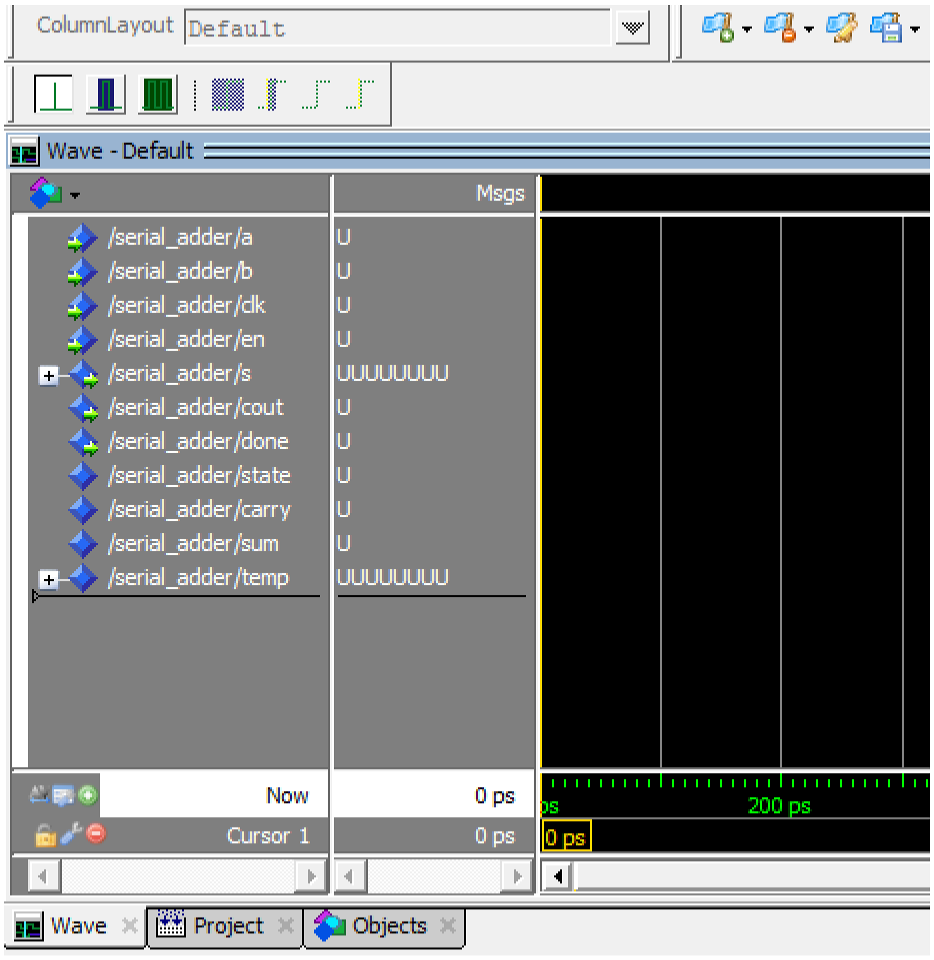
5.4 In Objects window, on Add pull-down menu select Add to wave > Signals in Design



Name	Value	Kind	Mode
N	8	Generic	In
a	U	Signal	In

5.5 Wave window

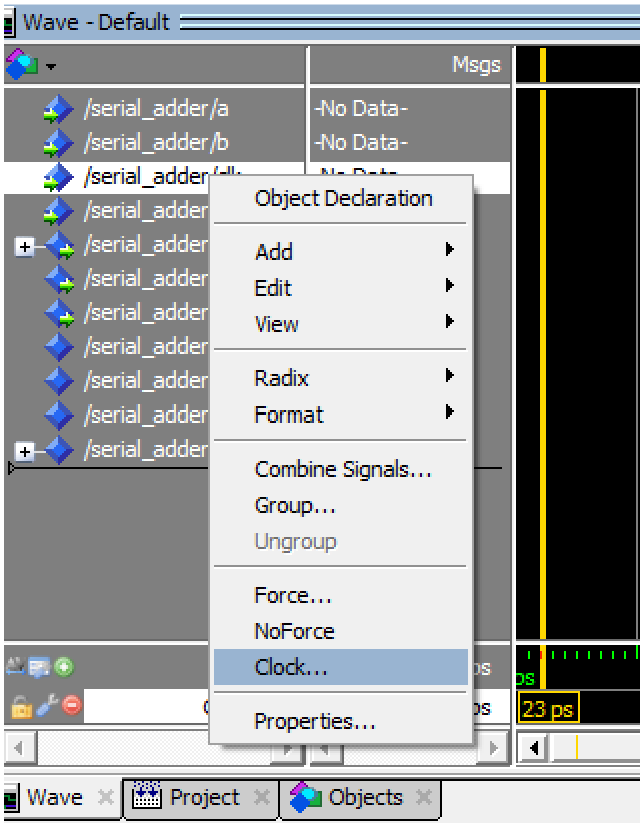
Select wave window



```
/serial_adder/a        U
/serial_adder/b        U
/serial_adder/clk      U
/serial_adder/en       U
/serial_adder/s        UUUUUUUU
/serial_adder/cout     U
/serial_adder/done     U
/serial_adder/state    U
/serial_adder/carry    U
/serial_adder/sum      U
/serial_adder/temp     UUUUUUUU

Now                    0 ps
Cursor 1               0 ps
```

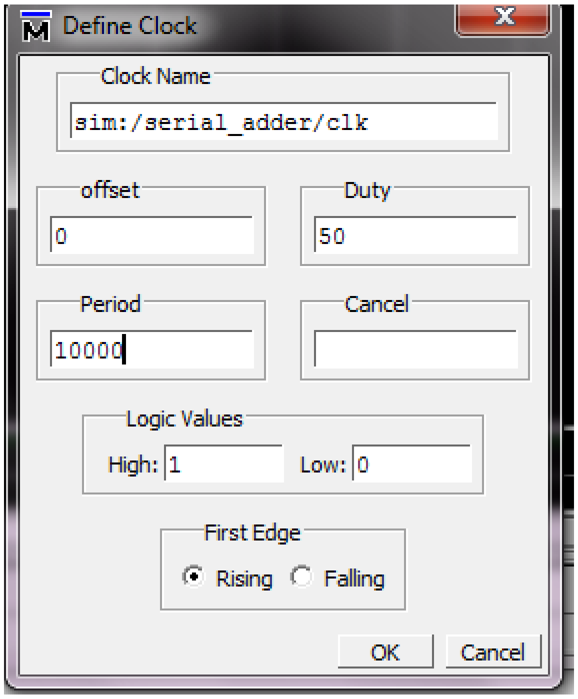
5.5 Force Clock Signal

Right click on /serial_adder/clk and select Clock…



5.6 Clock Parameters

The clock period is 10000 ps = 10 ns which will be the period for simulation step.


Define Clock

Clock Name: sim:/serial_adder/clk

offset: 0

Duty: 50

Period: 10000

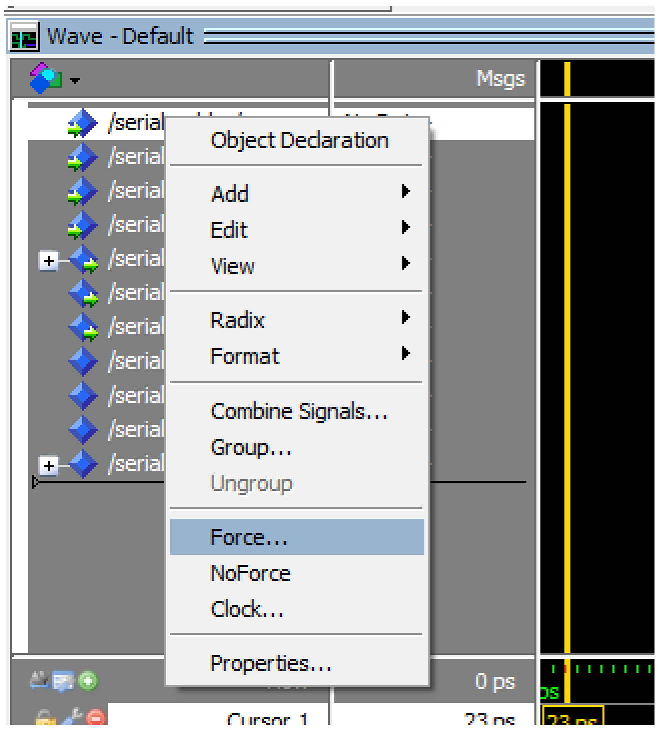
Cancel:

Logic Values — High: 1 Low: 0

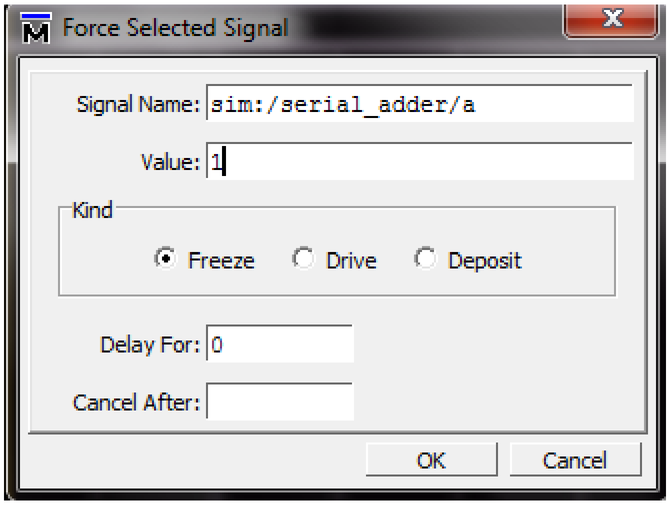
First Edge: [✓ Rising] Falling

5.7 Forcing Signal Values

Right click on signal /serial_adder/a and select Force…



Assign a one to signal a


Force Selected Signal

Signal Name: sim:/serial_adder/a

Value: 1

Kind: (● Freeze) ○ Drive ○ Deposit

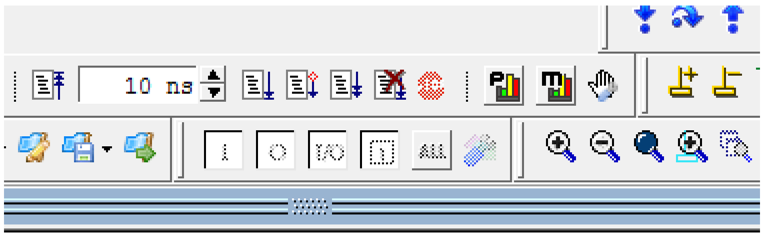
Delay For: 0

Cancel After:

Continue forcing b to a zero and en to a zero.

5.8 Single-Step Run

- Set the run period to 10 ns and click on the run icon. It is the first icon to the right of the 10 ns display
- Adjust the zoom scale to correctly view the waves
- After running (10 ns) one period of the clock appears

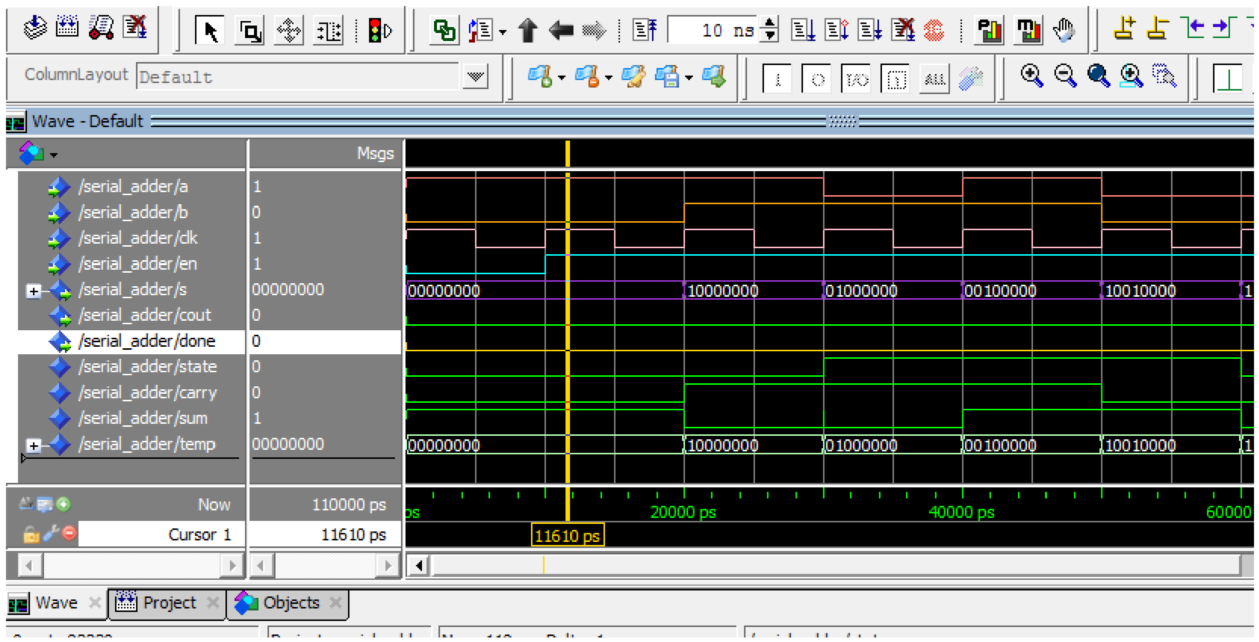


5.9 Complete a Simulation

- Force en to a one and click run. After this clock cycle the adder will add the least significant bit of the two 8-bit vectors a and b, (bit-0 of a and b are a(0) and b(0)).
- Let the test vectors a and b be
  a = “00001011”
  b = “00001110”
  ----
  s = “00011001” is the expected result.
- Continue to force a to ‘1’ and b to ‘1’ (forcing operands bit-1) then run
  the result a(0) plus b(0) plus carry appears as temp(0) or s(0) and
  the result a(1) plus b(1) plus carry will appear in s(0) on the next clock cycle
- Continue to force a to ‘0’ and b to ‘1’ (forcing bit-2) then run
  a(1) plus b(1) plus carry appears as s(0) and
  the result a(2) plus b(2) plus carry will appear in s(0) on the next clock cycle

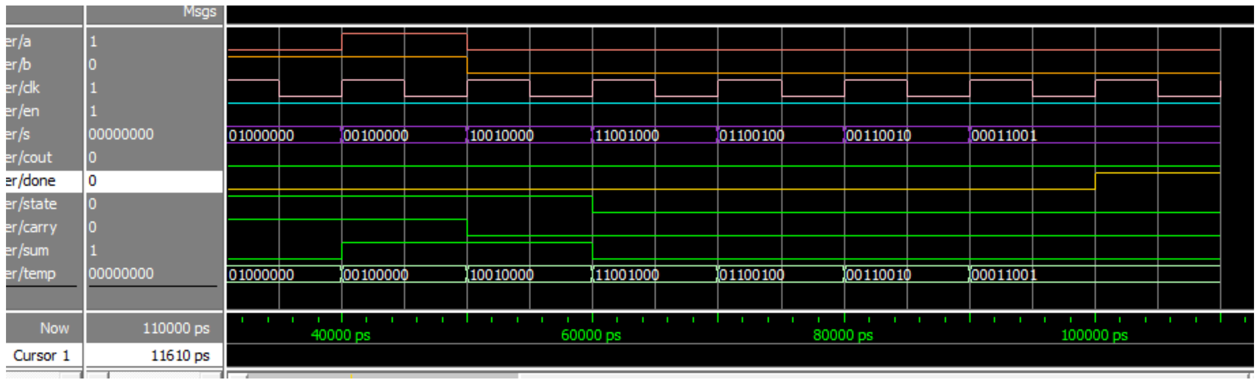
- Continue to force a to ‘1’ and b to ‘1’ (forcing bit-3) then run
  a(2) plus b(2) plus carry appears as s(0) and
  a(3) plus b(3) plus carry will appear in s(0) on the next clock cycle
- Continue to force a to ‘0’ and b to ‘0’ (forcing bit-4) then run
  a(3) plus b(3) plus carry appears as s(0) and
  a(4) plus b(4) plus carry will appear in s(0) on the next clock cycle
- Continue to apply 0 to a and b three more times(by hitting the run button)
- One more run the done signal goes high
- To apply another set of test vectors apply en low and then high and repeat the process

Observe the waves for correctness (see below). Waves can have different colors, right click > properties > wave colors, e.g., signal s below is a dark orchid.



	Msgs
/serial_adder/a	1
/serial_adder/b	0
/serial_adder/clk	1
/serial_adder/en	1
/serial_adder/s	00000000
/serial_adder/cout	0
/serial_adder/done	0
/serial_adder/state	0
/serial_adder/carry	0
/serial_adder/sum	1
/serial_adder/temp	00000000
Now	110000 ps
Cursor 1	11610 ps

```
s:    00000000  10000000  01000000  00100000  10010000
temp: 00000000  10000000  01000000  00100000  10010000
ps  20000 ps  40000 ps  60000
11610 ps
```



	Msgs
er/a	1
er/b	0
er/clk	1
er/en	1
er/s	00000000
er/cout	0
er/done	0
er/state	0
er/carry	0
er/sum	1
er/temp	00000000
Now	110000 ps
Cursor 1	11610 ps

```
s:    01000000  00100000  10010000  11001000  01100100  00110010  00011001
temp: 01000000  00100000  10010000  11001000  01100100  00110010  00011001
40000 ps  60000 ps  80000 ps  100000 ps
```

5.10 Exercise

Apply test vectors to verify the correctness for signal cout.