

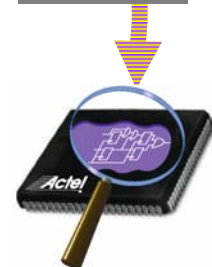
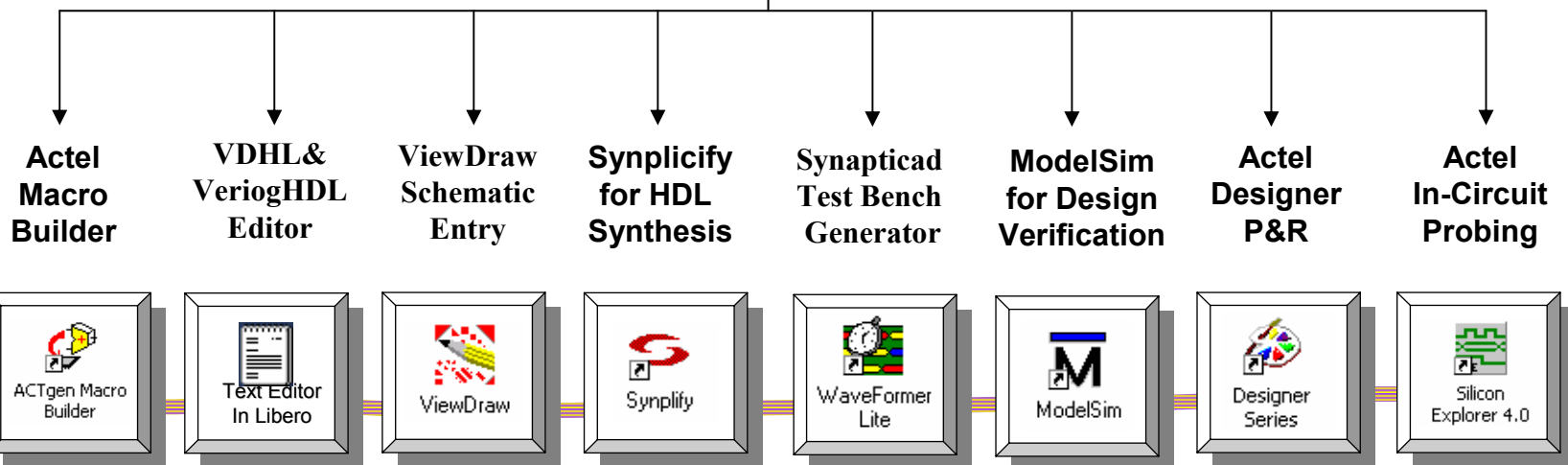


Libero Overview and Design Flow



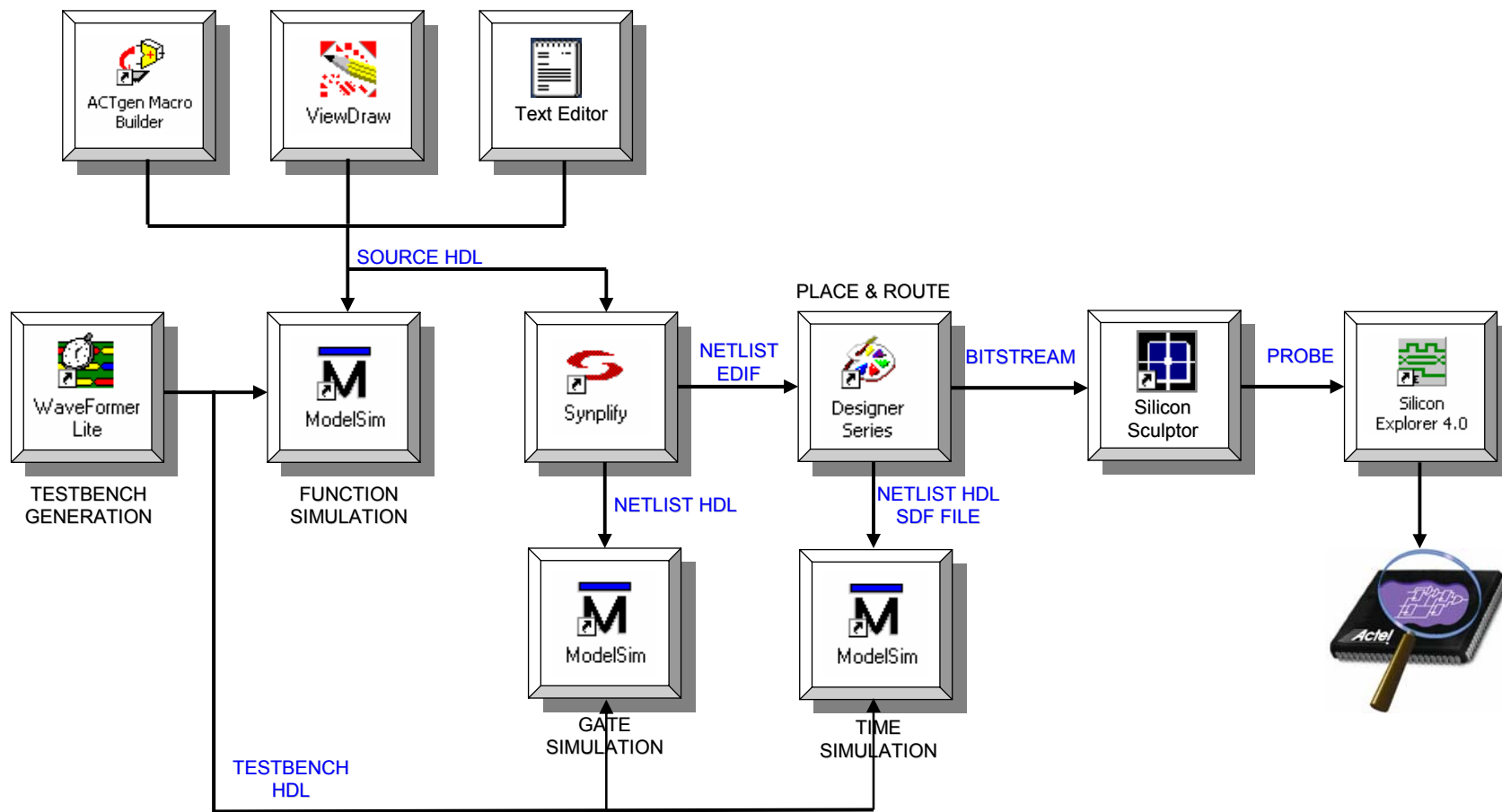


Libero Integrated Orchestra



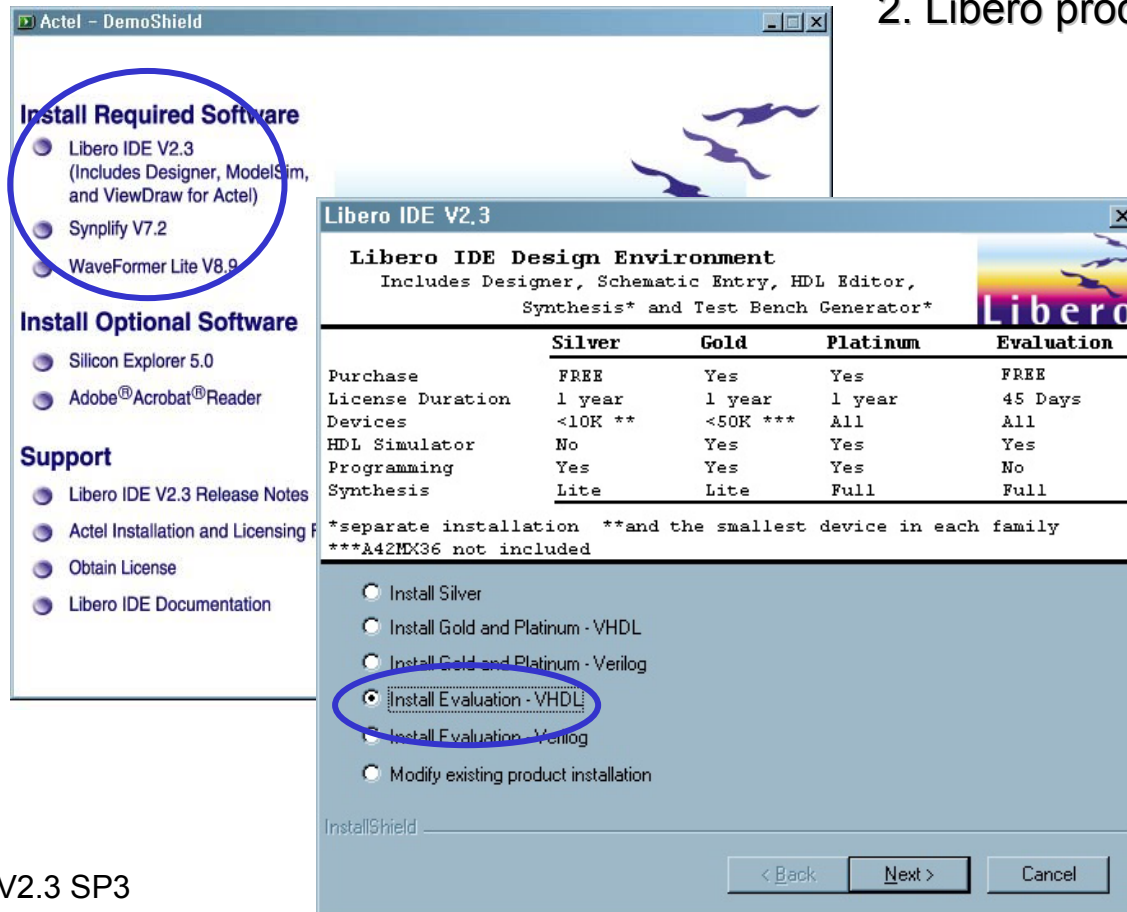


Libero Design Flow



Software Install

1. Libero, Synplify, Waveform install
2. Libero product : evaluation install





Libero Software Feature

	Function	Tool	Vendor	Libero IDE				Designer Software		
				Silver	Gold	Platinum	Eval	Gold	Platinum	Eval
Device Support	Maximum 10k Gate Devices*	All Actel FPGAs	Actel	✓	✓	✓	✓	✓	✓	✓
	Maximum 50k Gate Devices*	All Actel FPGAs	Actel		✓	✓	✓	✓	✓	✓
	All Actel Devices	All Actel FPGAs	Actel			✓	✓		✓	✓
	Synthesis	Synplify Lite for Actel/ Synplify for Actel	Synplicity	✓	✓	✓	✓			
	Simulation	ModelSim for Actel	Mentor Graphics		✓	✓	✓			
	Test Bench Generator	WaveFormer Lite	SynaptiCAD	✓	✓	✓	✓			
	Program File	Designer Software	Actel	✓	✓	✓		✓	✓	

License Request

1. <http://register.actel.com/RegSerial.asp> 접속
2. Evaluation 신청시 [here](#) click
3. Language and products 선택(Libero Evaluation)
4. C:\vol 과 synplify 의 hostid 입력
=> Synplify hostid는 설치후 synplify를 실행하면 얻을수 있음.

register



Designer and Libero Registration and Licensing

2
Click [here](#) to obtain a license for Designer and Libero

Please enter your Software ID number (new purchases and returning users)

The Software ID number is located on the purchased product on this site you license file.

[Licensing FAQs for Designer and Libero](#)

Free Products

Click on the product you want to register:
[Licensing FAQs for Designer and Libero](#)



Designer Evaluation

Try out Designer Platinum free for 45 days. You can receive one free evaluation license, which gives access to all the features and devices available in Actel's Designer, except for the ability to generate a programming file. Select one license type:

☐ PC Node-Locked ☐ PC Floating ☐ WS Floating



Designer Silver

A FREE 1 year license for the which supports all Actel devices available on PC platforms only



Libero Evaluation

3
Try out Libero Platinum free for evaluation license, which gives available in Actel's Libero, exc programming file. This product Select one HDL:

☐ Verilog ☒ VHDL



Libero Silver

A FREE 1 year license for the supports all Actel devices und on PC platforms only and is no

5. 이후 customer 정보 입력후 mail로 license 받음.

Libero Platinum Eval VHDL PC Node-locked Licensing

Enter your PC's hard-disk C drive Volume Serial Number: *

The Volume Serial Number is an 8 character hexadecimal number of the form xxxx-xxxx. To obtain your Volume Serial Number type the following at a DOS or Command Prompt:
(Note: You must use the C drive hexadecimal number)
C:> Vol C:

Enter your Synplify Host ID: *

To obtain the Synplify HostID you must first install Synplify from your Libero CD. Then select Start->Synplify->Synplify. A dialog box will display a message that includes the line
Example Your hostid is: 79C88961

Enter your hostID in the box provided above, then click the Cancel button in the dialog box to exit Synplify.

Please Specify platform: * ☐ Win NT4 ☐ Win 98 ☒ Win 2000

* = Required
[Licensing FAQs for Designer and Libero](#)

[Start over](#)





License Setup

1. Email로 받은 license.dat를 actel install directory에 copy

2. Os가 win2000일 경우

•내컴퓨터 등록정보->고급->환경변수

변수 LM_LICENSE_FILE

값 C:\<INSTALL DIR>\license.dat

변수 SYNPLICITY_LICENSE_FILE

값 C:\<INSTALL DIR>\license.dat

•위 두항목을 추가.

3. Os가 win98일 경우

•Autoexec.bat 파일에

SET LM_LICENSE_FILE = C:\<라이센스 위치>\license.dat

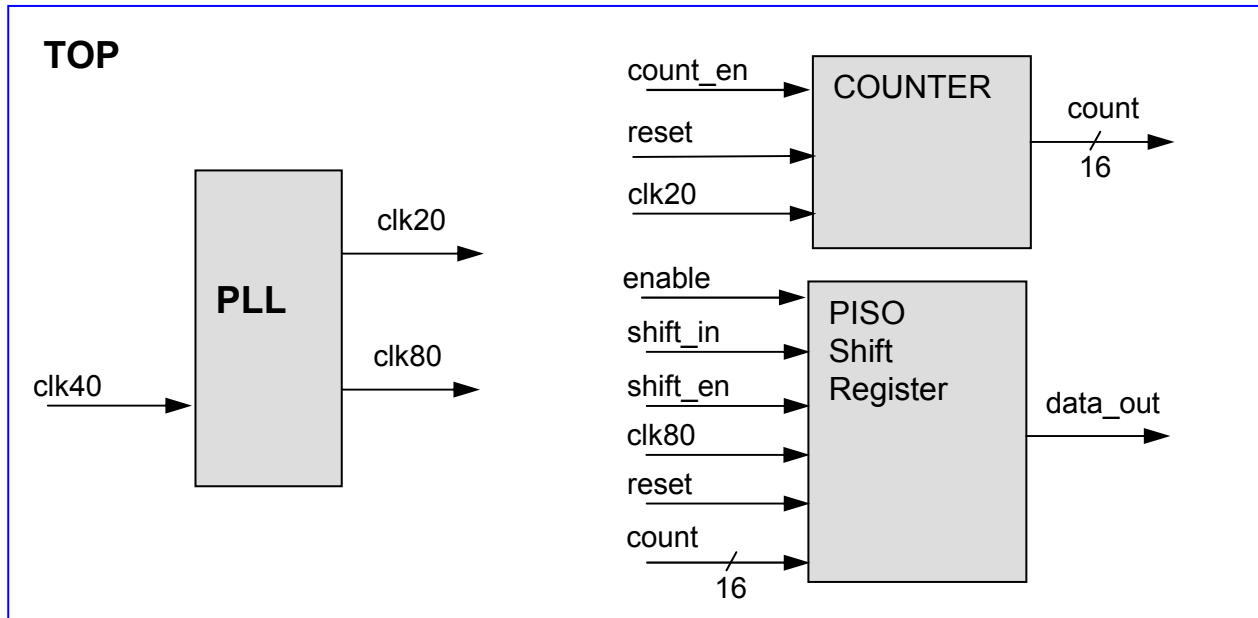
SET SYNPLICITY_LICENSE_FILE = C:\<라이센스 위치>\license.dat

SET SYNCAD_HOME = C:\<WaveFormer Install DIR>\synapticad

•위 세항목을 추가.



Demo Design Block Diagram



1. Pll_block : 40Mhz input을 20Mhz, 80Mhz 의 clock 생성
2. Count_block : 20Mhz의 clock으로 16bit counter
3. Piso_block : counter 의 output을 1bit serial로 출력
4. Top : 세개의 sub_block을 통합



Libero Project Manager

Design Explorer Window

- 디자인의 계층구조 표시:
 - Design hierarchy: source design에 대한 계층구조 표시
 - File Manager : design의 구성및 결과 file 관리창

HDL Editor Window

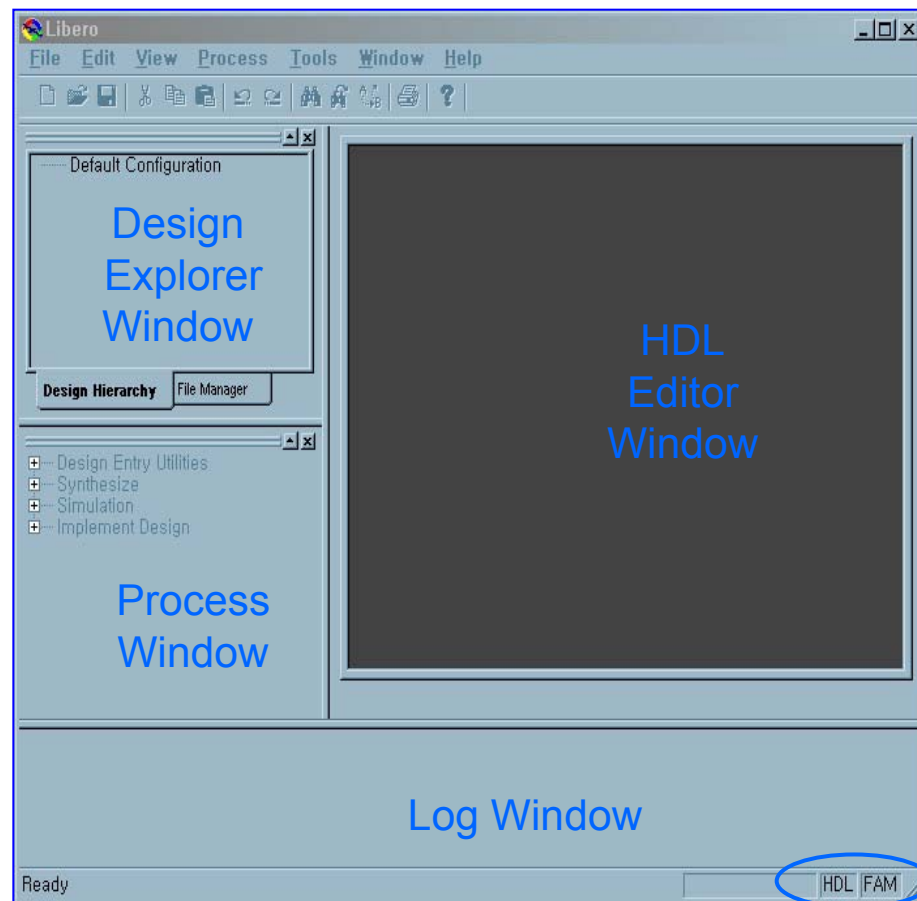
- Verilog와 VHDL93 text editor

Process Window

- Design을 위한 다른 tools로 연결

Log Window

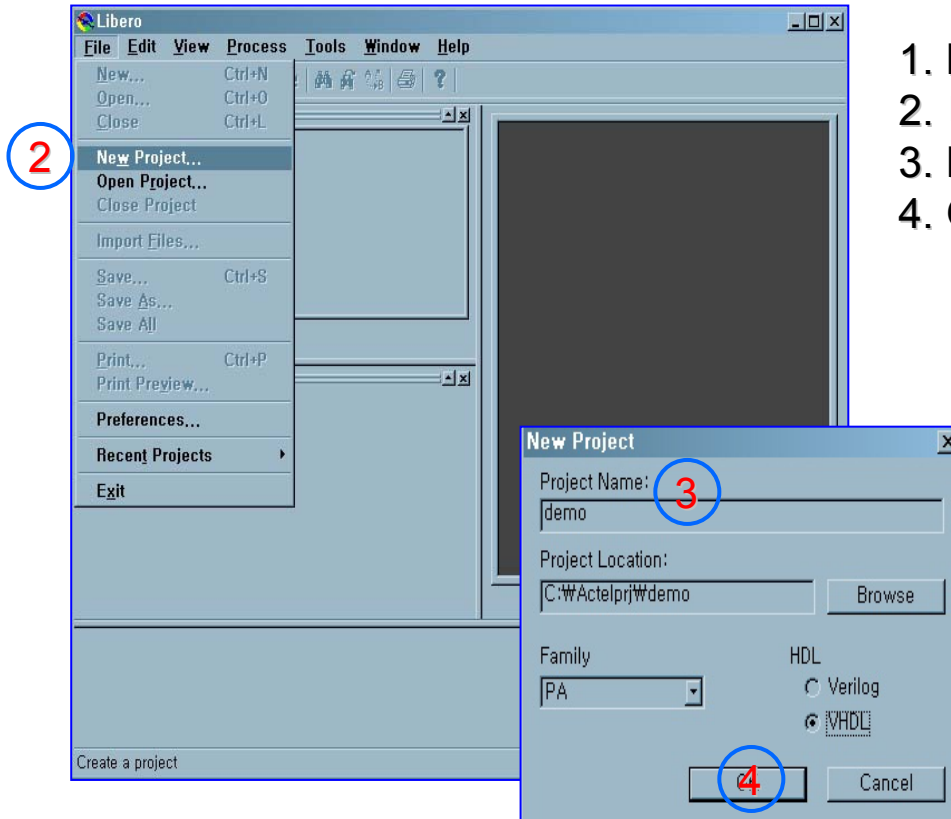
- Software 연결상태및 메시지창



Language and Family



Create Project

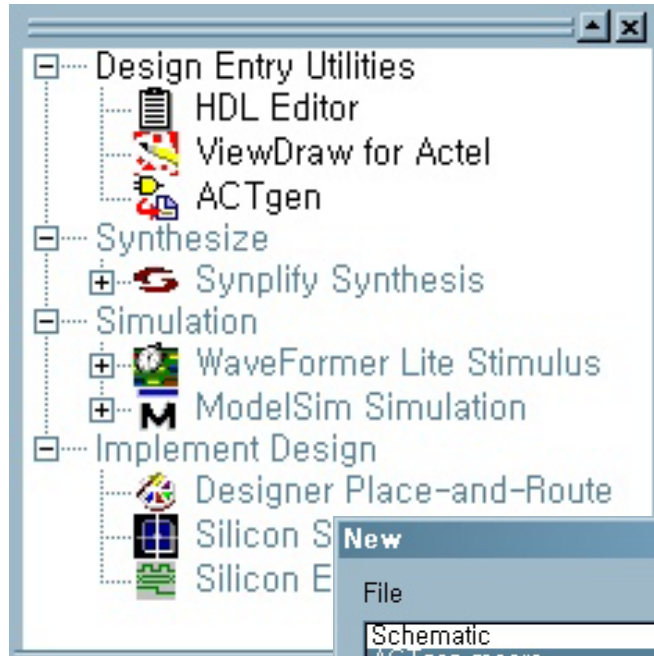


1. Libero IDE V2.3을 실행
2. 메뉴에서 File -> New Project 실행
3. Project Name, Location, Family, HDL 지정
4. OK.

Note : VHDL 과 Verilog와의 mixed design은 지원하지 않음.
한 개의 HDL과 Schematic은 가능



Process Window



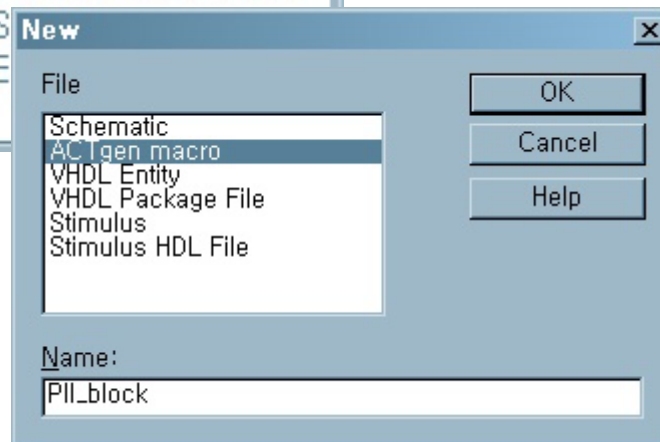
1. Design 방법 및 지원되는 S/W를 보여줌.
2. Design 단계 별로 사용 가능한지를 글씨체로 확인 가능하며 다음단계로의 이동을 쉽게 할 수 있음.

Design Entry Utilities

HDL Editor : VHDL & Verilog editor 생성

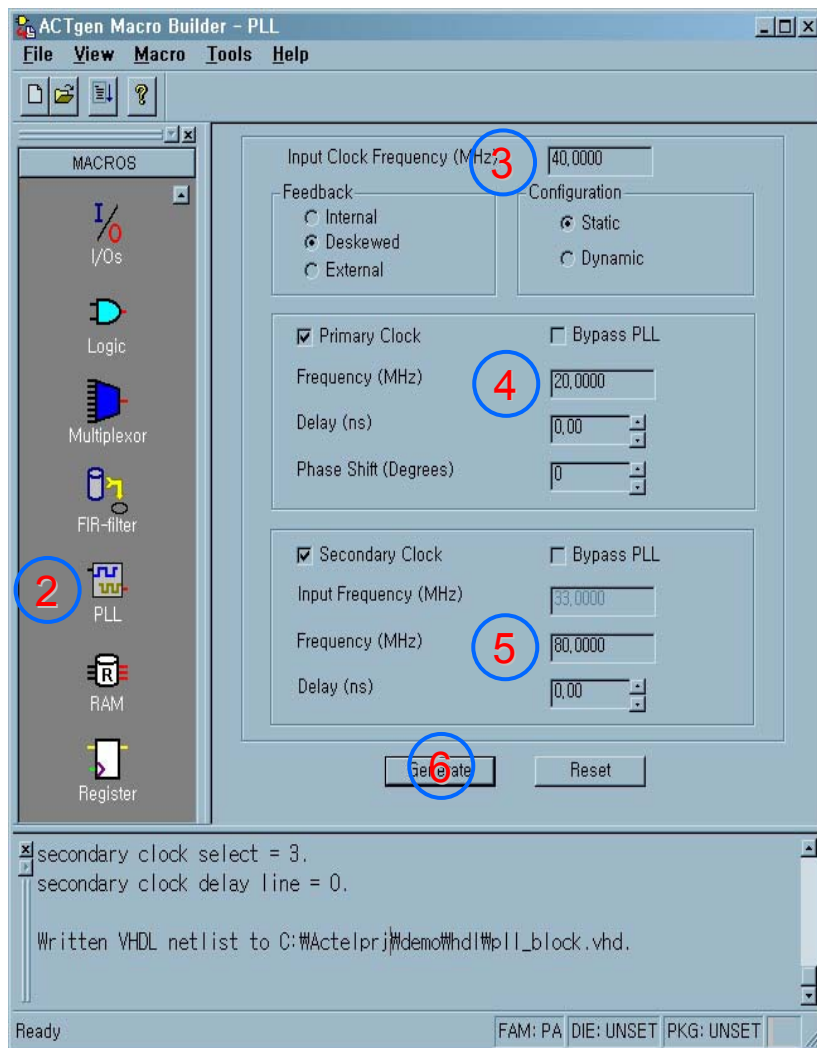
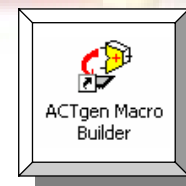
ViewDraw : Schematic editor 생성

ACTgen : Design Macro Builder



- NEW design 생성시
- File -> New
- Design Entry Utilities에서 선택

Create Design – PLL Block



1. Process window에서 actgen 을 선택
2. 왼쪽 MACROS 메뉴에서 PLL 선택
3. Input clock 40Mhz.
4. Primary clock output 20Mhz
5. Secondary Clock output 80Mhz
6. Generate Pll_block.vhd file

Creat Design – Count Block

1. 새로운 vhdl 생성시-> New -> VHDL Entity
2. cnt_block.vhd file 생성
3. HDL Editor Window 에서 cnt_block coding
4. 저장시 Design hierachy 에 cnt_block 추가됨

The image displays two overlapping screenshots of the Libero IDE interface. The background screenshot shows the 'New' dialog box with 'VHDL Entity' selected, and the 'Design Hierarchy' window showing the project structure. The foreground screenshot shows the 'HDL Editor Window' with the code for 'cnt_block.vhd'.

Step 1: The 'New' dialog box is open, showing the 'File' menu. The 'VHDL Entity' option is selected under the 'Design Entry Utilities' section.

Step 2: The 'cnt_block.vhd' file is created and added to the 'Design Hierarchy' window.

Step 3: The 'HDL Editor Window' is open, showing the code for 'cnt_block.vhd'.

Step 4: The code for 'cnt_block.vhd' is shown in the HDL Editor Window. The code is as follows:

```

1  -- cnt_block
2
3  library ieee;
4  use ieee.std_logic_1164.all;
5  use ieee.std_logic_arith.all;
6  use ieee.std_logic_unsigned.all;
7  entity cnt_block is
8      port (count_en, reset, clk20 : in std_logic;
9            count : out std_logic_vector(15 downto 0));
10 end cnt_block;
11 architecture behave of cnt_block is
12     signal tmp : std_logic_vector(15 downto 0);
13 begin
14     process(reset, clk20)
15     begin
16         if reset = '0' then
17             tmp <= (others => '0');
18         elsif clk20'event and clk20='1' then
19             if count_en = '1' then
20                 tmp <= tmp + '1';
21             end if;
22         end if;
23     end process;
24     count <= tmp;
25 end behave;
    
```

The 'Design Hierarchy' window shows the project structure, including the 'cnt_block' entity.

The 'HDL Editor Window' shows the code for 'cnt_block.vhd'.

The 'Warning' message at the bottom states: 'Warning: Unable to recognize top module in file: "C:\Actelprj\demo\hdl\cnt_block.vhd". You can use HDL syntax checker to check it by right-clicking the file name.'



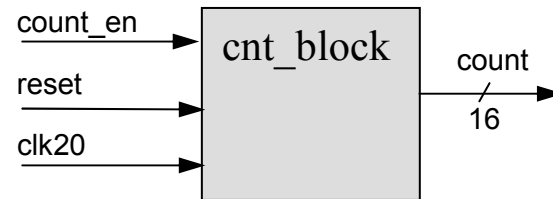
Creat Design – Count Block

```
-- cnt_block
library ieee;
  use ieee.std_logic_1164.all;
  use ieee.std_logic_arith.all;
  use ieee.std_logic_unsigned.all;
entity cnt_block is
  port (count_en, reset, clk20 : in std_logic;
        count : out std_logic_vector(15 downto 0));
end cnt_block;
architecture behave of cnt_block is
  signal tmp : std_logic_vector(15 downto 0);
begin
  process(reset, clk20)
  begin
    if reset = '0' then
      tmp <= (others => '0');
    elsif clk20'event and clk20='1' then
      if count_en = '1' then
        tmp <= tmp + '1';
      end if;
    end if;
  end process;
  count <= tmp;
end behave;
```

1. Library 선언구문

2. Entity 구문

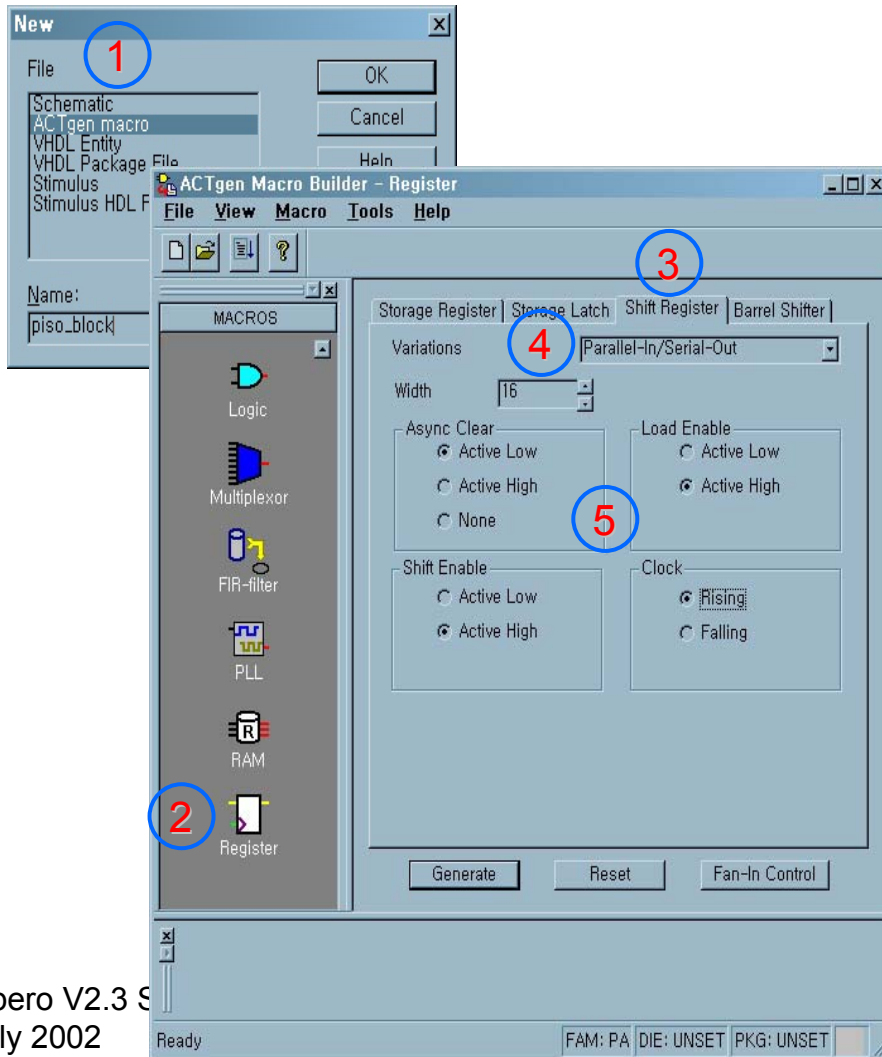
cnt_block의 입출력 signal 선언구문



3. Achitecture 구문

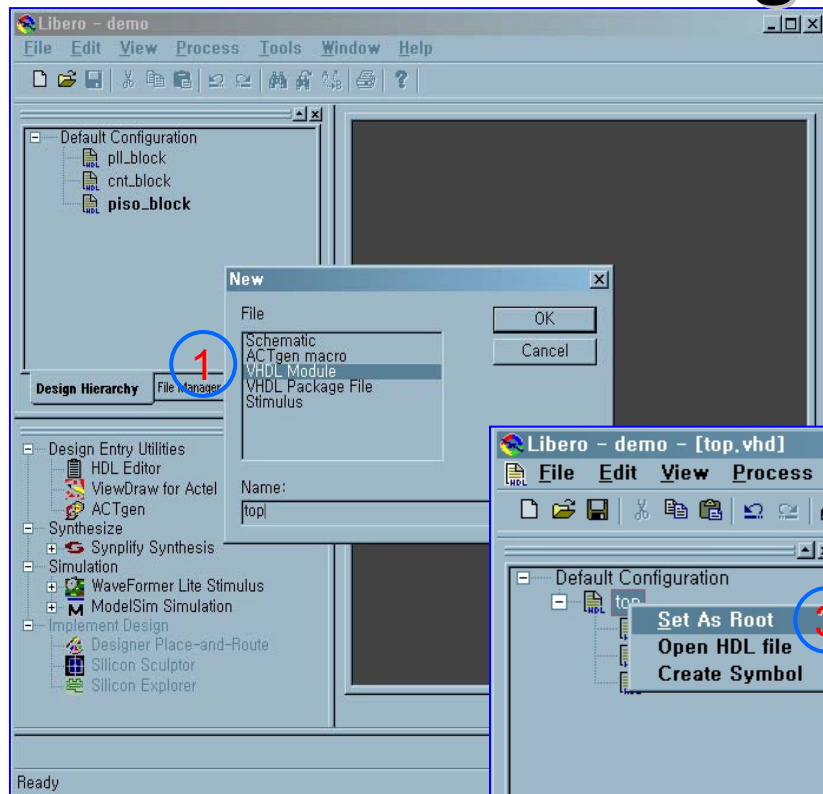
16bit counter with count enable
and asynchronous reset

Creat Design – PISO Block

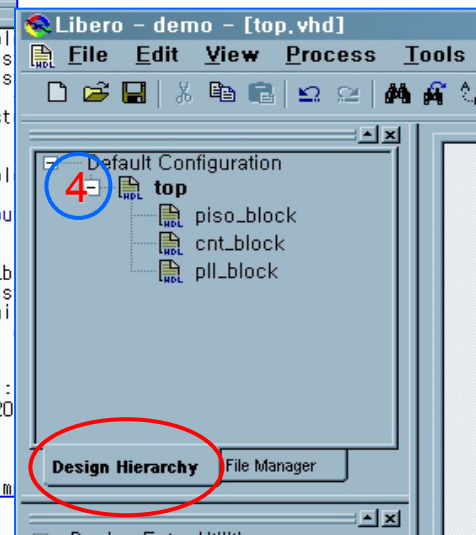
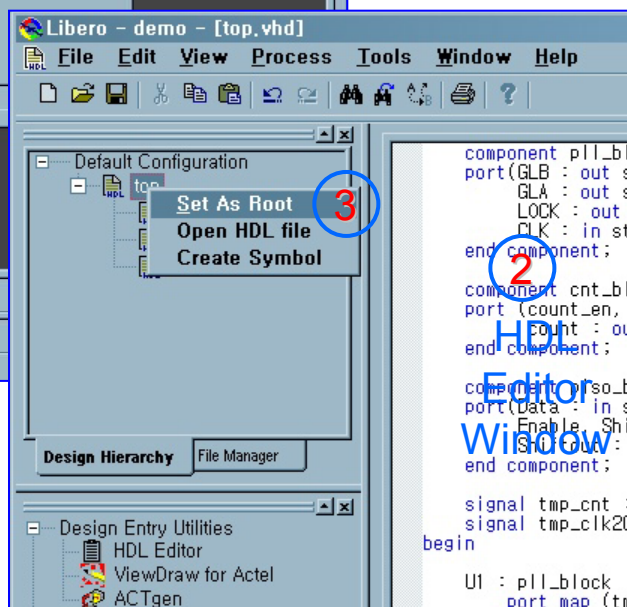


1. New actgen file 생성(new->actgen macro)
2. Macros 에서 Register 선택
3. Register 종류중에 Shift Register 선택
4. Variations => Parallel input to serial output register
5. Register condition 입력
6. Generate

Creat Design – TOP BLOCK



1. 새로운 vhd 생성시-> New -> VHDL Module
2. top.vhd file 생성 및 HDL Editor Window coding 작업
3. Top design을 root로 지정
4. Top design을 굵은 글자체로 전환





Creat Design – Top Design

```
-- top
library ieee;
  use ieee.std_logic_1164.all;
  use ieee.std_logic_arith.all;
  use ieee.std_logic_unsigned.all;

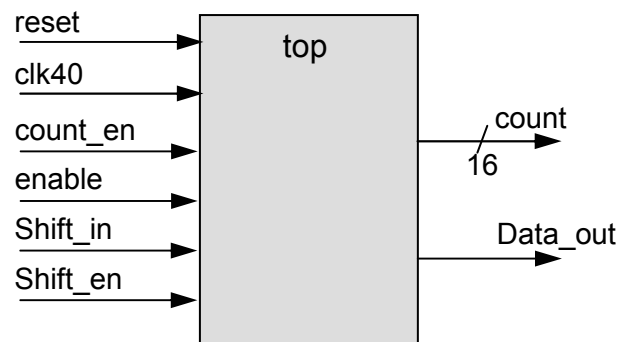
entity top is
  port (reset, clk40 : in std_logic;
        count_en, enable, shift_in, shift_en : in std_logic;
        count : out std_logic_vector(15 downto 0);
        data_out : out std_logic);
end top;

architecture behave of top is
  component pll_block is
    port(GLB, GLA, LOCK : out std_logic;
         CLK : in std_logic);
  end component;
  component cnt_block is
    port (count_en, reset, clk20 : in std_logic;
         count : out std_logic_vector(15 downto 0));
  end component;
  component piso_block is
    port(Data : in std_logic_vector(15 downto 0);
         Enable, Shiften, Shiftin, Aclr, Clock : in std_logic;
         Shiftout : out std_logic);
  end component;
  signal tmp_cnt : std_logic_vector(15 downto 0);
  signal tmp_clk20, tmp_clk80 : std_logic;
begin
  U1 : pll_block
    port map (tmp_clk20, tmp_clk80, open, clk40);
  U2 : cnt_block
    port map (count_en, reset, tmp_clk20, tmp_cnt);
  U3 : piso_block
    port map (tmp_cnt, enable, shift_en, shift_in, reset,
             tmp_clk80, data_out);
  count <= tmp_cnt;
end behave;
```

1. Library 선언구문

2. Entity 구문

top design의 입출력 signal 선언구문



3. Architecture 구문

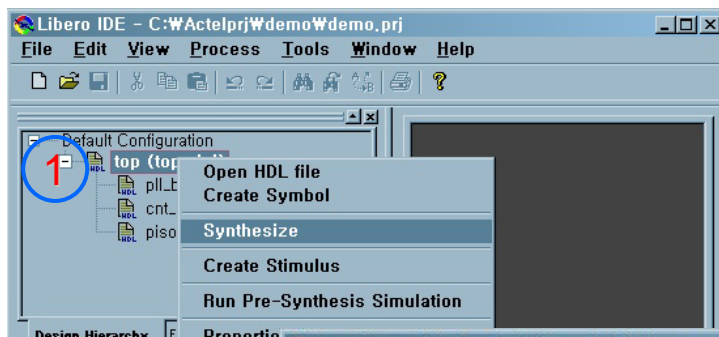
4. Component 선언구문

Sub block pll_block, cnt_block, piso_block 선언

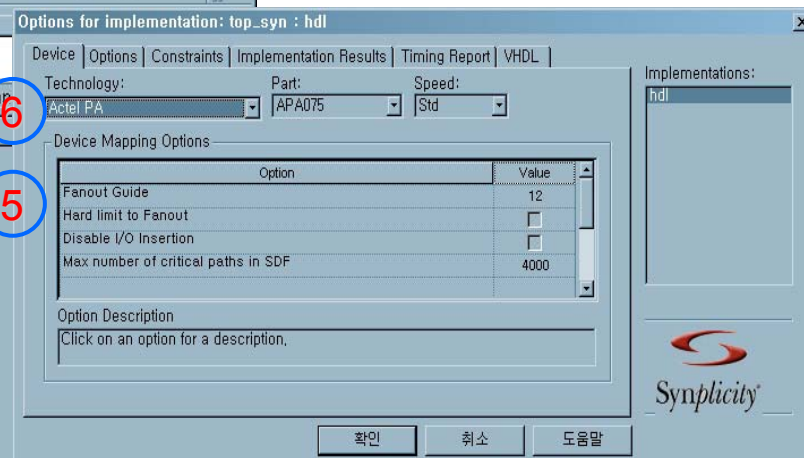
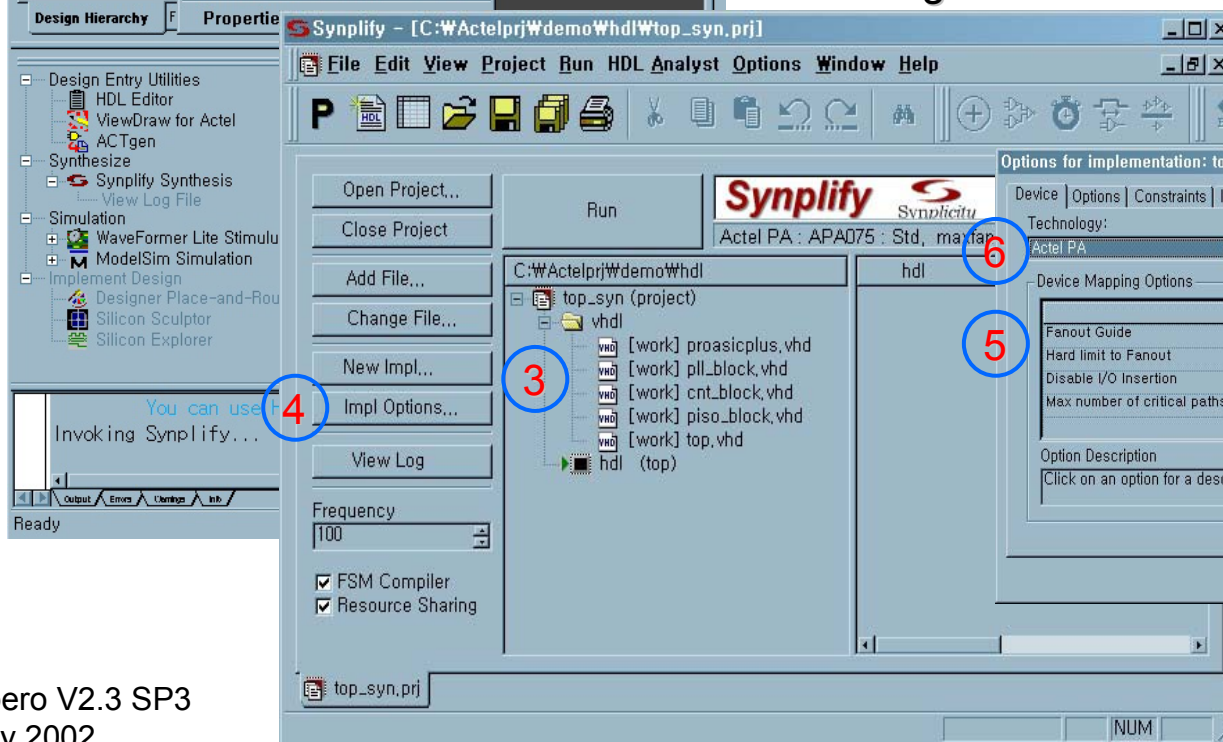
5. Component port map 구문

Sub block간의 signal 연결

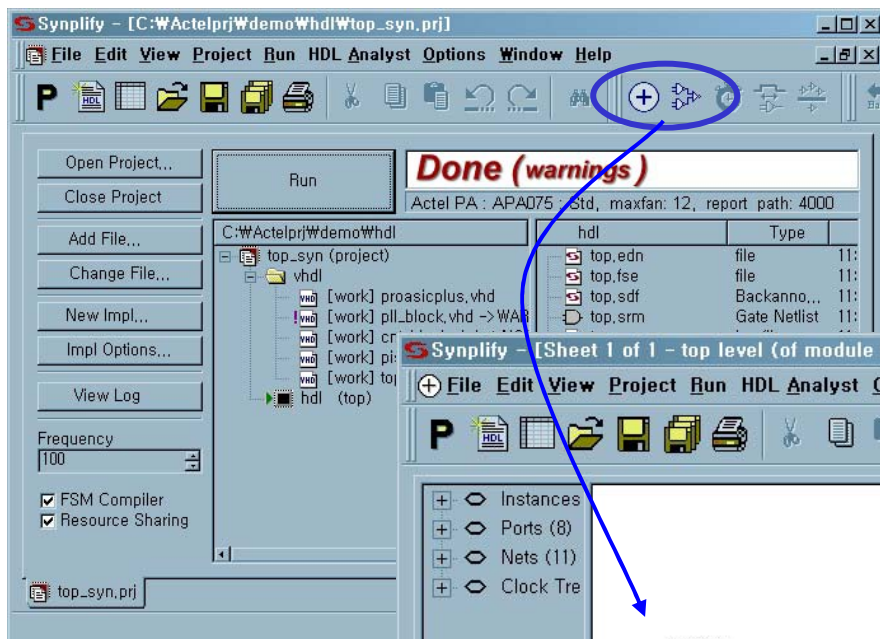
Synplify – Synthesis



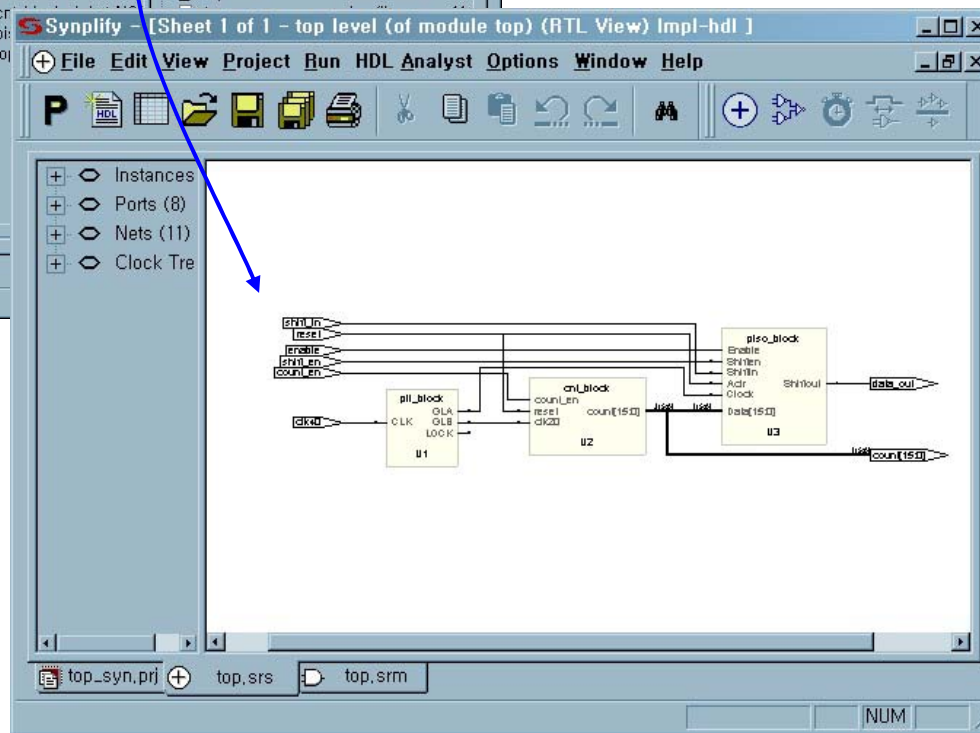
1. Top design에서 오른쪽 마우스 클릭
또는 process window 에서 Synthesize 선택
2. Synthesize 실행
3. Design library & source file list
4. Synthesis implementation options
5. Target device change



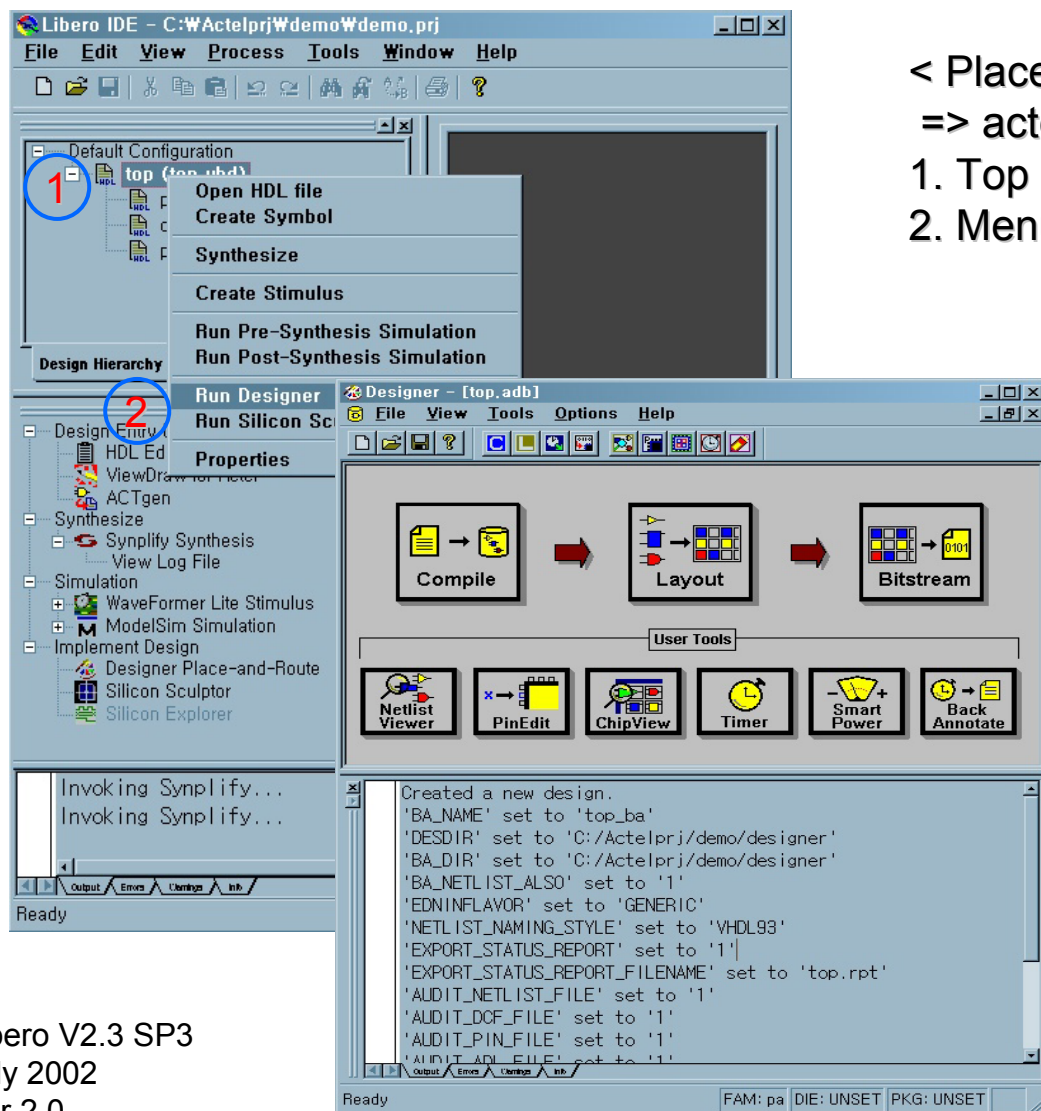
SYNPLICITY – SYNTHESIS



1. Run 실행
2. RTL / Technology View 통해 회로도 검토
3. 출력은 edif (.edn)으로 출력됨
4. Libero 의 file manager에서 확인 가능



Designer Series – Compile 1



< Place & Route 과정 >

=> actel designer series s/w 사용

1. Top design에서 오른쪽 마우스 클릭
2. Menu 중 Run Designer 실행

* Main menu

<Compile> device 선택 및 logic size 분석

<Layout> design place & route

<Bitstream> program하기 위한 bit file 생성

* Sub menu

<Netlist Viewer> gat level circuit 분석

<PinEdit> pin assign

<ChipView> layout 후 내부 cell의 위치정보 확인

<Timer> design timing 정보

<SmartPower> power 소모량 측정

<BackAnnotate> Timing sim을 위한 파일 출력



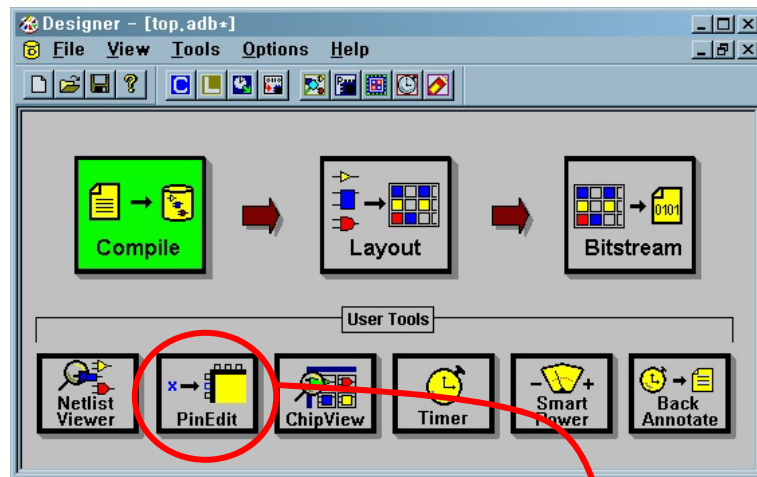
Designer Series – Compile



1. Device (gate,package,speed,voltage)설정
- 2 Jtag pin 설정
3. Device (Operating Conditions) 설정
Temperature & Voltage : com, ind, mil
4. 마침



Designer Series – PinEdit



1. Compile 완료시 녹색 아이콘으로 변함.

2. Device 사용량

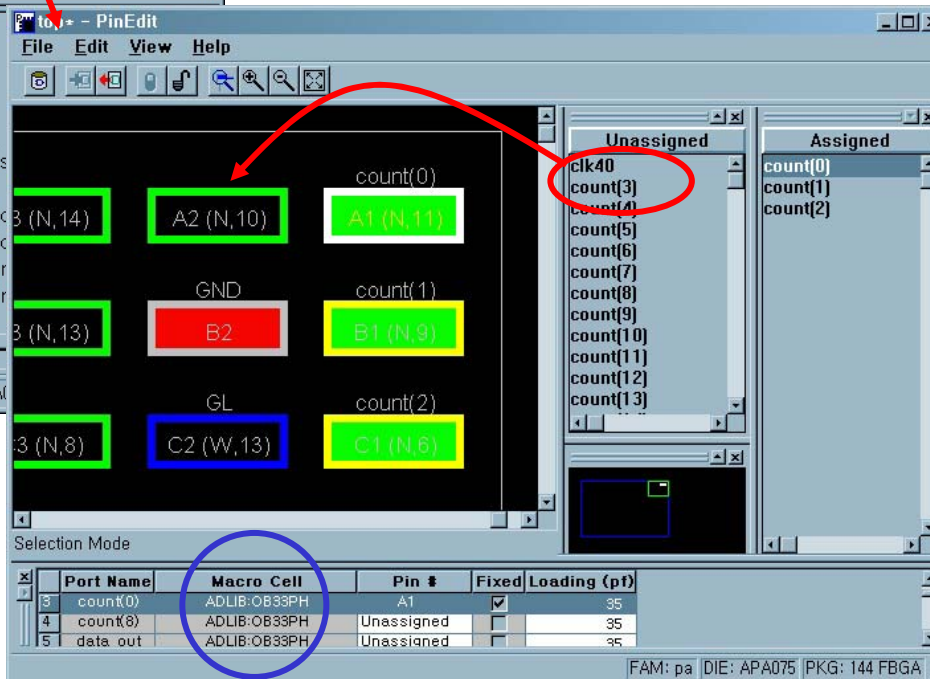
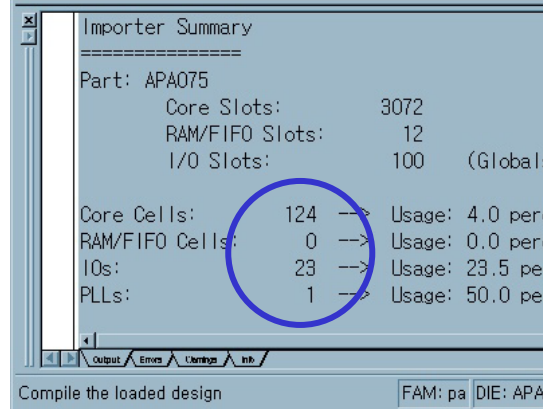
Core cells : 124/3072

RAM/FIFO Cells : 0/12

I/Os : 23/100

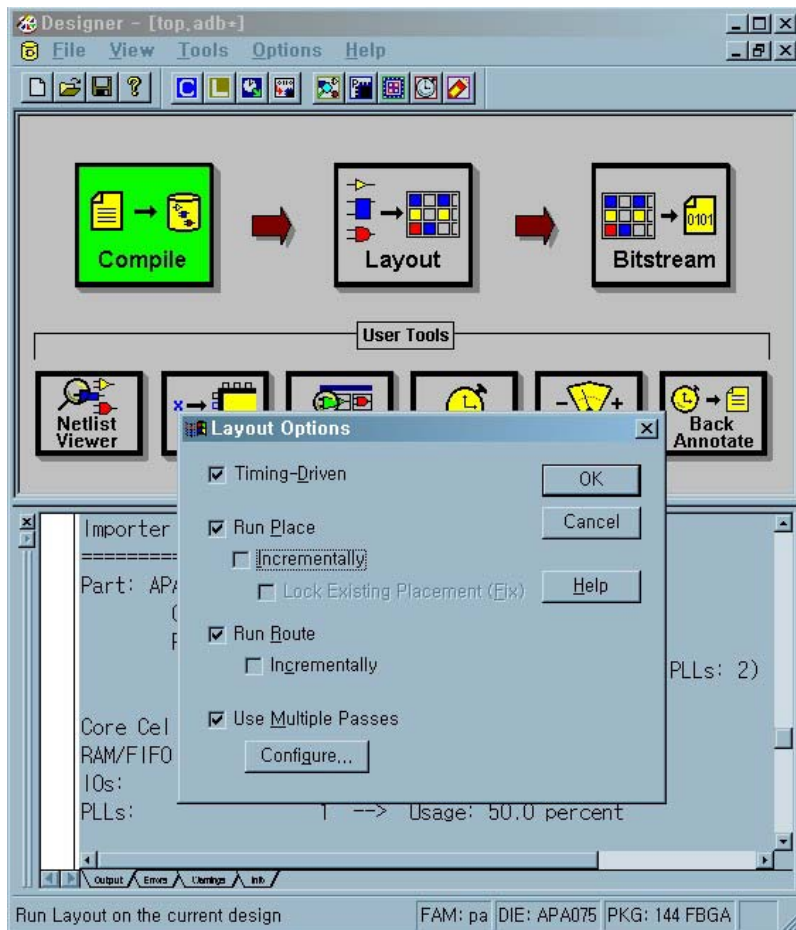
PLLs : 1/2

3. PinEdit => drag and drop 방식으로 pin assign





Designer Series – Layout



1. Layout options

<Timing-Driven> constraint file(gcf)에 명시된 조건에 따라 layout을 실행함.

<Run Place>

Incrementally : Select to use previous placement data as the initial placement for the next place run.

Lock Existing Placement (fix) : Select to preserve previous placement data during the next incremental placement run.

<Run Route>

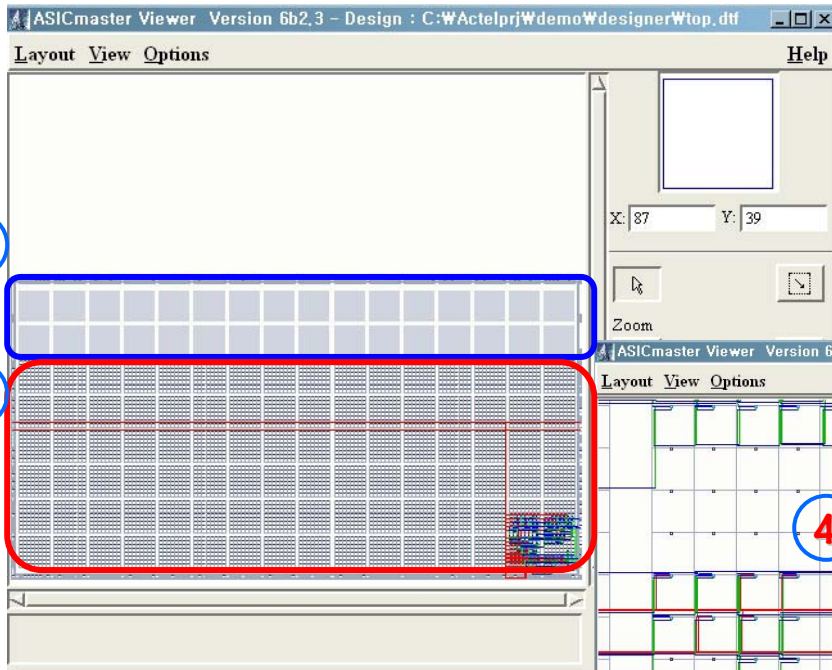
Incrementally :

<Use Multiple Passes> layout 시 시작 시점(Seed)을 다르게 하여 전혀 다른 layout 결과를 가져와 성능 향상에 도움이 됨

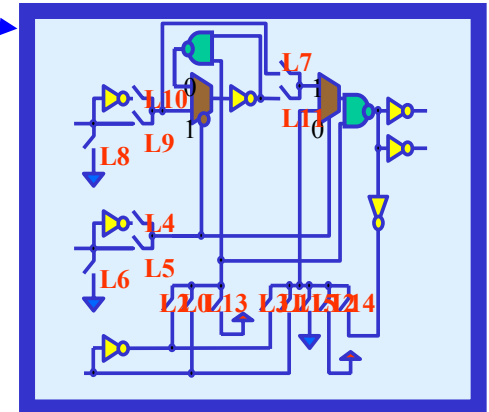
2. Ok



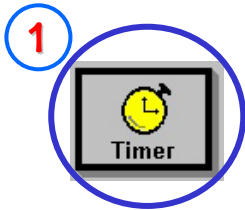
Designer
Series



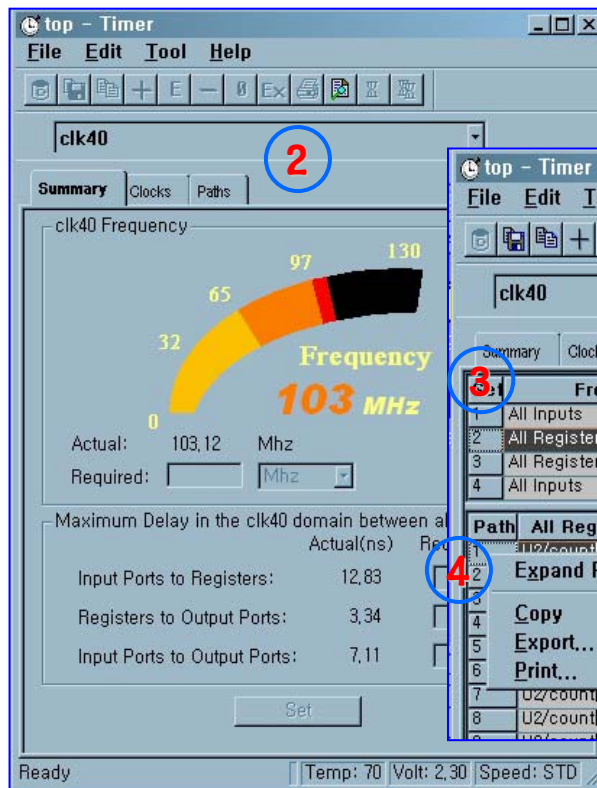
-
- ASICmaster Viewer Version 6b2.3 - Design : C:\W\Telprj\demo\Wdesigner\Wtop.dtf
- Layout View Options Help
- 4
- 5
- X: 126 Y: 3
- Zoom
- Select
- Cell
- ID Area Enlarge
- Core Cell: G_18_1 S
- Type: XOR2
- Hide Exit



Designer Series – Timer

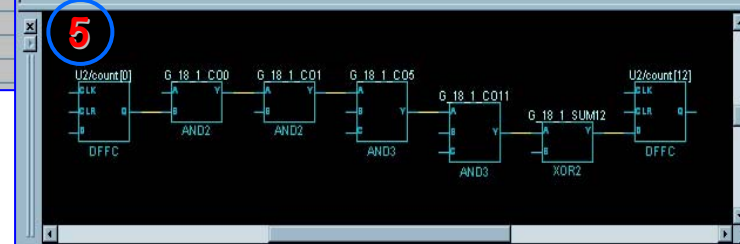


1. Timer 열기
2. Clk40에 대한 clock performance
3. Path 별 timer 분석
4. Register to Register delay 분석



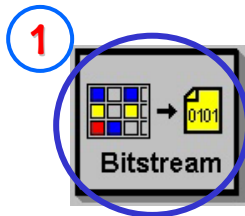
	From	To
1	All Inputs	All Registers / clk40
2	All Registers / clk40	All Registers / clk40
3	All Registers / clk40	All Outputs
4	All Inputs	All Outputs

Grid1	Instance	Net	Macro	Delay	Type	Total	Fanout
1	U2/count[12]:D	tmp_3[12]	ADLIB:DF	0.66 (r)	Setup Del	9.70	0
	U2/tmp_3[12]	U2_tmp_3_12	cnt_block	0.32 (f)	Net Delay	9.04	0
	G_18_1_SUM12:Y		ADLIB:XO	0.55 (f)	Gate Del	8.72	1
	G_18_1_SUM12:A	G_18_1_COZ	ADLIB:XO	1.28 (f)	Net Delay	8.16	1
	G_18_1_CO11:Y		ADLIB:AN	0.57 (f)	Gate Del	6.88	1
	G_18_1_CO11:A	G_18_1_COZ	ADLIB:AN	0.96 (f)	Net Delay	6.32	1
	G_18_1_CO5:Y		ADLIB:AN	0.33 (f)	Gate Del	5.36	10
	G_18_1_CO5:A	G_18_1_COZ	ADLIB:AN	2.26 (f)	Net Delay	5.03	1
	G_18_1_CO1:Y		ADLIB:AN	0.41 (f)	Gate Del	2.77	5
	G_18_1_CO1:A	G_18_1_COZ	ADLIB:AN	0.46 (f)	Net Delay	2.36	1
	G_18_1_CO0:Y		ADLIB:AN	0.68 (f)	Gate Del	1.90	2
	G_18_1_CO0:B	tmp_cnt[0]	ADLIB:AN	0.51 (f)	Net Delay	1.22	1
	U2/tmp_cnt[0]	tmp_cnt[0]	cnt_block	0.00 (f)	Net Delay	0.71	3
	U2/count[0]:Q		ADLIB:DF	0.71 (f)	Gate Del	0.71	1
	U2/count[0]:CLK	tmp_clk20	ADLIB:DF	0 (r)	Net Delay	0.00	1





Designer Series – Timer



1. Bitstream 열기

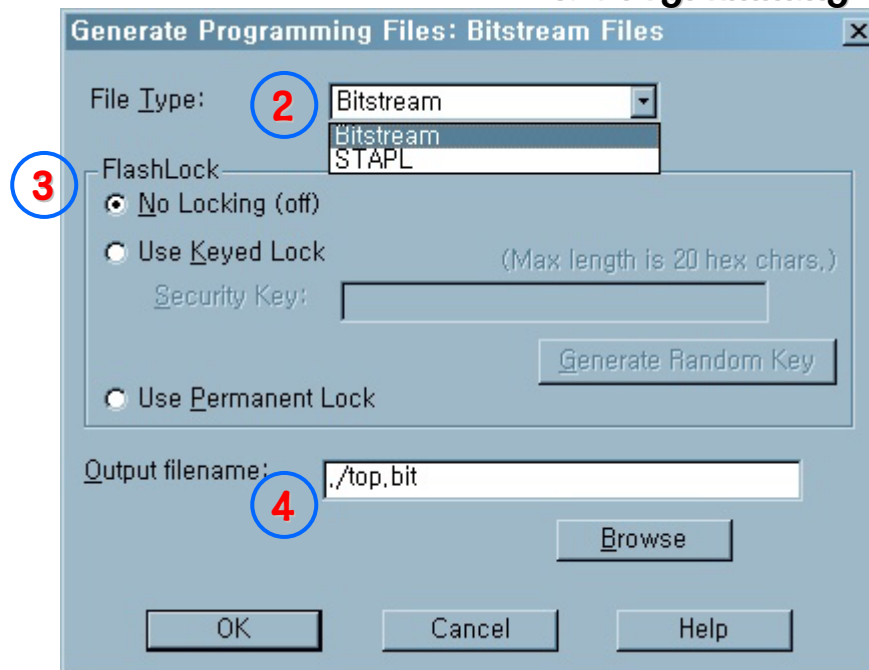
2. File Type 선택

BitStream : silicon sculptor를 이용한 programming

STAPL : flash pro and lite 를 이용한 programming

3. FlashLock 입력시 program, erase 불가

4. Programming file 생성



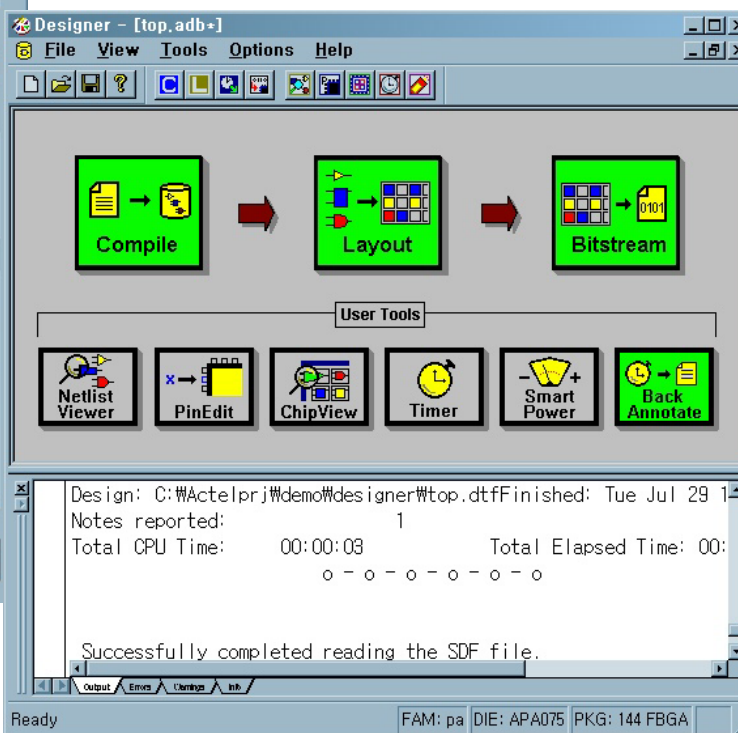
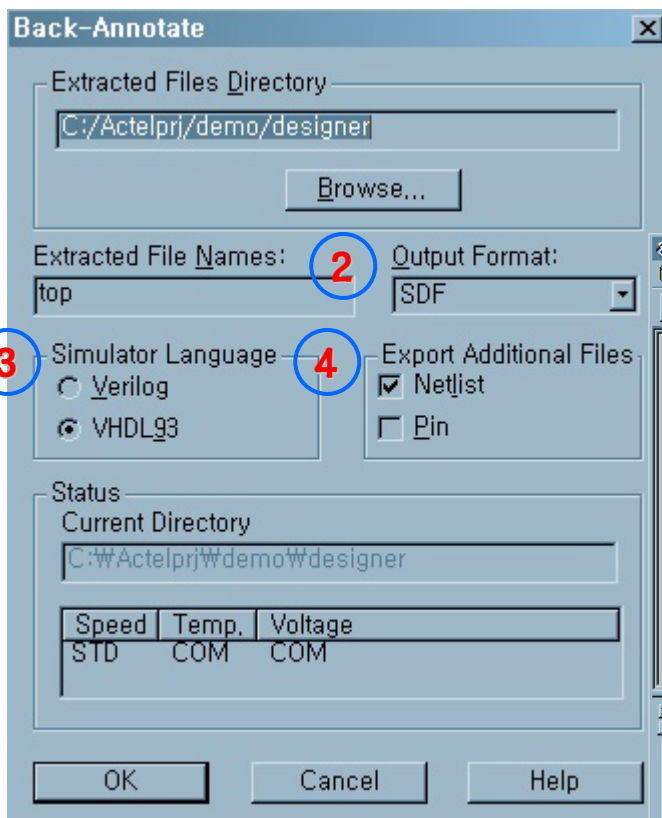
Designer Series – Backannotation



1

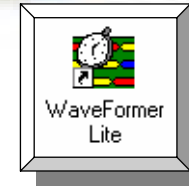


1. Backannotate 열기
2. SDF(Standard Delay Format) 출력
3. Timing simulation을 위한 netlist VHDL 파일 출력
4. Pin file / Netlist 출력 가능
5. Designer Place and Route 완료, top.adb 저장





Waveformer Lite



1. TOP design에서 Create Stimulus 열기
2. Clk40에 clock 속성 부여

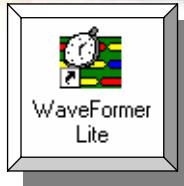
<note> signal list에서 진하게 출력되는 signal 은 input을 나타내고 얇게 출력되는 signal은 output을 나타낸다.

Adjust signal levels

I/O signals

2

1



WaveFormer
Lite

Edit Clock Parameters

Name:

Reference Clk:

Freq: ⓂHz / ns

Period: ns

Period Formula: ex. 2*CLK0.period

Starting Offset:

Duty Cycle %:

Rise Jitter:

Fall Jitter:

Buffer Delay

Min L to H:

Max L to H:

Min H to L:

Max H to L:

Rising Delay Correlation: %

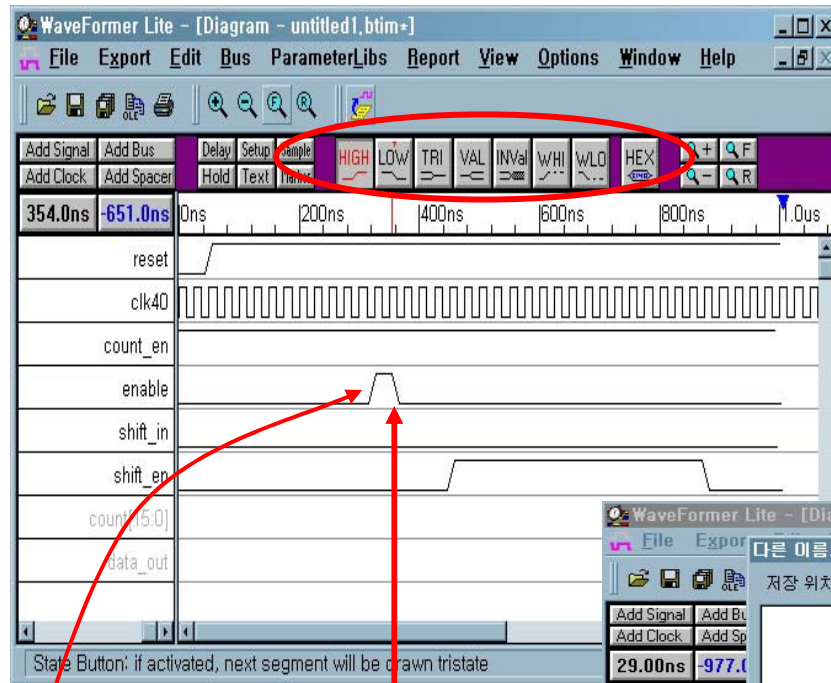
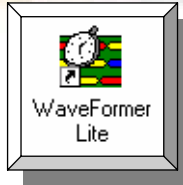
Falling Delay Correlation: %

Rise to Fall Correlation: %

☐ Invert (Starts Low)

- **Clock duty cycle**

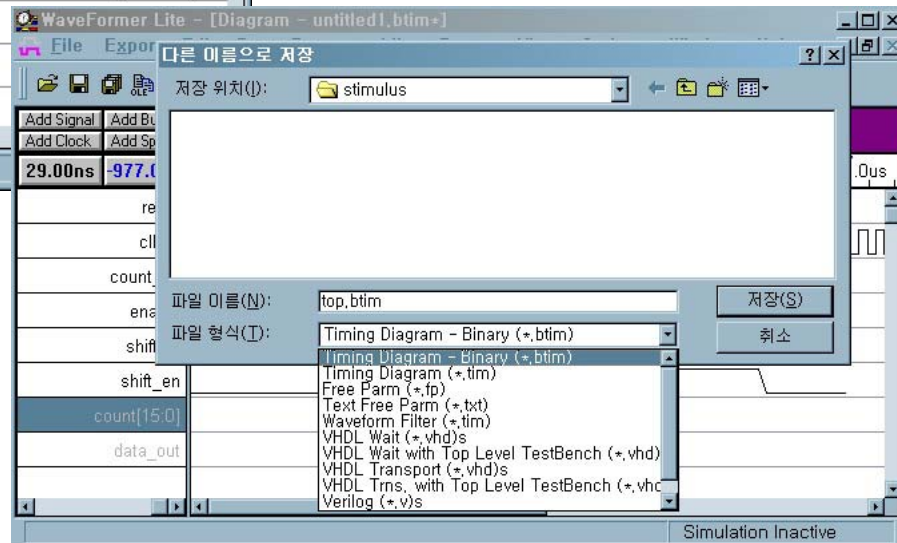
Waveformer Lite – Export files 1



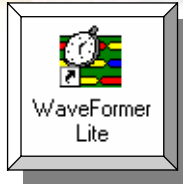
Adjust the transition times

Edge를 Click하여 move 가능
중간에 drag 하여 signal 값 변경가능

1. Bit 입력은 high/low icon을 이용하여 입력
2. Save files
Timing Project => top.tim
Testbench VHDL=> top.vhd
=> VHDL Wait with Top Level
TestBench 을 선택



WaveFormer Lite – Export files 2



WaveFormer Lite

File Export Edit Bus Libraries Report View Options Window Help

Diagram - top_tb.tim

1.862us -134.0ns

0ns 500ns 1.0us 1.5us 2

reset

clk40

count_en

enable

shift_in

shift_en

Timing Project File

Parameter - top_tb.tim

name	min	max	margin	comment

Report - top_tb.vhd

-- Contains an instance of the Stimulus Module and an instance of the Product Module

library ieee, std;

use std.textio.all;

use ieee.std_logic_1164.all;

use work.all;

entity testbench is

end testbench;

architecture tbGeneratedCode of testbench is

-- Component declaration for the stimulus module

component stimulus

port(

reset : out std_logic;

clk40 : out std_logic;

count_en : out std_logic;

enable : out std_logic;

shift_in : out std_logic;

shift_en : out std_logic;

count : in std_logic_vector(15 downto 0);

data_out : in std_logic

);

end component;

-- Component declaration for the product module

component top

port(

reset : in std_logic;

clk40 : in std_logic;

count_en : in std_logic;

enable : in std_logic;

shift_in : in std_logic;

shift_en : in std_logic;

count : out std_logic_vector(15 downto 0);

data_out : out std_logic

);

end component;

-- Signal Declaration for the test bench module

signal reset : std_logic;

signal clk40 : std_logic;

signal count_en : std_logic;

signal enable : std_logic;

signal shift_in : std_logic;

signal shift_en : std_logic;

signal count : std_logic_vector(15 downto 0);

signal data_out : std_logic;

-- Ports are connected by matching the port names of the Test Module

begin

stimulus_0 : stimulus

end

Testbench VHDL File

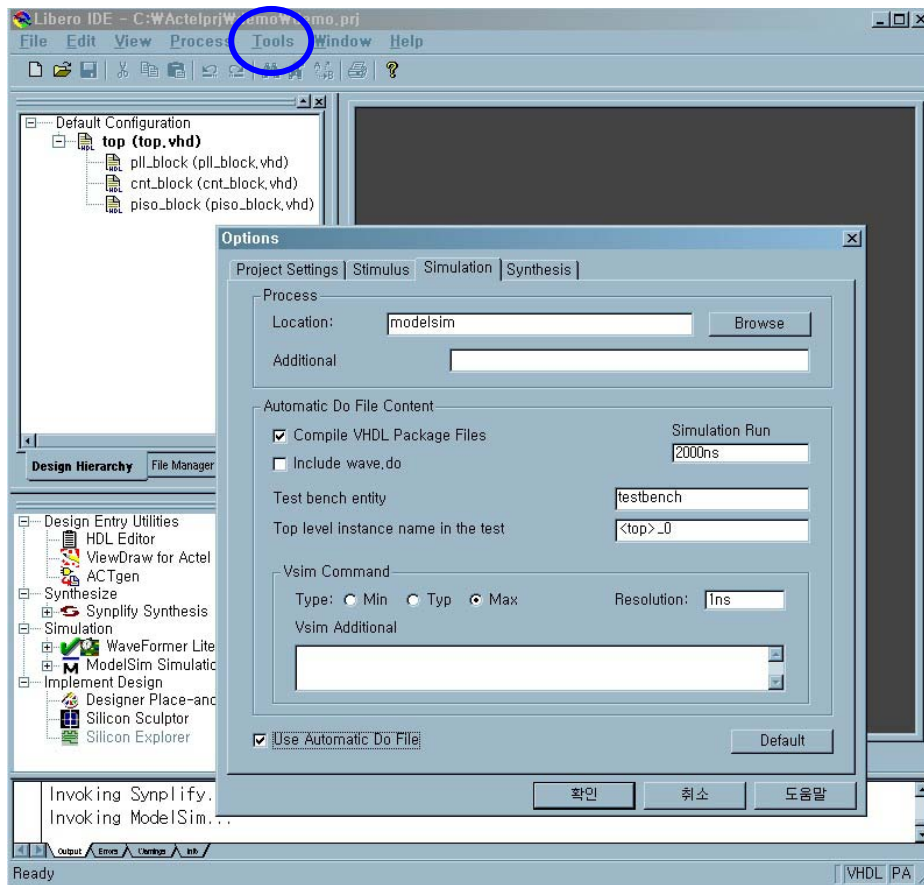
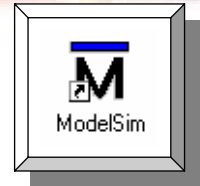
Row: 1 Line: 92 Col: 13

verilog.log / waveperl.log / Errors / Differences / Grep / top_tb.vhd

Simulation Inactive



ModelSim – Simulation 1



1. Simulation 하기 위한 options

2. Tools -> options 선택

3. Simulation tab 을 선택

<Simulation Run> : Simulation 의 끝점을 선정.

(ex: 2000ns : 결과를 2us 까지 출력)

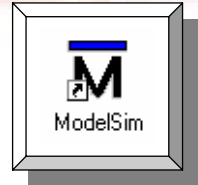
<Compile VHDL Package Files> : simulation 할때마다 package file을 다시 compile 함.

<Include wave.do> : user가 원하는 signal을 wave.do 로 저장하여 다음 simulation 시 적용됨.

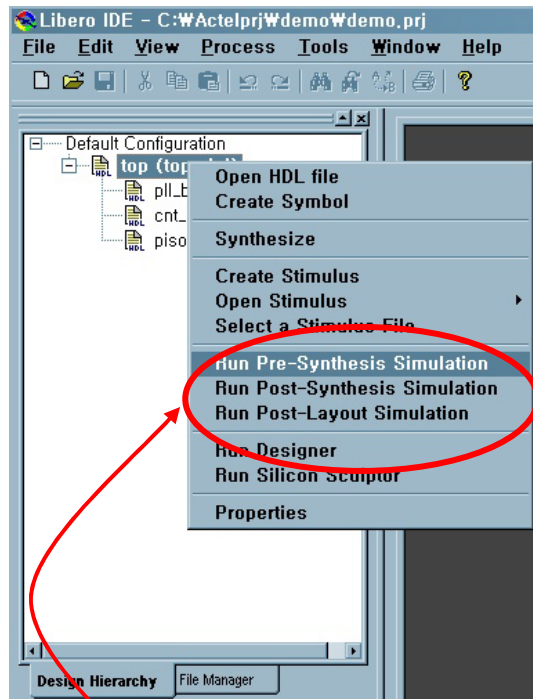
<Vsim Command> : Timing Simulation 시 Min(best)/Typ(typical)/Max(worst) case 를 선택하여 simulation 함.

<Resolution> : simulation 결과 출력시 sampling time을 의미함. 값이 작을수록 더 정확한 결과를 얻을 수 있음.

ModelSim – Simulation 2

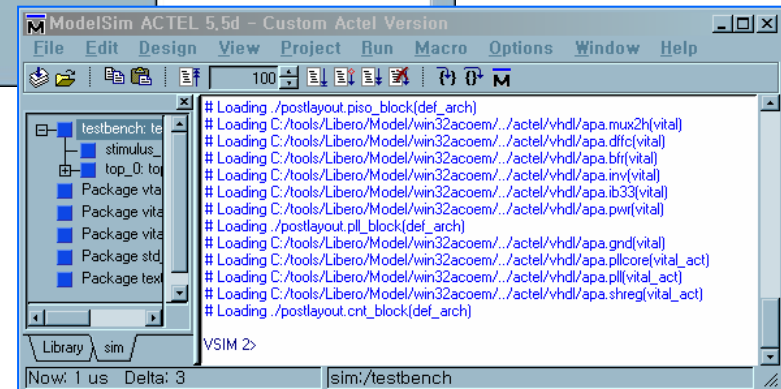
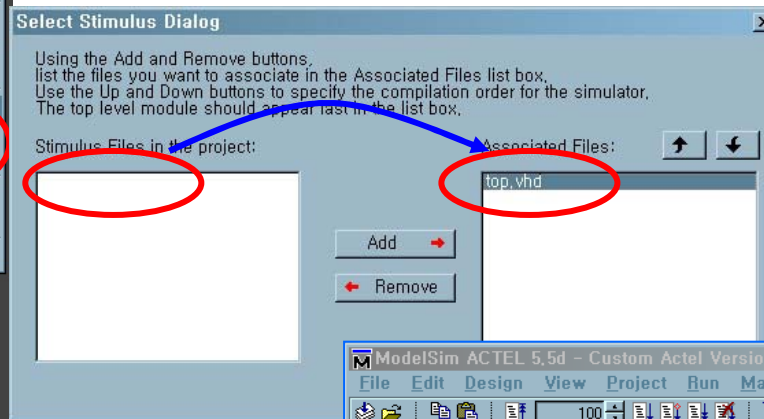


1. Modelsim Simulator 열기
2. 여러 개의 stimulus 가 있을 경우 한 개를 선택
3. ModelSim 이 열리면서 자동적으로 compile, load design, simulation 실행



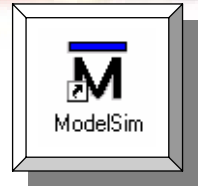
Run Pre_Synthesis Simulation : Function simulation
 Run Post_Synthesis Simulation : Gate simulation
 Run Post_Layout Simulation : Timing simulation

***정확한 design의 검증을 위해서는 위 세 가지의 모든 경우에 대해서 simulation 결과가 일치해야 함.**



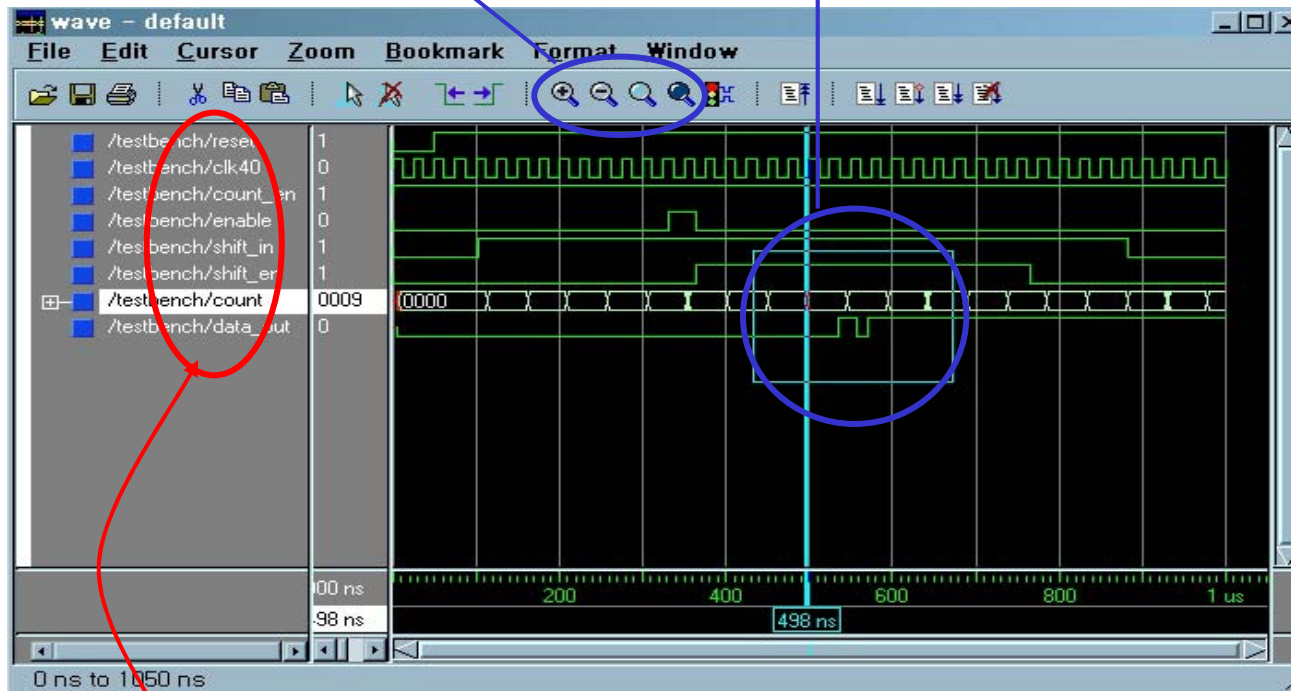


ModelSim – Simulation 3



Zoom in/out icon

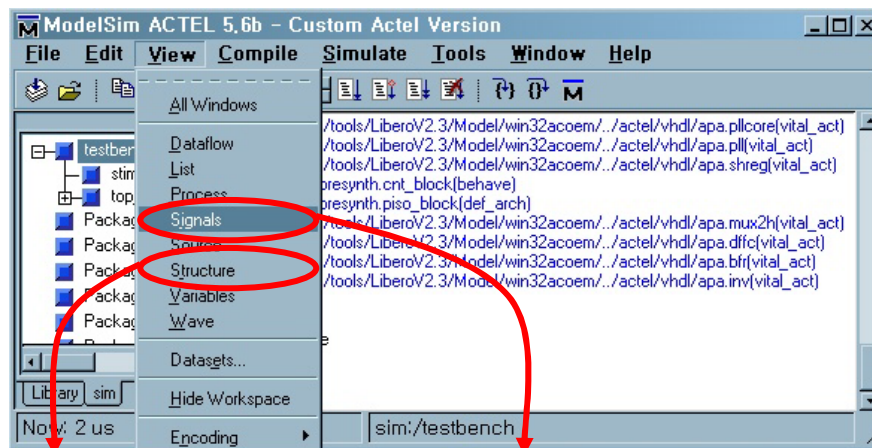
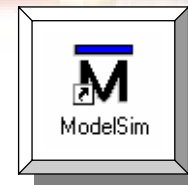
가운데 마우스 키를 이용하여 zoom area 가능



Input/output signal list

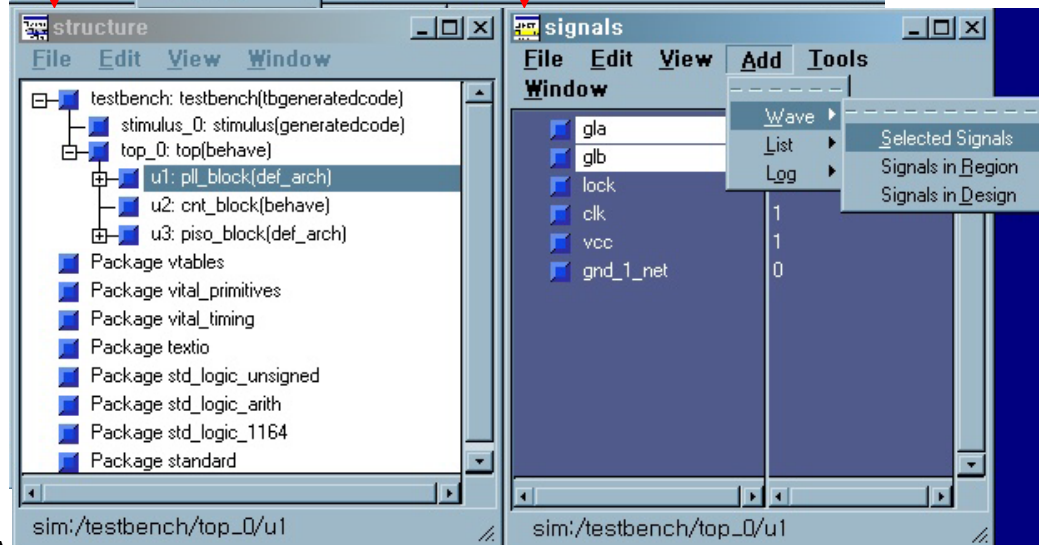


ModelSim – Simulation 4

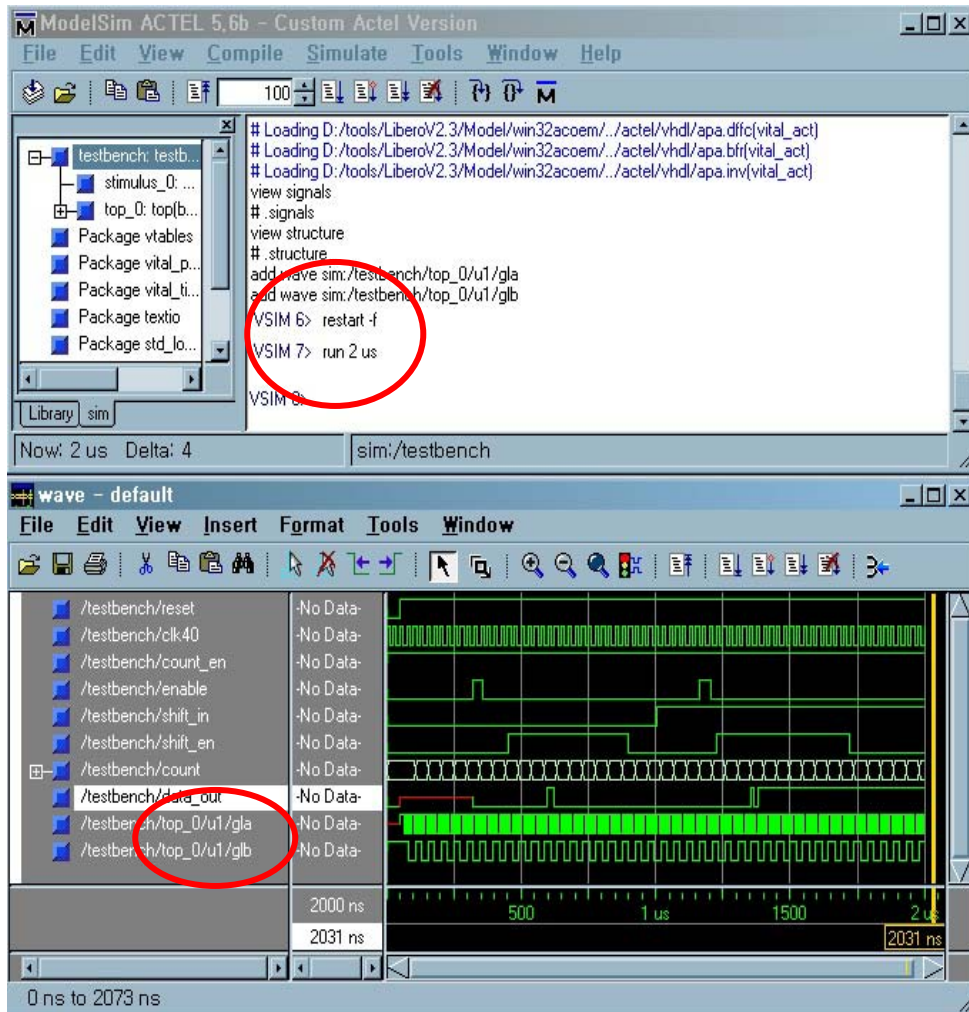
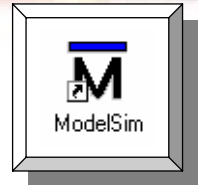


<내부 signal을 보기위한 방법>

1. View -> Signals 과 Structure 창을open
2. Structure 창에서 내부 block으로 이동
3. Signals 창에서 원하는 signal 을 선택하여 add->wave 를 한다.

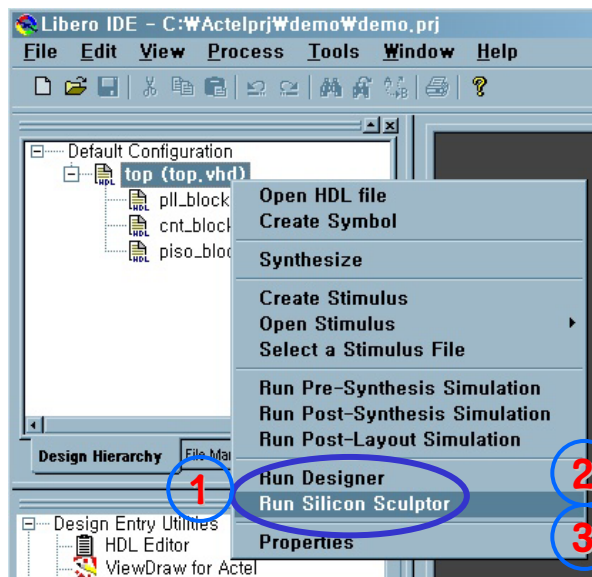
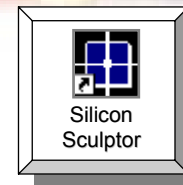


ModelSim – Simulation 4

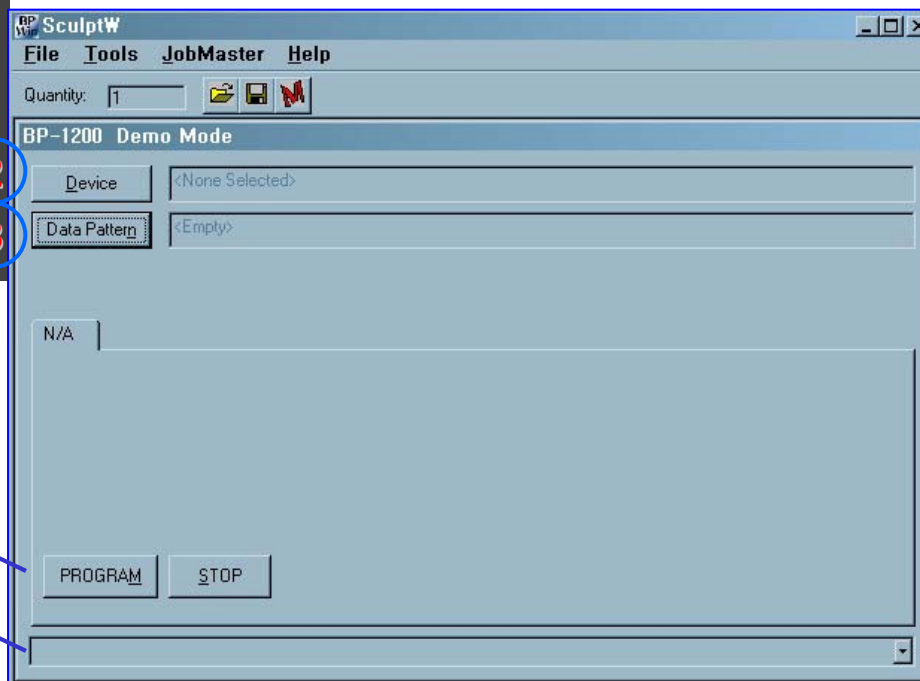


1. Modelsim main windows에서
vsim #> restart -f
vsim #> run 2 us
를 실행
2. Wave 창에서 결과 확인
3. Wave 창에서 wave.do로 저장하게
되면 다음에 이 추가된 signal이
자동으로 올라오게 된다.

Silicon Sculptor – programming



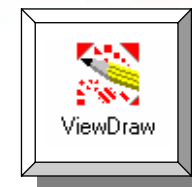
1. Silicon Sculptor Programmer 실행
2. Device 선택
3. Fusing file(bitstream 선택)



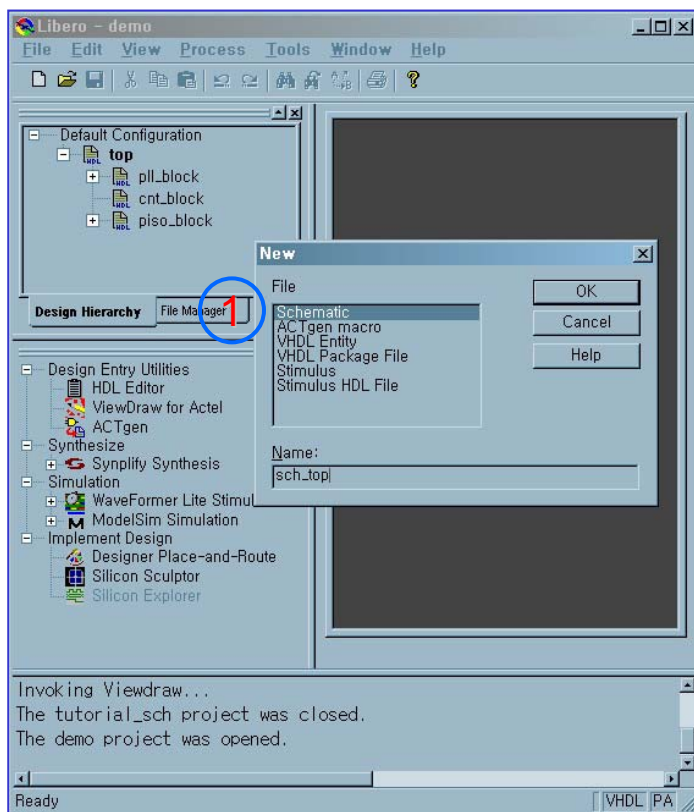
4. Program
5. Program status



Creat Design – Schematic



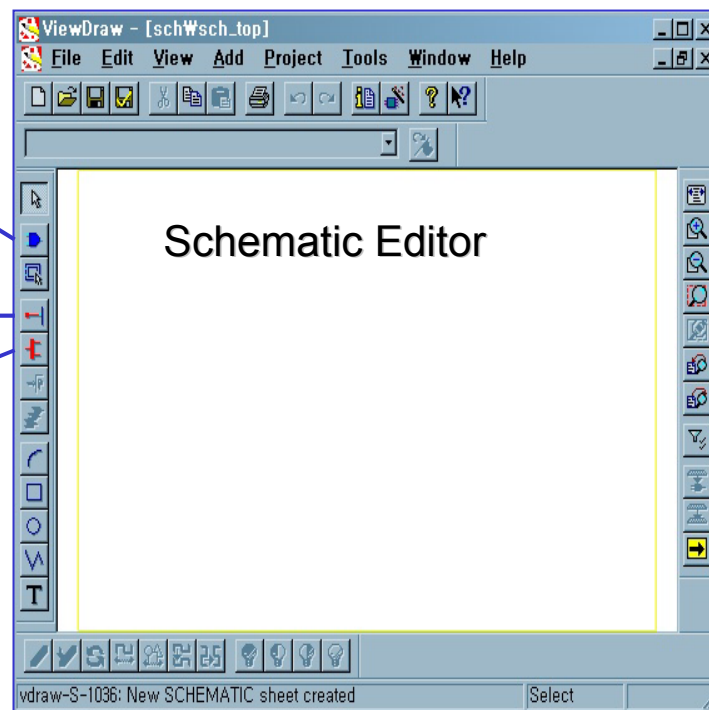
1. New Schematic file 생성(new->Schematic)



Component

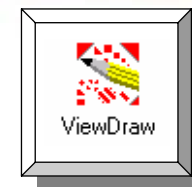
Net

Bus





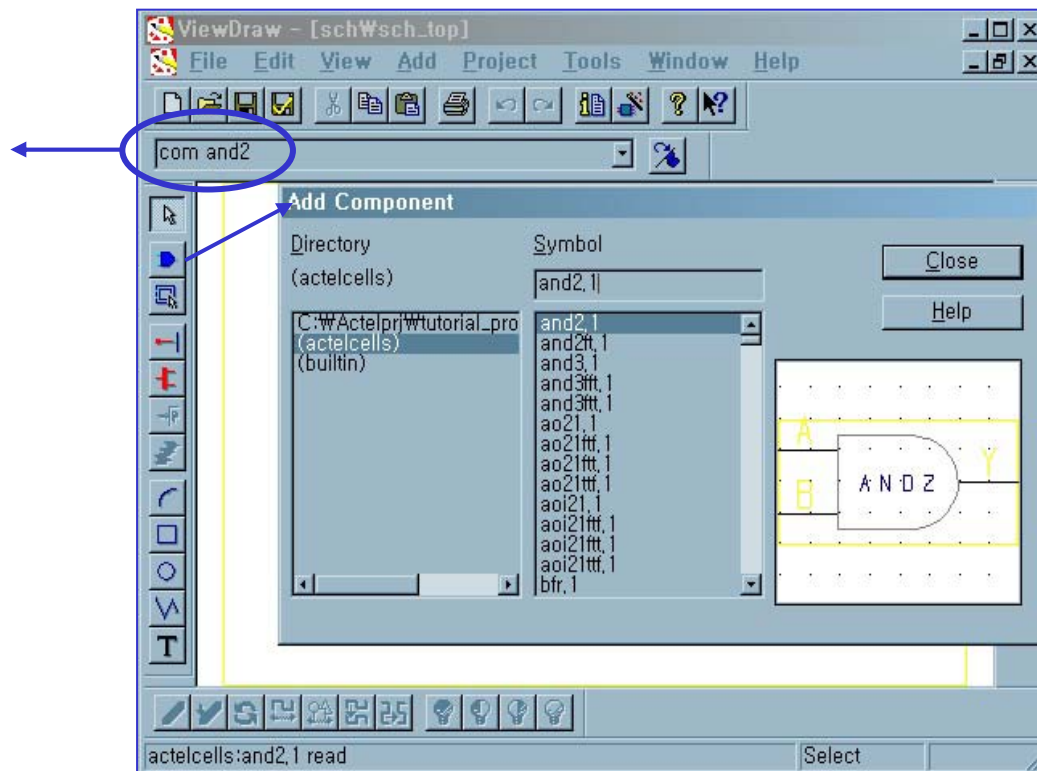
Creat Design – Schematic



1. Component 선택시 두가지 방법

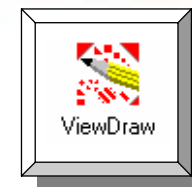
i. Command line 에서 원하는 component 를 입력
=> com and2

ii. Add component 에서 원하는 component 를 선택





Creat Design – Schematic



1. Net 연결은 net icon을 이용하여 component 간의 pin을 연결한다.

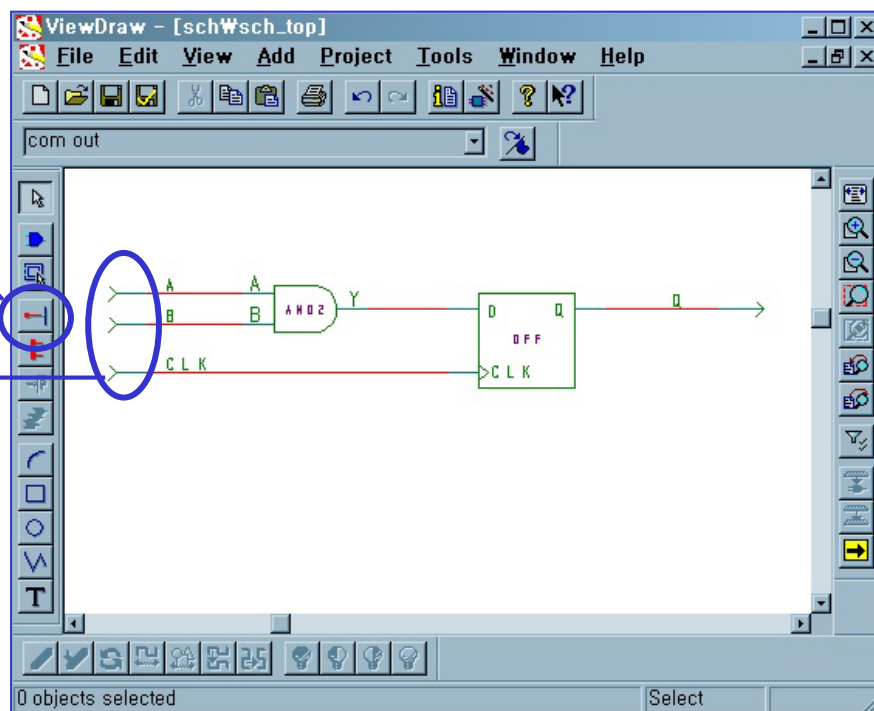
i. 단축키 <n>를 이용할 수 있다.

2. Block의 in/out은 component 에서 in 또는 out을 선택하여 net 끝에 연결한다.

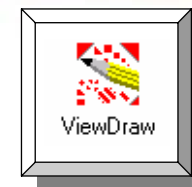
3. Net name은 net properties에 label에 이름을 준다.

i. In/out의 모든 net

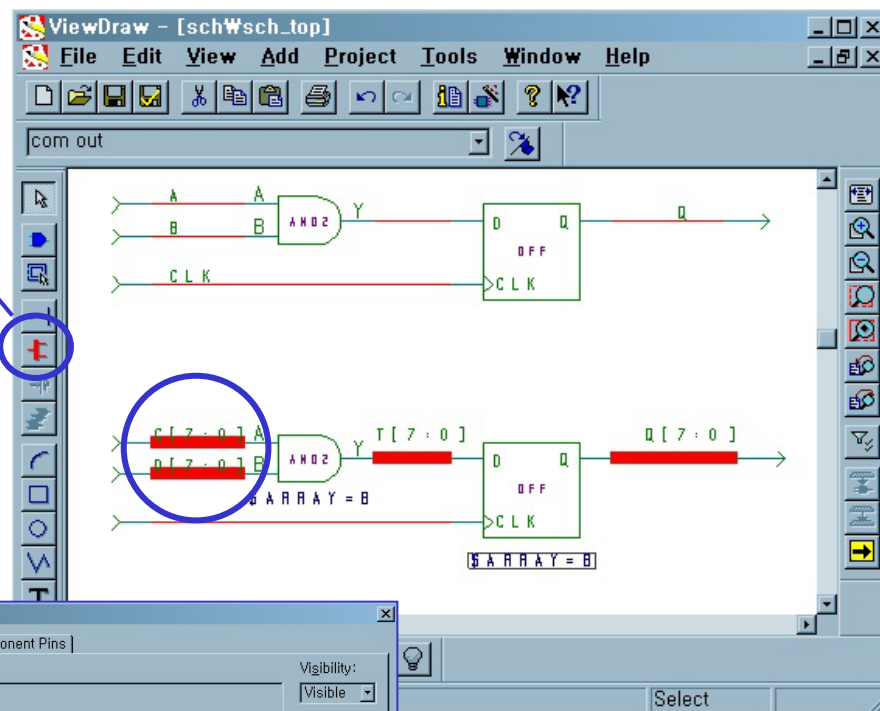
ii. 모든 bus 에는 이름을 주어야 한다.



Creat Design – Schematic

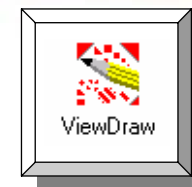


1. Bus 연결은 bus icon을 이용하여 component 간의 pin을 연결한다.
i. 단축키 를 이용할 수 있다.
2. Bus name은 Bus properties에 label에 name을 준다.
3. Index를 위하여 [7:0]의 형태를 갖는다.
4. 배열을 주기위해서 component properties에서 array option을 이용한다.

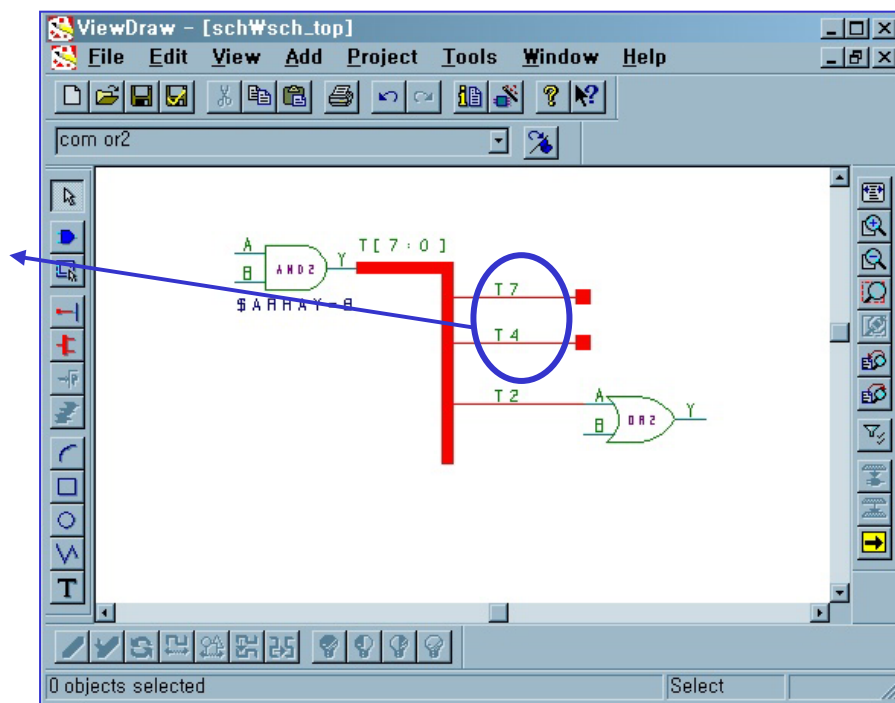




Creat Design – Schematic

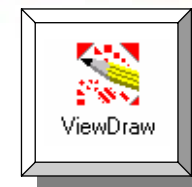


1. Bus 에서 bit 의 분리는 bus라인에서 net 바로 연결하여 사용한다.
2. Index를 bus name 에 바로 붙여사용

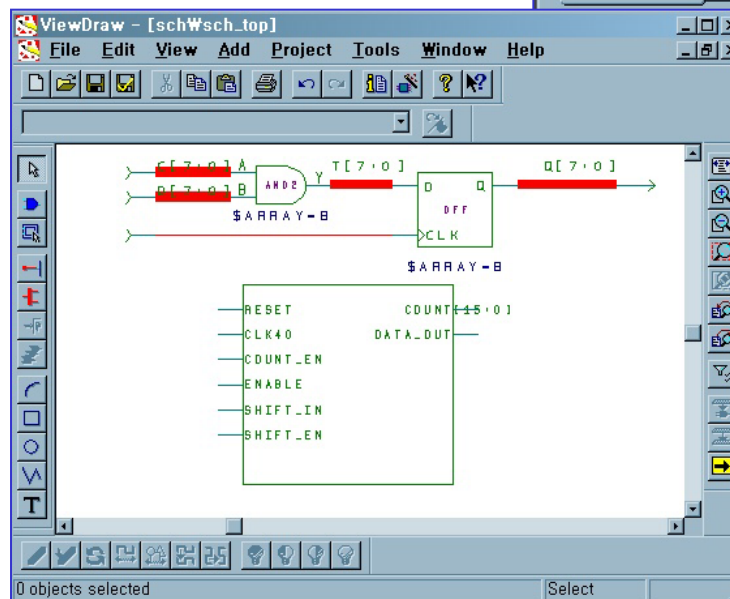
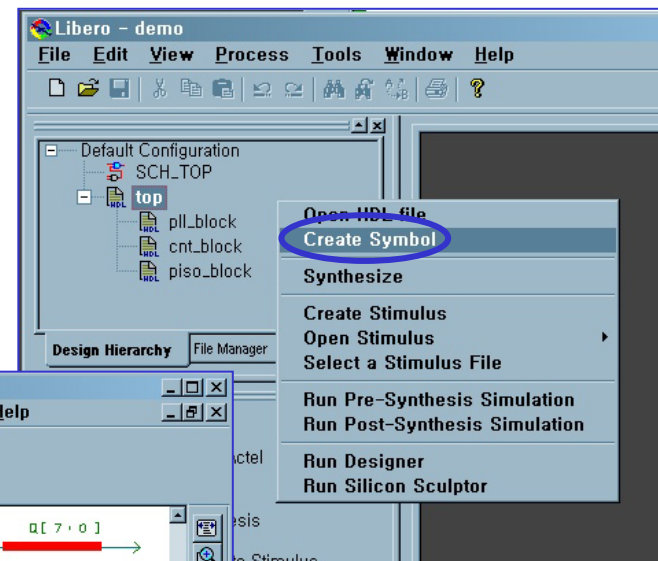


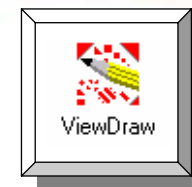


Creat Design – Schematic



1. Vhdl 로 구성된 top design을 schematic의 sub block으로 만들기 위해서 create symbol
2. Add component에서 top design을 import 하여 net를 연결한다.





ViewDraw

-

