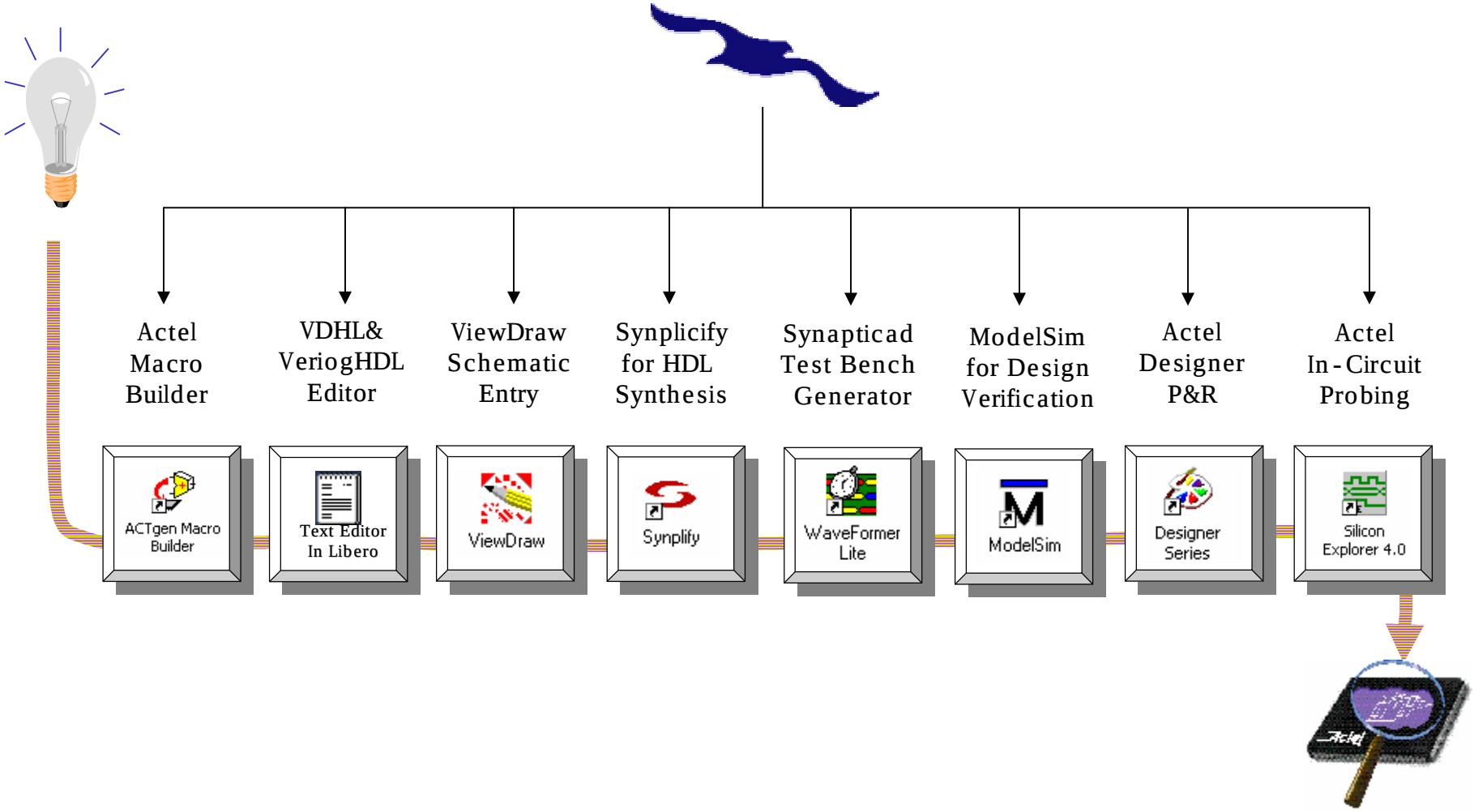
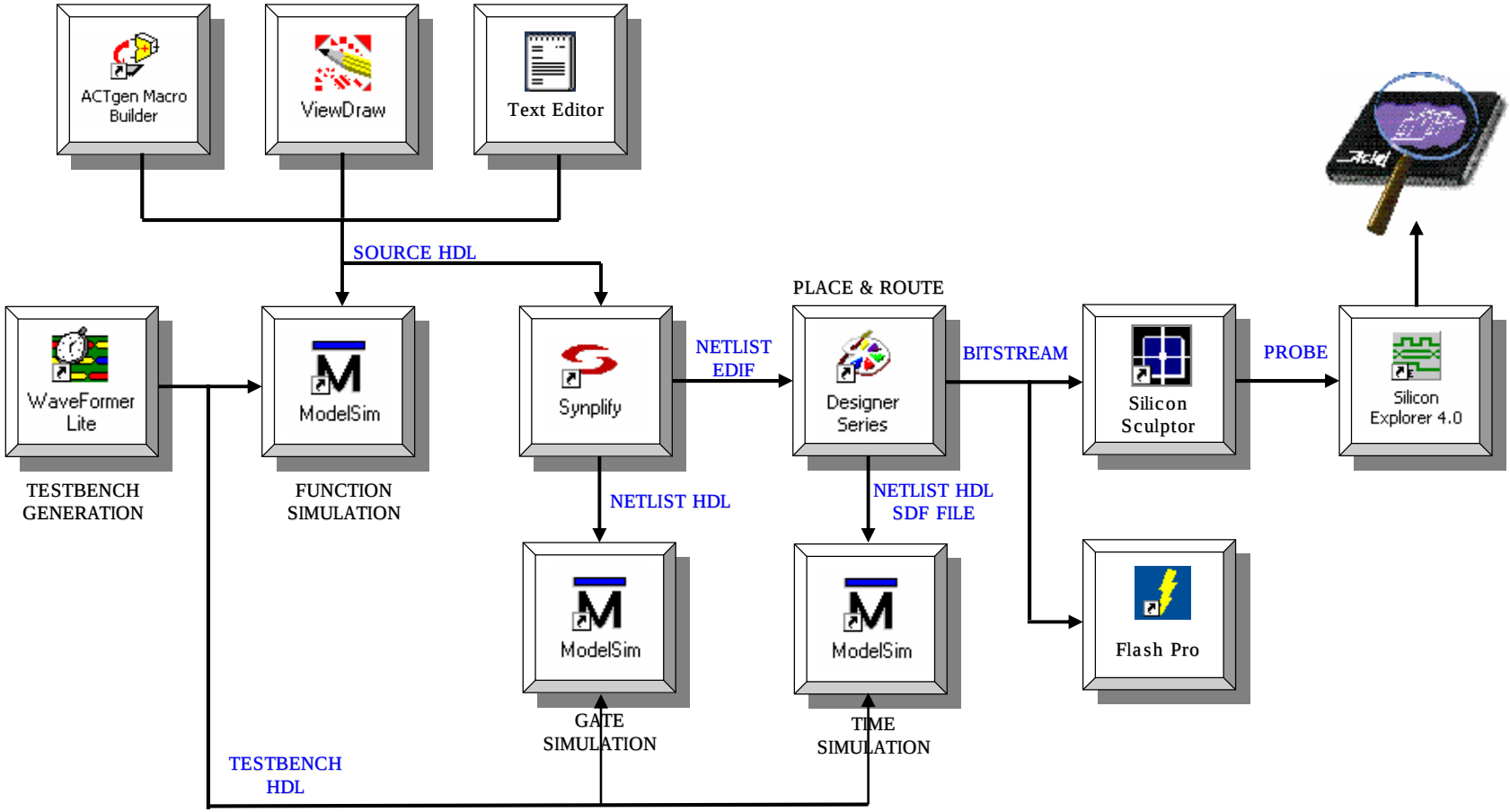
Libero Over view and Design Flow

Libero V5.2 SP1
Apr 2004
Ver52

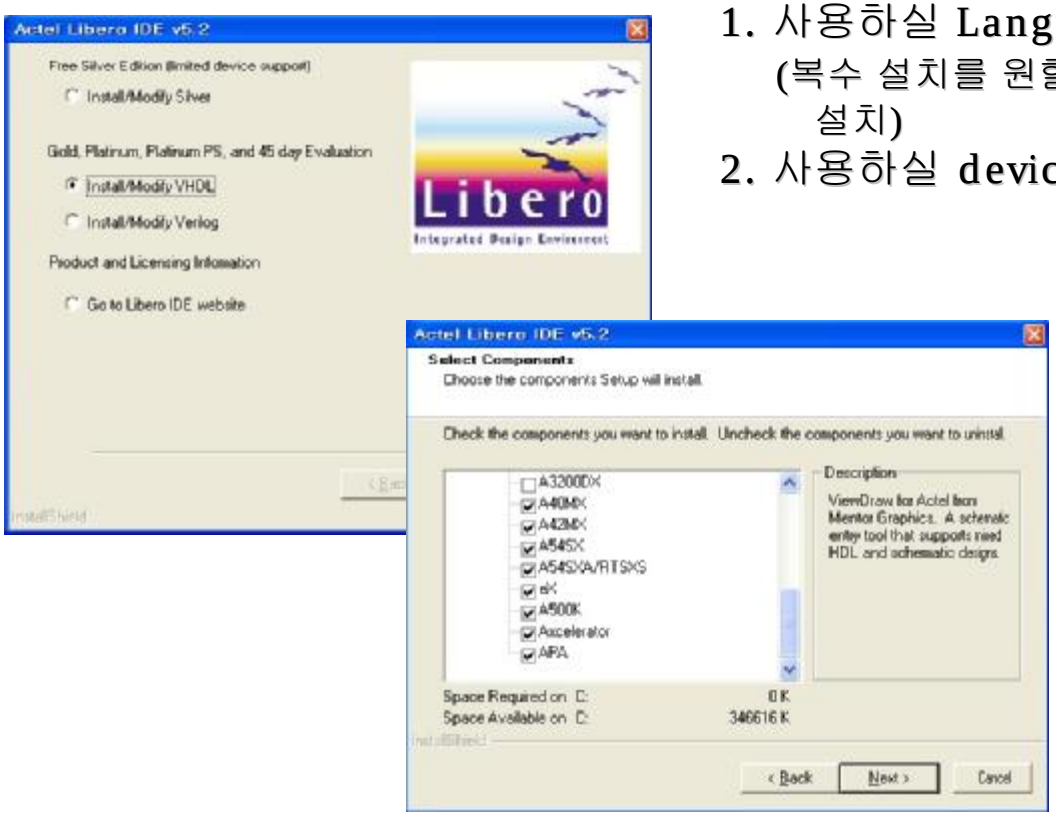
Libero Integrated Orchestra



Libero Design Flow



Software Install

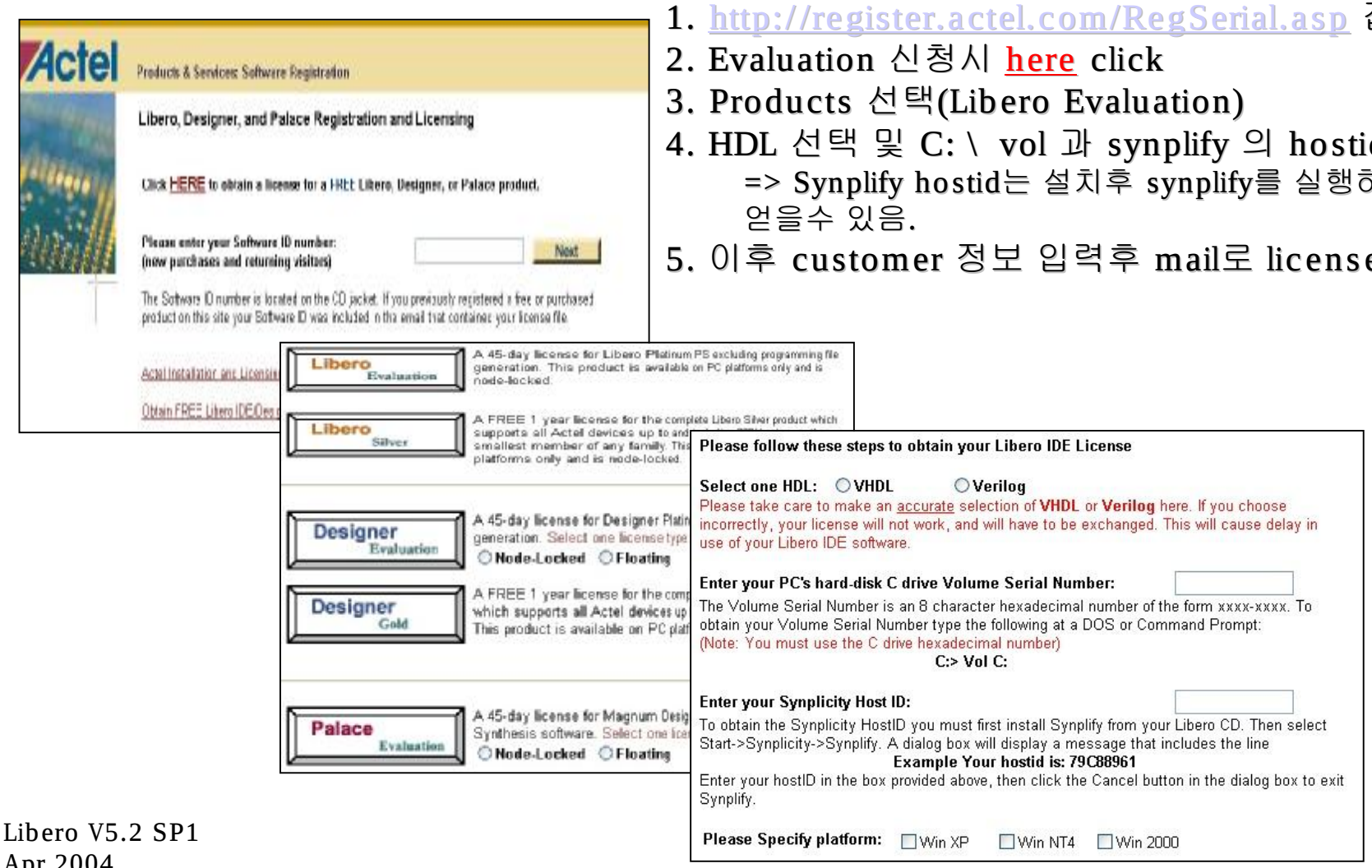


1. 사용하실 **Language**에 따라 **VHDL, Verilog** 선택
   (복수 설치를 원할 경우 한번 설치 후 다른 언어 선택 후 설치)
2. 사용하실 **device**에 맞는 **library** 선택

Libero Software Feature

Function	Tool	Vendor	Libero Silver	Libero Gold	Libero Platinum	Libero Platinum PS	Libero Platinum PS Eval	Designer Gold	Designer Platinum	Designer Eval
Device Support										
300,000 Gates and smaller		Actel	✦	✦	✦	✦	✦	✦	✦	✦
All Actel Devices		Actel	-	-	✦	✦	✦	-	✦	✦
Features Provided										
Project Manager	Libero Software	Actel	✦	✦	✦	✦	✦	-	-	-
Design Entry (HDL)	Libero Software	Actel	✦	✦	✦	✦	✦	-	-	-
Design Entry (Schematic)	ViewDraw AE[1]	Actel	✦	✦	✦	✦	✦	-	-	-
Synthesis	Synplify Lite[2]/ Synplify AE	Synplicity	✦[2]	✦[2]	✦	✦	✦	-	-	-
Simulation	Model*Sim* AE	Mentor Graphics	-	✦	✦	✦	✦	-	-	-
Physical Synthesis	PALACE	Magma	-	-	-	✦	✦	-	-	-
Test Bench Generator	WaveFormer Lite AE	SynaptiCAD	✦	✦	✦	✦	✦	-	-	-

License Request



1. http://register.actel.com/RegSerial.asp 접속
2. Evaluation 신청시 here click
3. Products 선택(Libero Evaluation)
4. HDL 선택 및 C:\ vol 과 synplify 의 hostid 입력
   => Synplify hostid는 설치후 synplify를 실행하면 얻을수 있음.
5. 이후 customer 정보 입력후 mail로 license 받음.

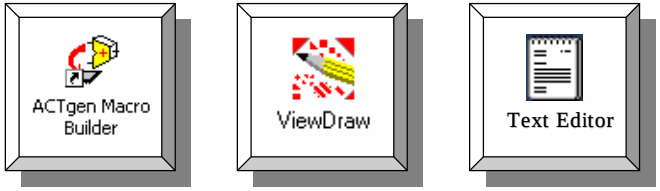
License Setup

1. Email로 받은 license.dat를 actel install directory에 copy

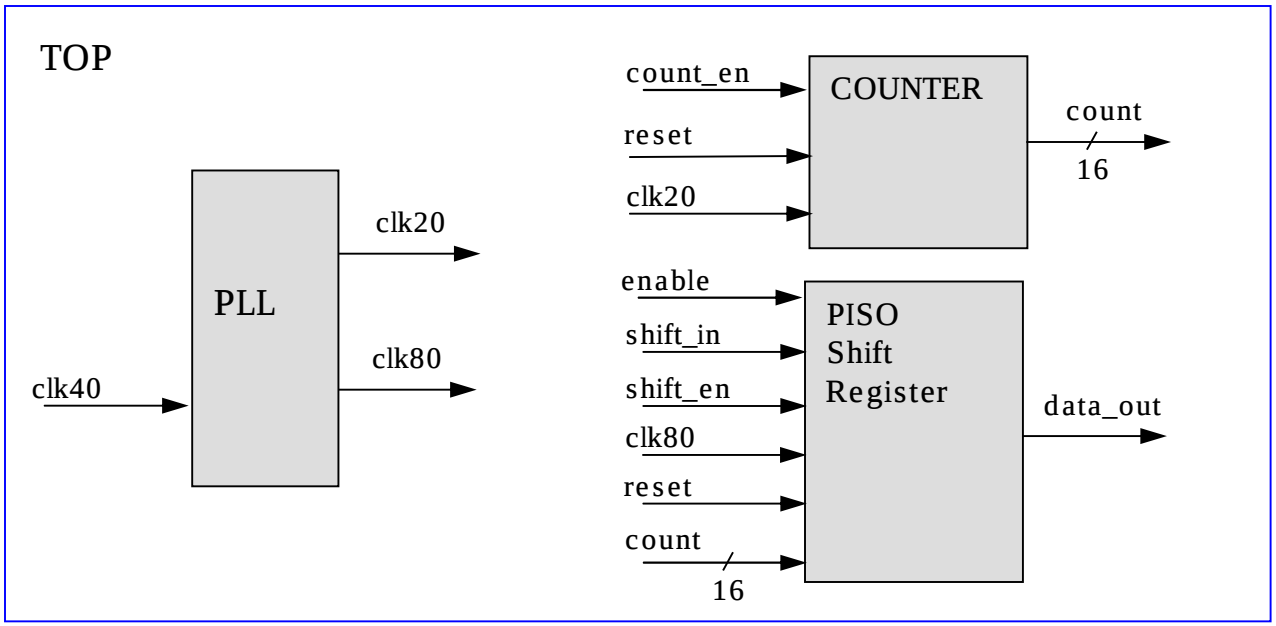
2. Os가 win2000, XP 일경우
   내컴퓨터 등록정보->고급->환경변수
   변수 LM_LICENSE_FILE 값 C:\<INSTALL DIR>\license.dat
   변수 SYNPLICITY_LICENSE_FILE 값 C:\<INSTALL DIR>\license.dat
   위 두항목을 추가.

3. OS 가 WIN98인 경우.
   libero V50 이상에서는 win98을 지원하지 않으므로 win2000, winXP로 사용할것.

Design Entry



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

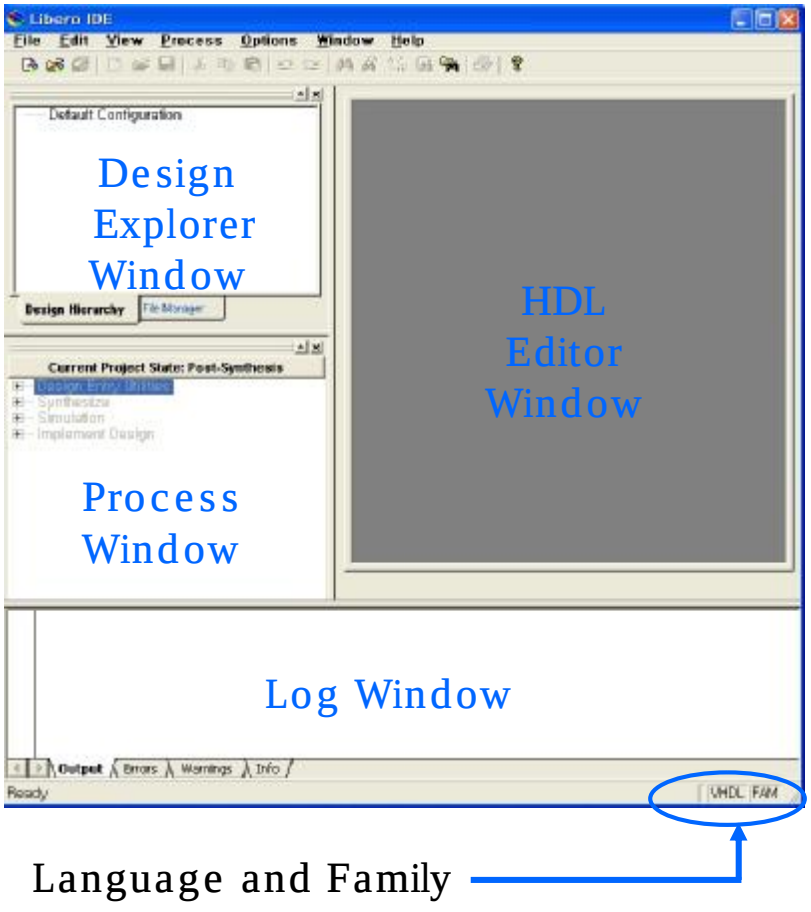
- Verilog와 VHDL93 text editor

Process Window

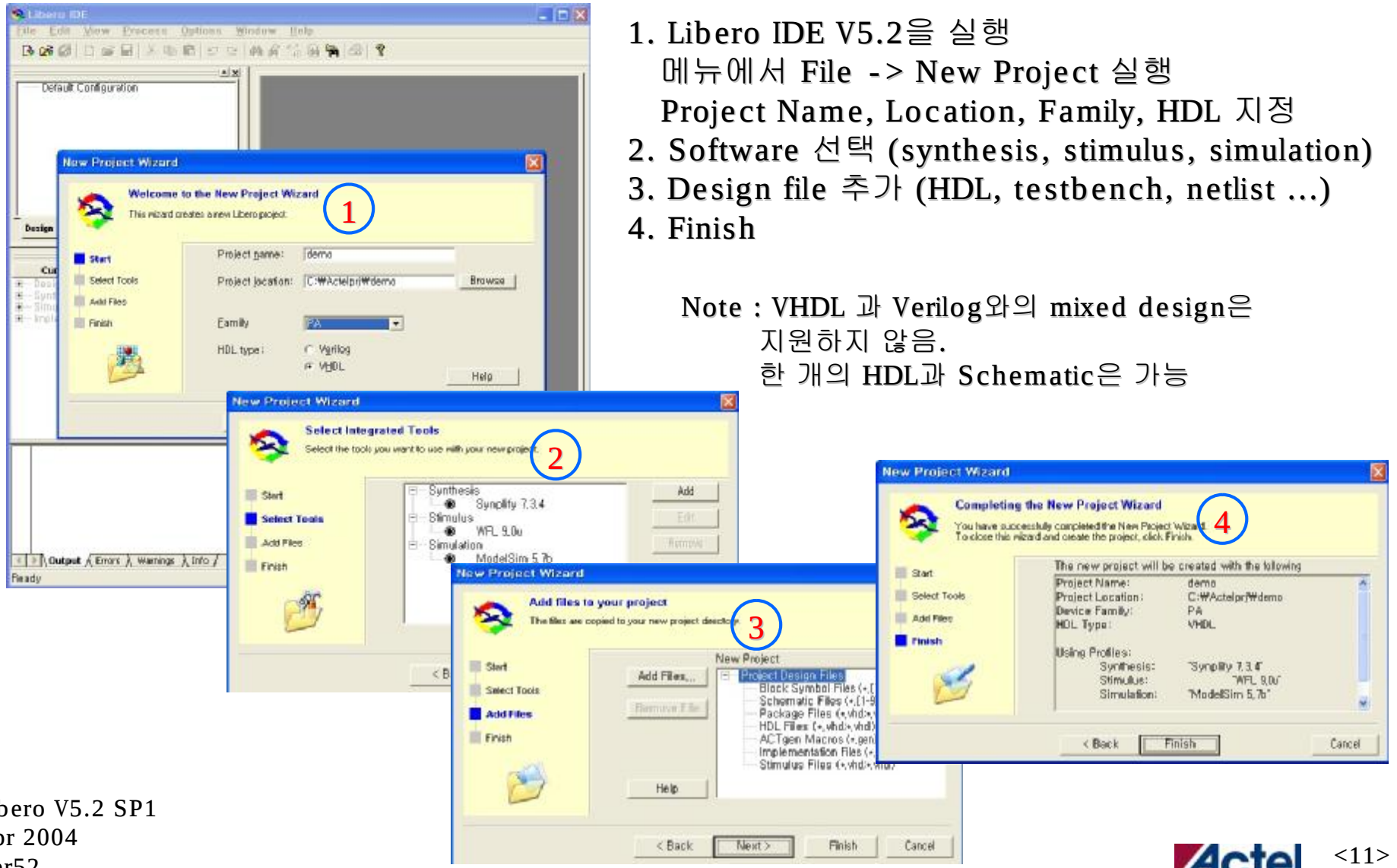
- Design을 위한 다른 tools로 연결

Log Window

- Software 연결상태및 메세지창



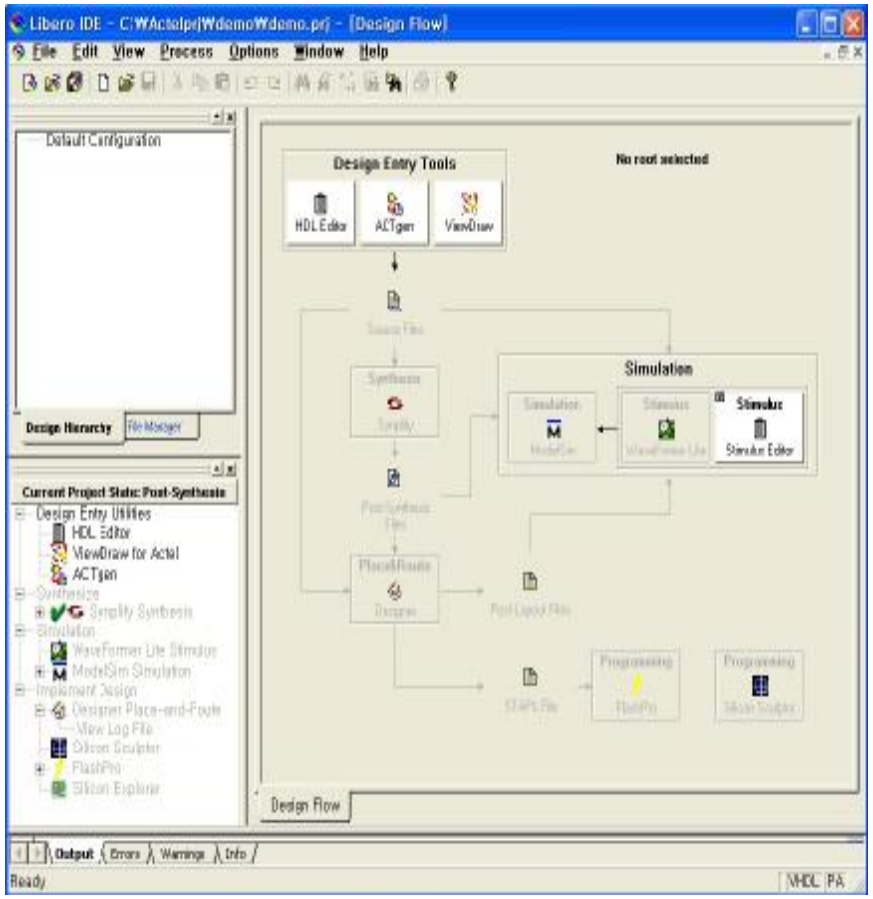
Create Project



1. Libero IDE V5.2을 실행
   메뉴에서 File -> New Project 실행
   Project Name, Location, Family, HDL 지정
2. Software 선택 (synthesis, stimulus, simulation)
3. Design file 추가 (HDL, testbench, netlist ...)
4. Finish

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

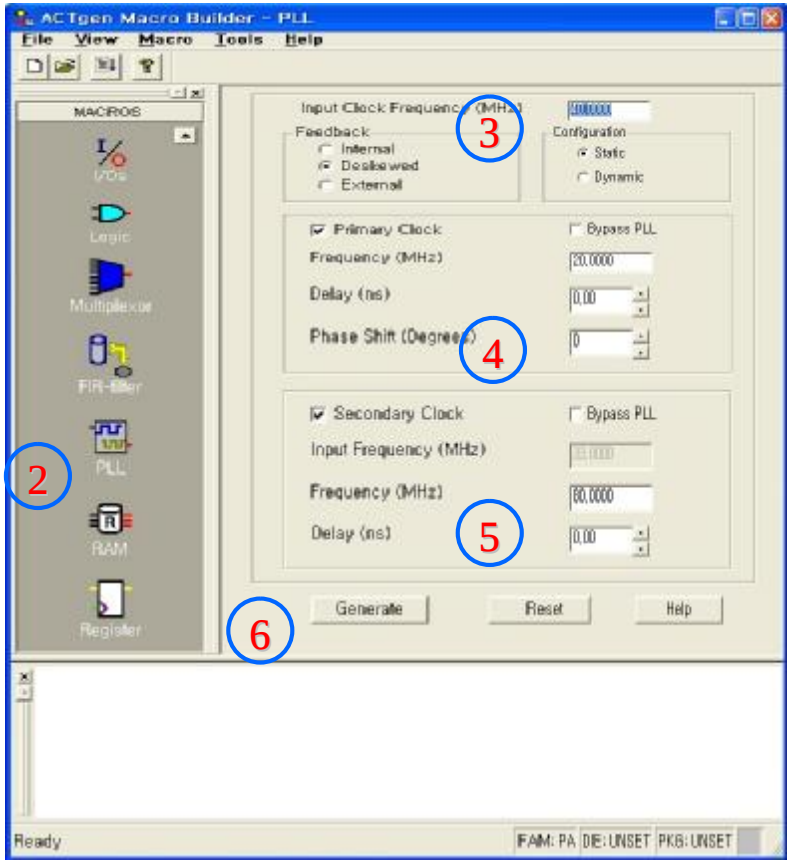
Design Flow

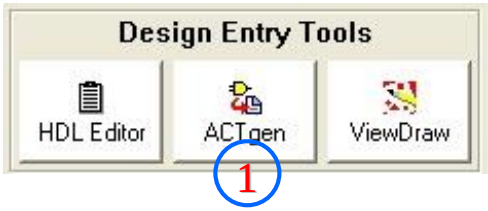


< 단계별로 실행 가능한 기능을 보여줌>

1. Design Entry Tools
   - => HDL Text Editor
   - => ACTgen Macro Builder
   - => Schematic View draw
2. Synthesis
3. Simulation
   - => Modelsim
   - => WaveFormer Lite
   - => Stimulus Editor
4. Place & Route
5. Programming
   - => Flash Pro
   - => Silicon Sculptor

Create Design – PLL Block



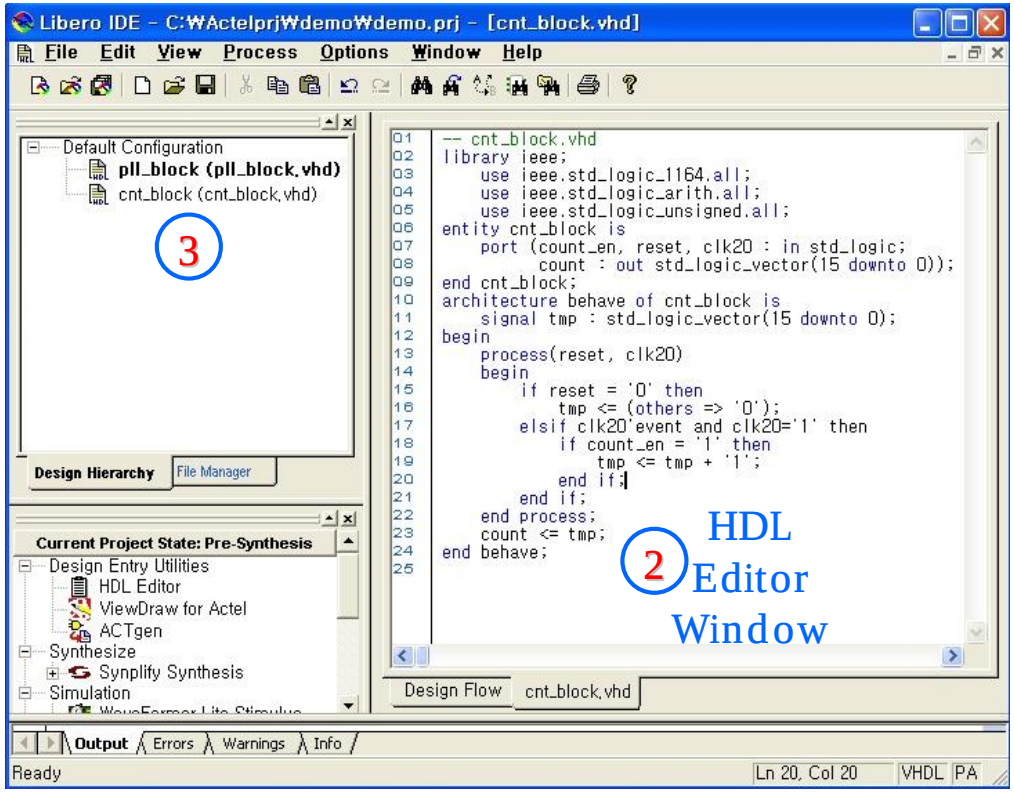


1. Design Flow 에서 actgen 을 선택
   => 선택후 name : pll_block 입력
2. 왼쪽 MACROS 메뉴에서 PLL 선택

< Parameter Setting >

3. Input clock 40Mhz.
4. Primary clock output 20Mhz
   => Delay : 기본 250ps 단위로 -4ns~8ns 지정가능
   => Phase Shift : 90, 180, 270 위상 변환 가능
5. Secondary Clock output 80Mh
   => Delay : 기본 250ps 단위로 -4ns~8ns 지정가능
6. Generate Pll_block.vhd file

Creat Design – Count Block



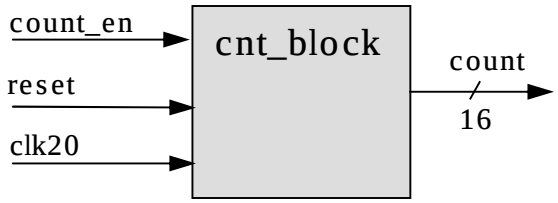
1. Design Flow 에서 HDL Editor 을 선택
   => 선택후 name : cnt_block 입력
2. HDL Editor Window 에서 cnt_block coding
3. 저장시 Design hierachy 에 cnt_block 추가됨

<기존 design import 방법>
File => import files.. 메뉴에서 기존의 Design file 을 import 가능

Creat Design – Count VHDL Design

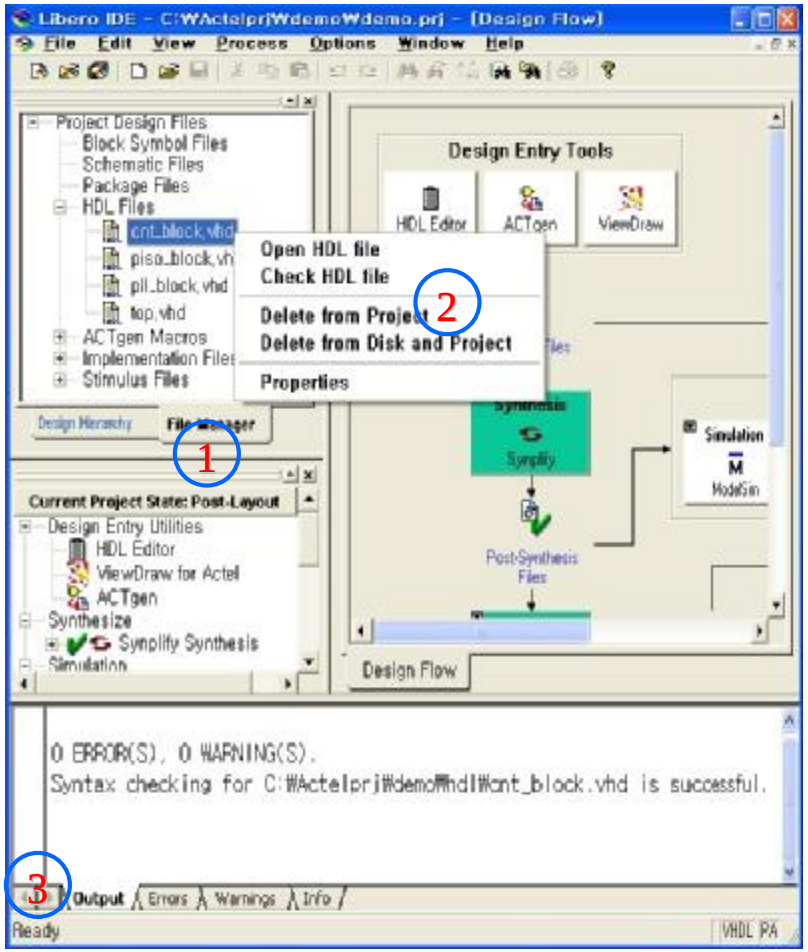
```
-- cnt_block.vhd
library ieee;
   use ieee.std_logic_1164.all;
   use ieee.std_logic_arith.all;
   use ieee.std_logic_unsigned.all;
entity cnt_block is
   port (count_en, reset, clk20 : in std_logic;
         count : out std_logic_vector(15 downto 0));
end cnt_block;
architecture behave of cnt_block 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



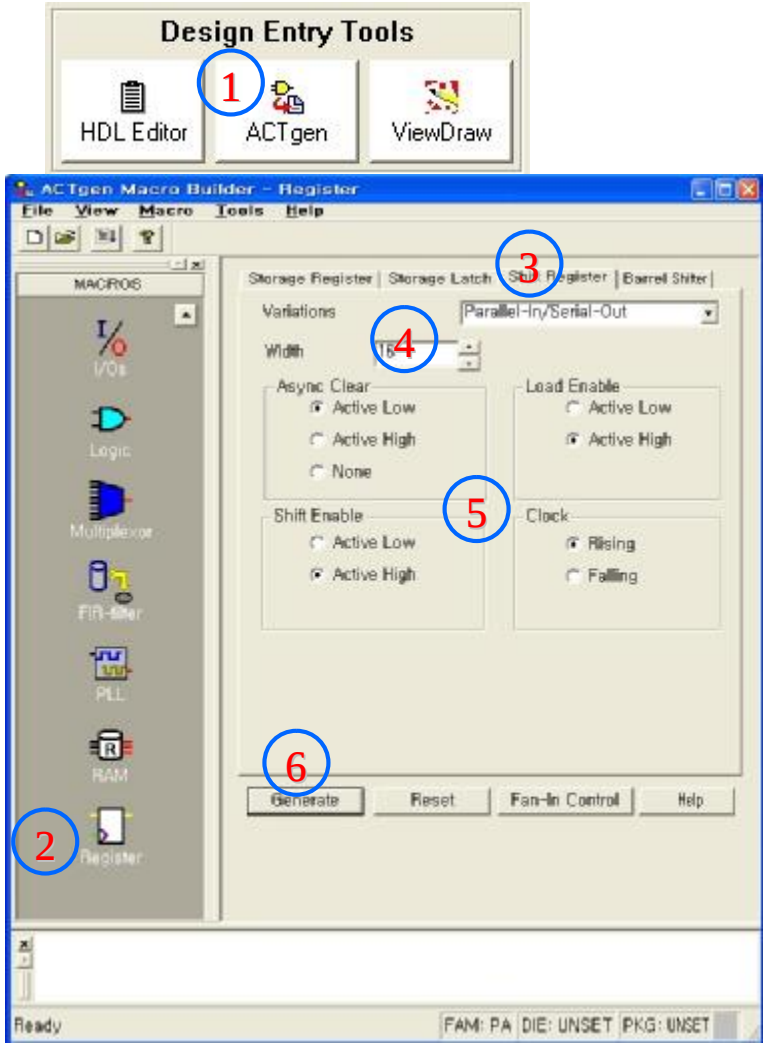
3. Achitecture 구문
   16bit counter with count enable
   and asynchronous reset

Creat Design – Syntax check



1. Coding 한 design의 syntax check를 위해 File Manager 로 이동
2. HDL Files에서 원하는 vhdl 파일을 선택 우클릭을 이용하여 Check HDL file 을 선택
3. 아래 output 창에서 syntax check 결과 확인

Creat Design – PISO Block

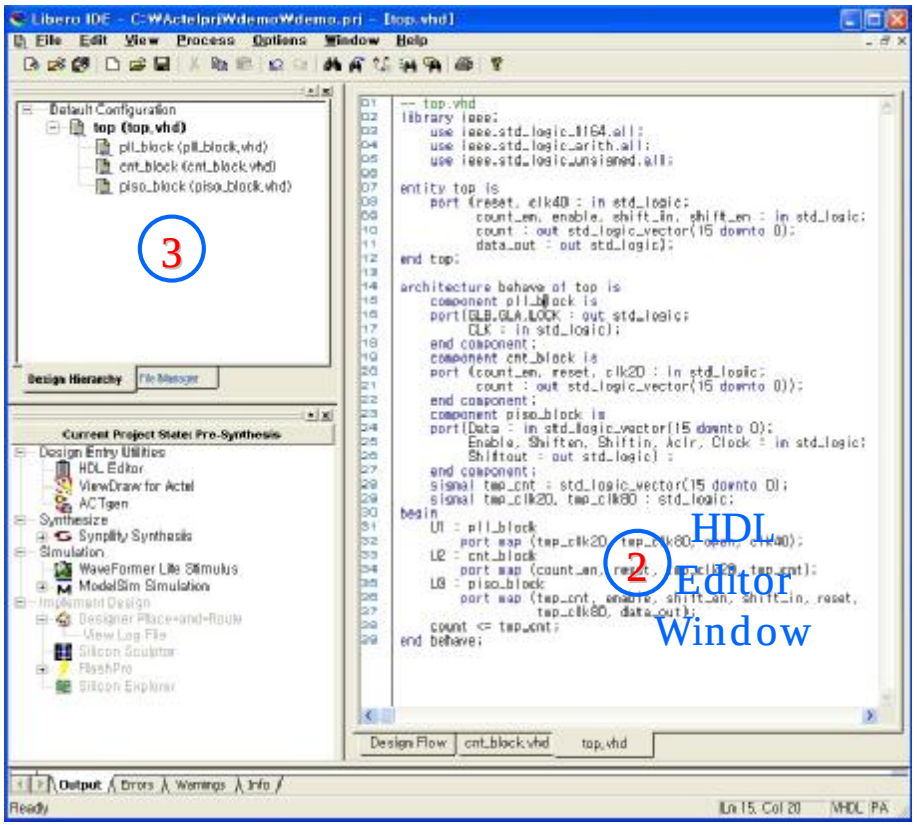


1. Design Flow 에서 actgen 을 선택
   => 선택후 name : piso_block 입력
2. 왼쪽 MACROS 메뉴에서 Register 선택

< Parameter Setting >

3. Register의 종류선택 => Shift Register
4. Variations => Parallel input to serial output register
5. 추가적인 옵션 선택
   => Clear, Enable, Load, Clock 속성 명시
6. Generate Pll_block.vhd file

Creat Design – TOP BLOCK



1. Design Flow 에서 HDL Editor 을 선택
   => 선택후 name : top 입력
2. HDL Editor Window 에서 top design coding
3. 저장시 Design hierachy 에 top design 추가됨

<기존 design import 방법>
#File => import files.. 메뉴에서 기존의 Design file 을 import 가능
#Note : design 의 상하 관계를 인식하여 자동으로 계층구조 표시됨

Libero V5.2 SP1
Apr 2004
Ver52

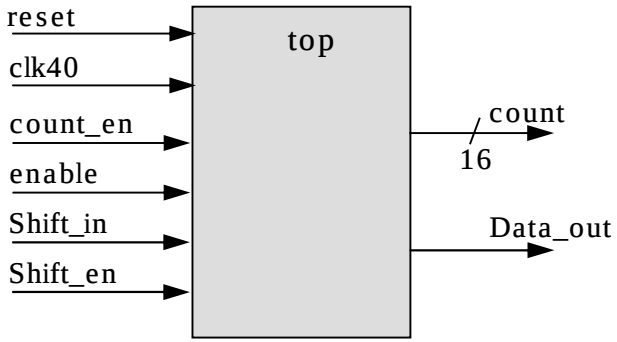
Creat Design – Top VHDL Design

```
-- top.vhd
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;
```

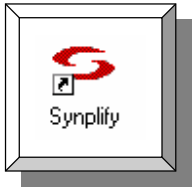
1. Library 선언구문
2. Entity 구문
   Top design의 입출력 signal 선언구문



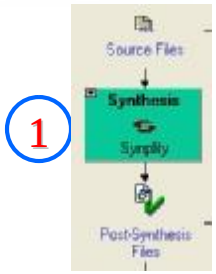
3. Architecture 구문
4. Component 선언구문
   Sub block pll_block, cnt_block, piso_block 선언
5. Component port map 구문
   Sub block간의 signal 연결

=> Top.vhd file도 syntax check를 통해 coding 검증을 함.

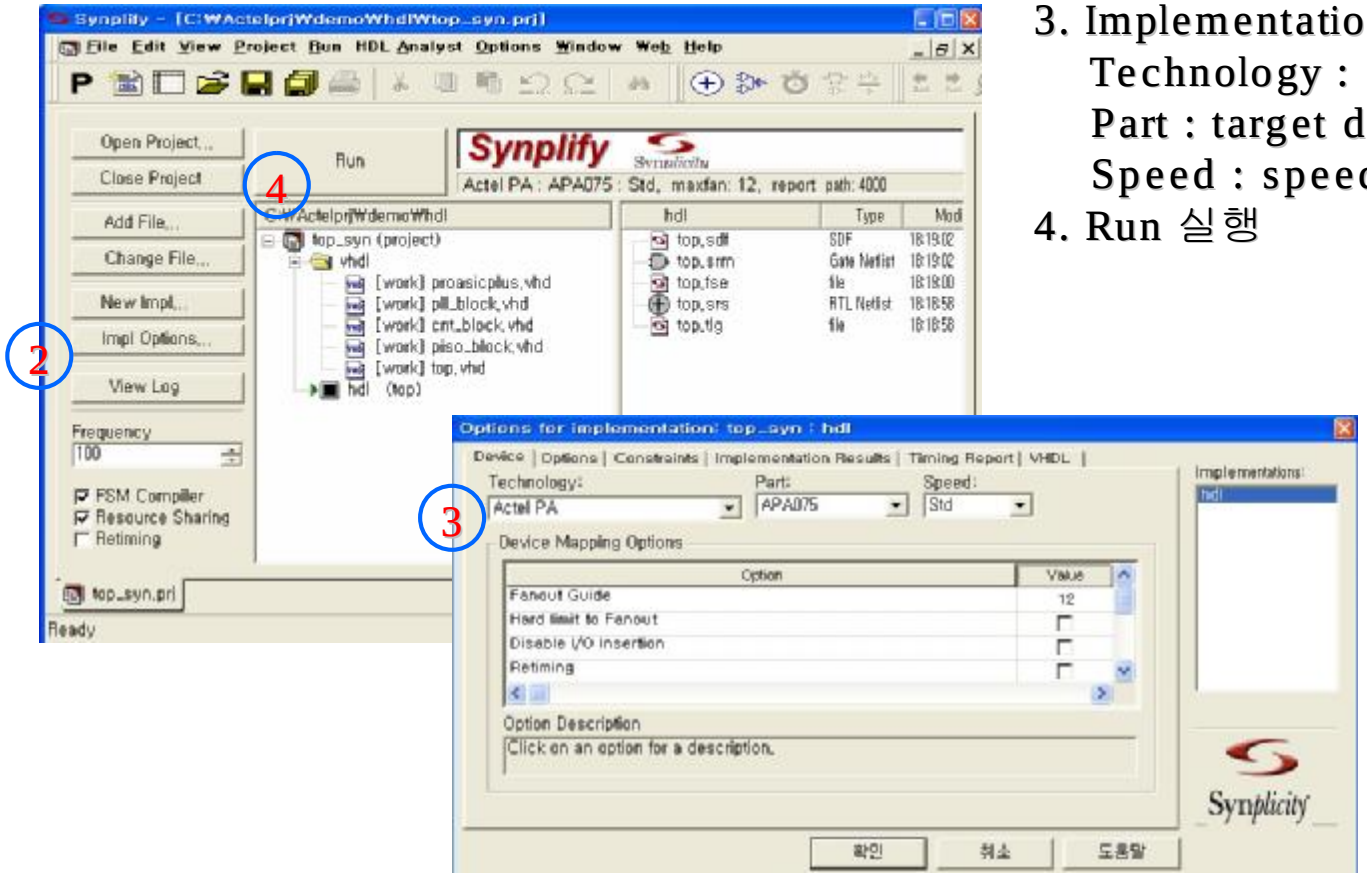
Synthesis - Synplify



Synplify – Synthesis

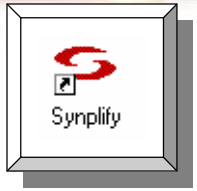


1. Design Flow에서 Synthesis 선택
   또는 process window 에서 Synthesize 선택



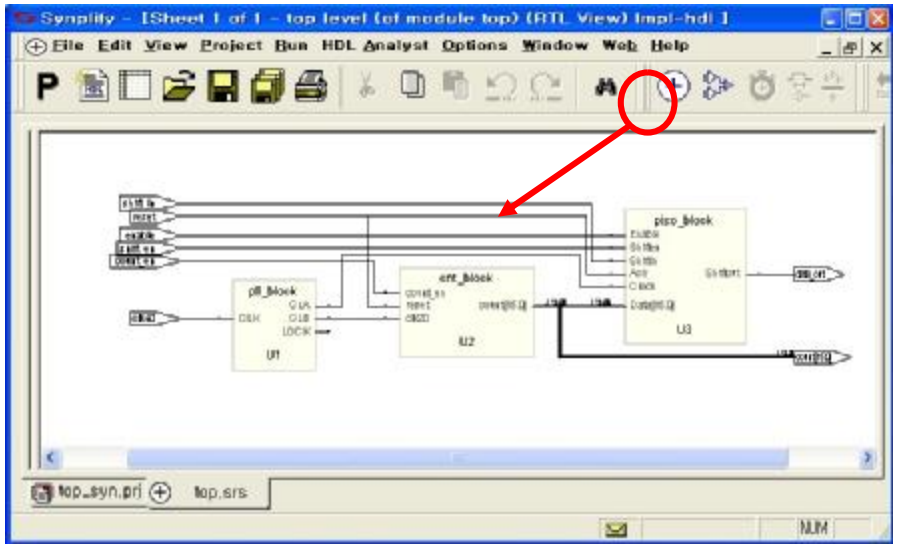
3. Implementations options
   Technology : actel target library 선택
   Part : target device 선택
   Speed : speed grade 선택
4. Run 실행

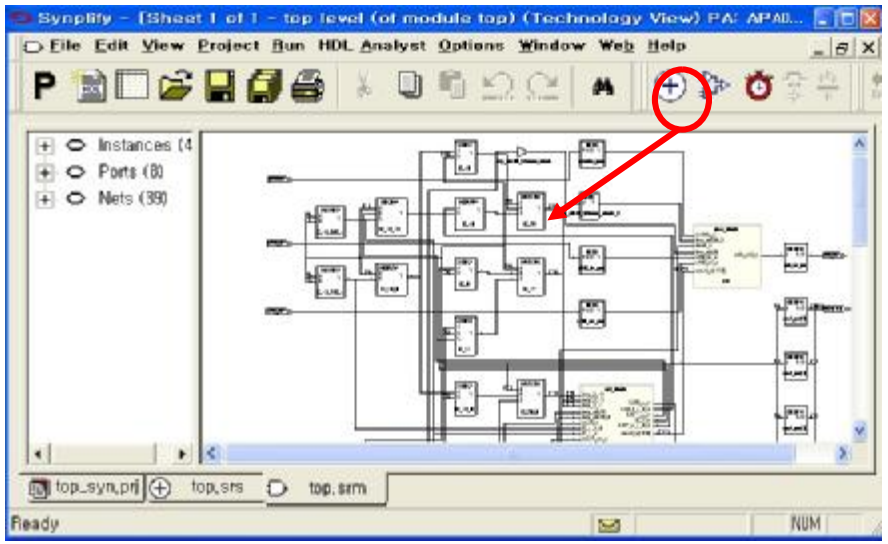
Synplify – Viewer



1. 합성된 schematic의 viewer
2. RTL view 는 source 를 이론적인 형태로 schematic으로 표시
3. Technology view 는 actel library로 변환된 netlist를 schematic으로 표시
   HDL로 설계된 design을 검토하기에 RTL view는 자주 사용됨.

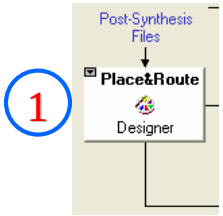
<NOTE> Schematic viewer는 Synplicity 정품에서만 지원가능함. Actel OEM은 지원하지 않음

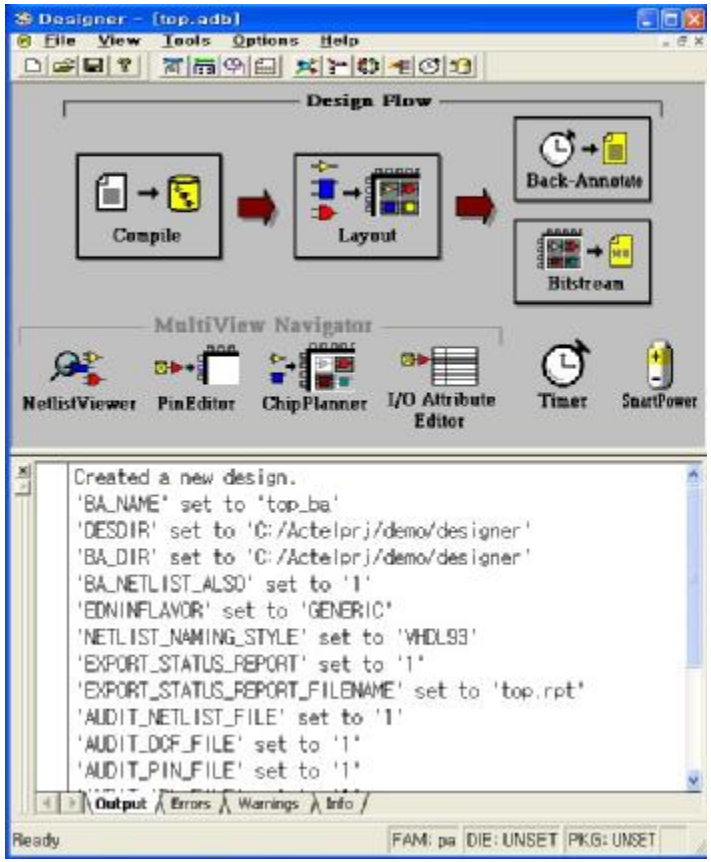




Layout – Designer Series

Designer Series – Compile 1



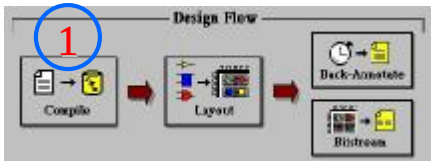


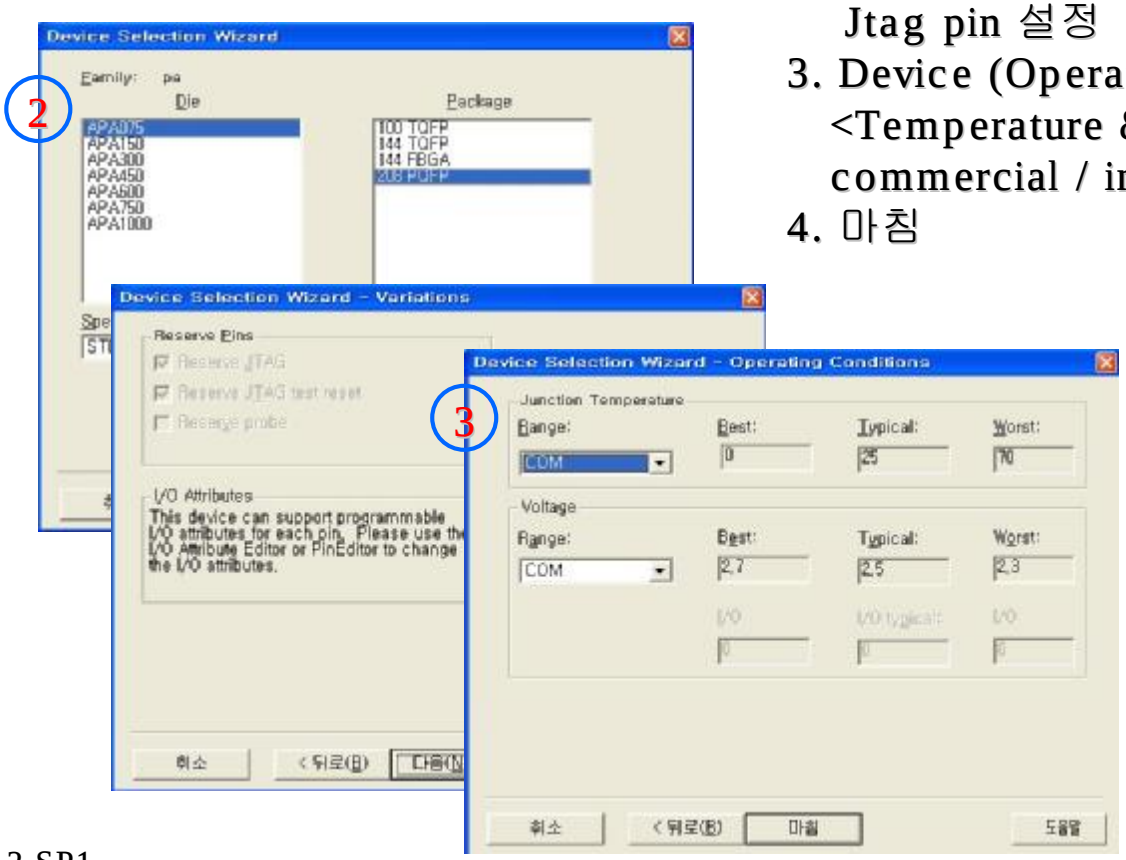
< Place & Route 과정>
1. Design Flow에서 Please & Route 선택 실행

* Designer s/w 에서의 Design Flow
<Compile> device 선택 및 logic size 분석
<Layout> design place & route
<Bitstream> program하기 위한 stp/bit file 생성
<Back-Annotate> Timing sim을 위한 파일 출력

* Sub menu
<Netlist Viewer> gat level circuit 분석
<PinEditor> pin assign
<ChipPlanner> layout 후 내부 cell의 위치정보 확인 및 변경 가능
<I/O Attribute Editor> I/O의 특성 부여
<Timer> design timing 정보
<SmartPower> power 소모량 측정

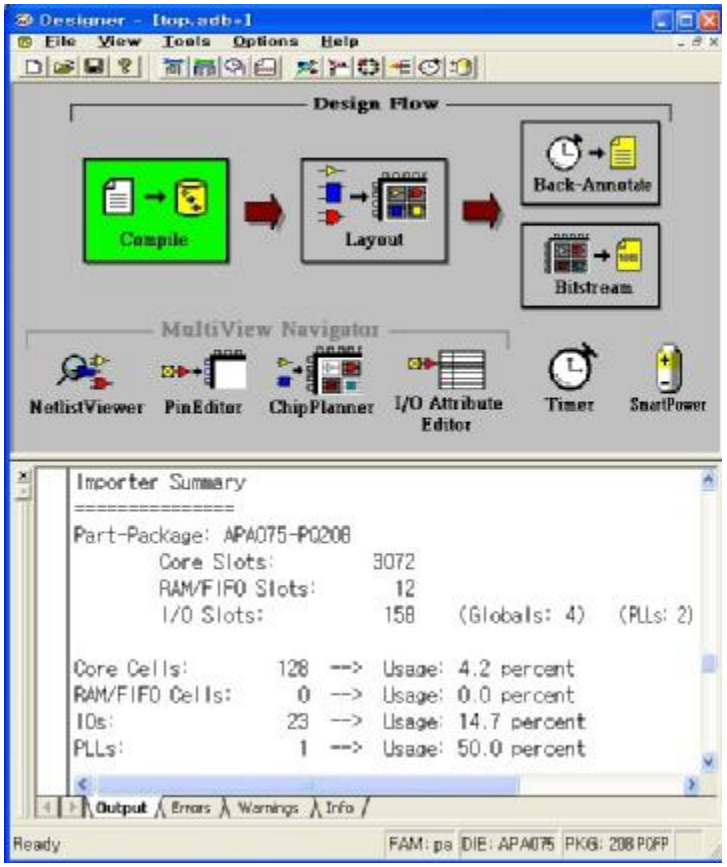
Designer Series – Compile





1. Compile 선택
2. Device (gate,package,speed,voltage)설정
   Jtag pin 설정
3. Device (Operating Conditions) 설정
   <Temperature & Voltage>
   commercial / industrial / military
4. 마침

Designer Series – Compile



1. Compile 완료시 device에 사용량 체크

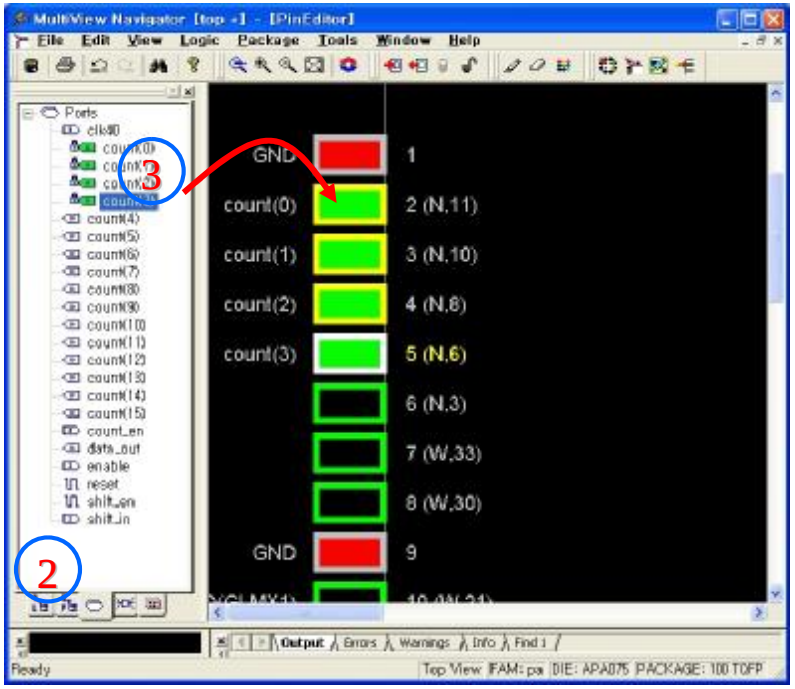
   Core Cells : 순수 사용할 수 있는 gate
   RAM/FIFO Cells : Memory block에 대한 사용량
   IOs : 사용된 I/O 수
   PLLs : 사용된 PLL 개수

<추가적인 check point>

global resources : 네개의 external global 사용된 signal

외부 GL pin에 할당을 하여도 내부적으로 사용된 fanout 에 의해서 global 지정은 다르게 나타날수 있음.

Designer Series – PinEdit

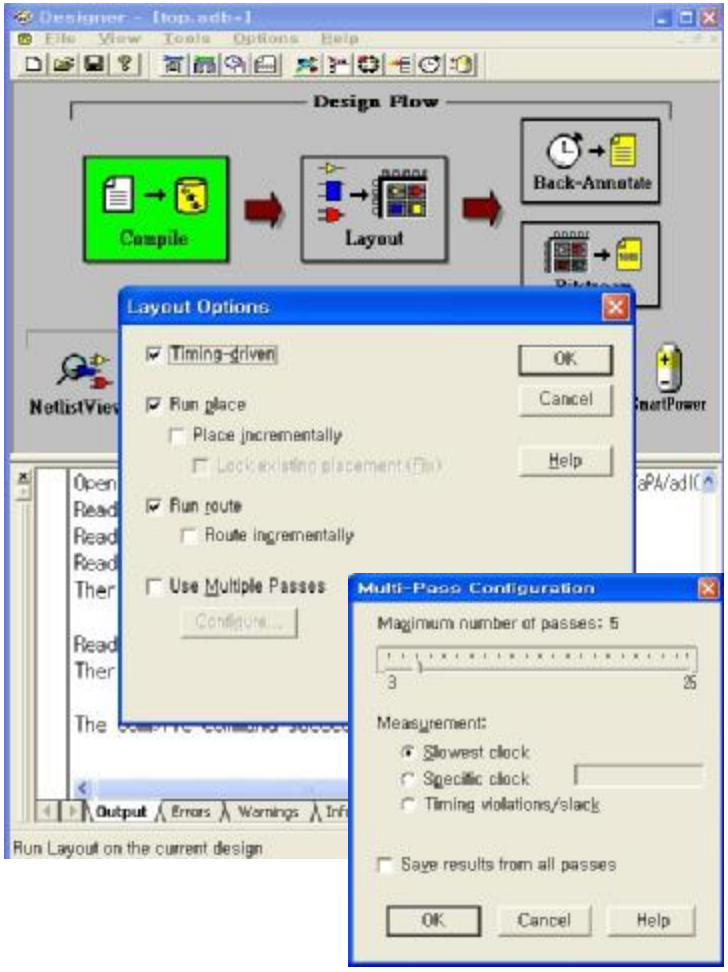


1. PIN ASSIGN 을 위한 방법
   # pineditor를 이용
   # pin file을 이용
   # I/O attribute editor 이용
2. MultiView Navigator 왼쪽 항목에서 PORT 선택
3. Ports list에서 원하는 pin을 선택후 drag & drop방식으로 오른쪽 창 pin 에 assign

pin(gcf) file을 이용하는 방법
pin file을 이용하기 위해서는 먼저 위 pin assign방식을 한번 실행하고 assign 된 정보를 pin file로 출력한 뒤 사용가능
=> Antifuse Device : xxx.pin
=> Flash Device : xxx.gcf

Designer main 화면에서
<pin file 출력> file->export->constraint files->gcf 출력
<pin file 입력> file->import source file->add 후 gcf 입력

Designer Series – Layout



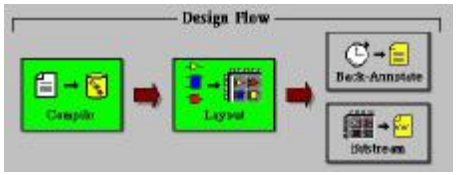
1. Layout 선택
2. Layout options

<Timing-Driven> constraint file(gcf)에 명시된 조건에 따라 layout을 실행함.
<Run Place> place를 재실행
<Incrementally> 기존의 place 된 정보를 유지
<Lock Existing Placement (fix)> 고정시킨 cell에 대해서 유지

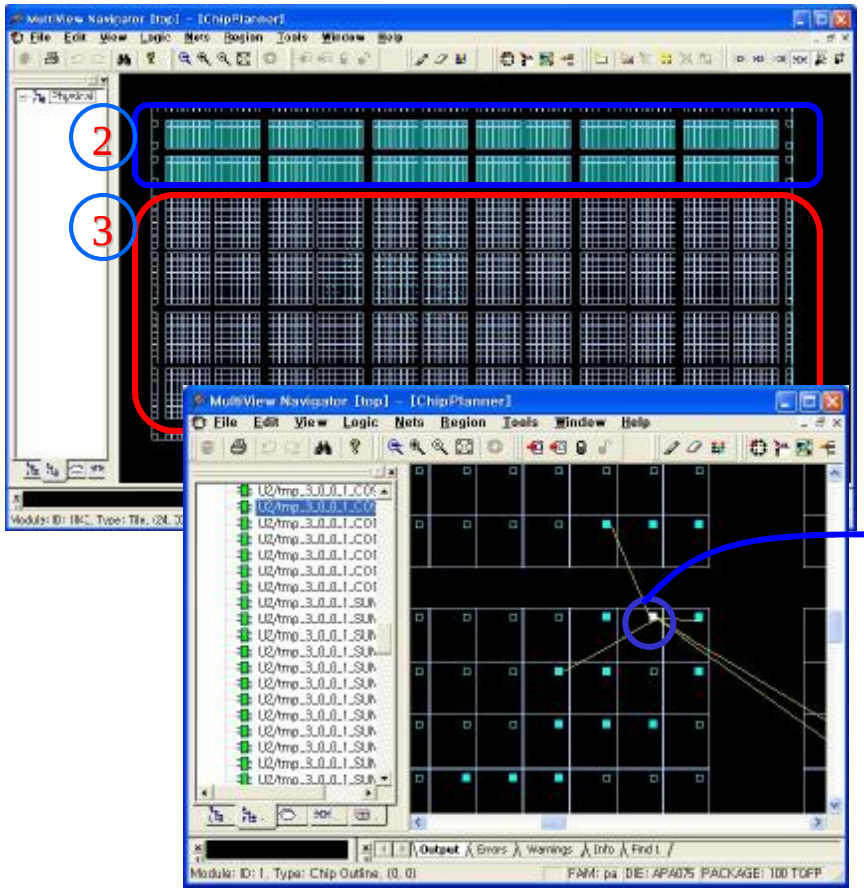
<Run Route> route를 재실행
<Incrementally> 기존의 route 된 정보를 유지

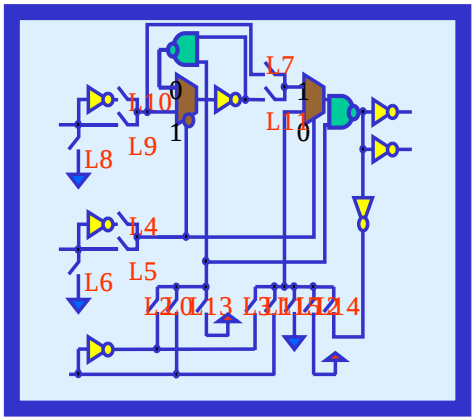
<Use Multiple Passes> layout 시 시작 시점(Seed)을 다르게 하여 전혀 다른 layout 결과를 가져와 성능이 좋게 나온 결과를 채택함.
=> layout의 횟수를 지정
=> specific clock에 성능향상을 하기 위한 clock 지정

2. Ok



Designer Series – ChipPlanner



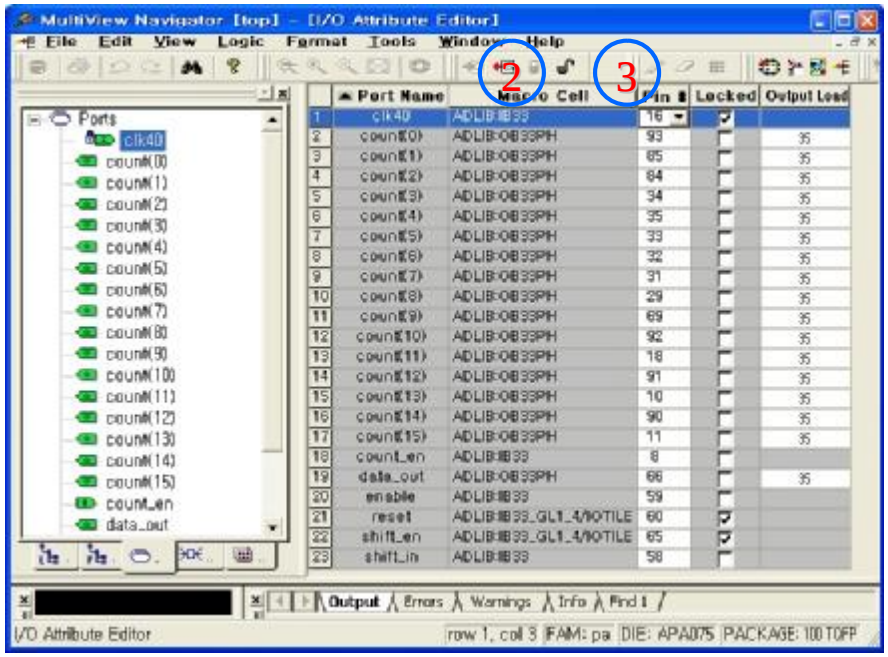


1. ChipPlanner 열기
2. Memory block
   APA 075/150 : 위에만 Memory 위치
   APA 300이상 : 위 아래 두곳에 Memory 위치
3. Core cell(tiles)
4. Core cell의 내부 구조

block 단위로 location 위치 지정 가능

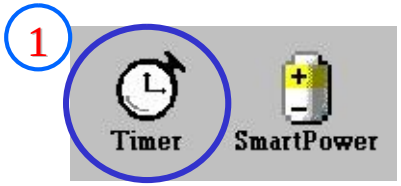
Libero V5.2 SP1
Apr 2004
Ver52

Designer Series – I/O attribute

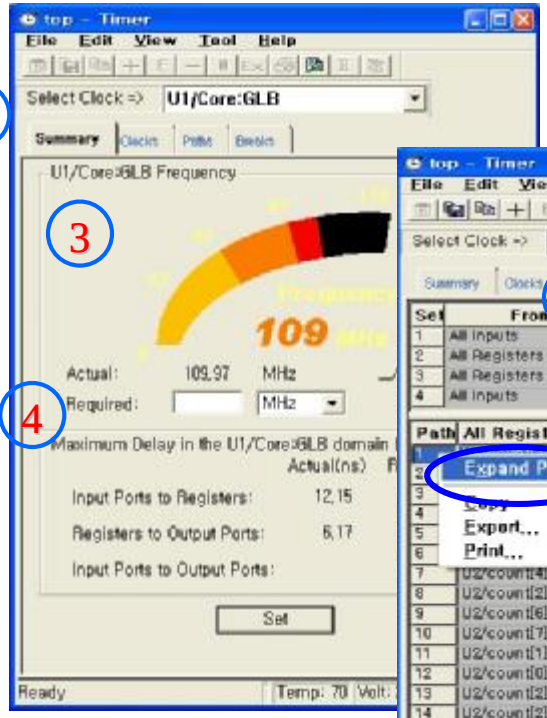


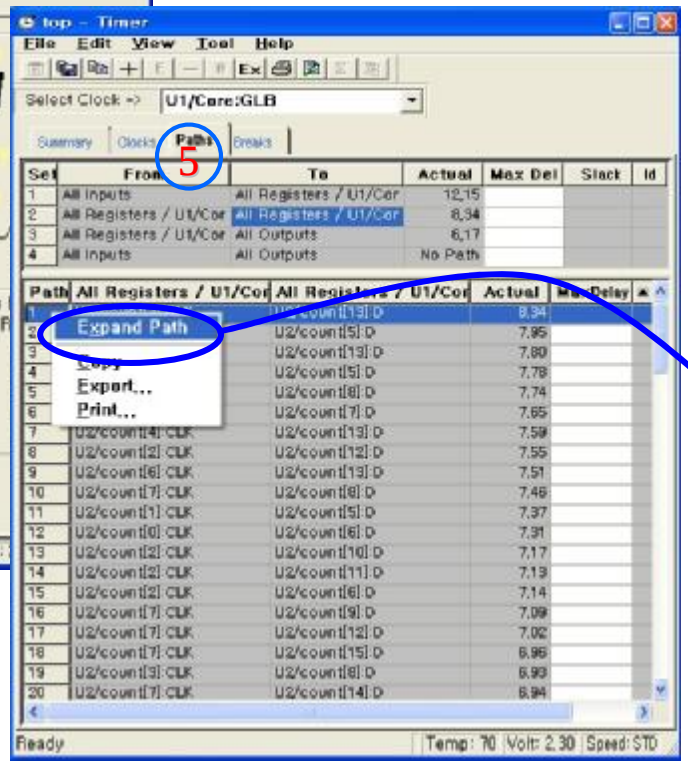
1. I/O attribute editor 선택
2. IO 별 사용 macro cell 표시
3. Pin # 지정 가능

Designer Series – Timer

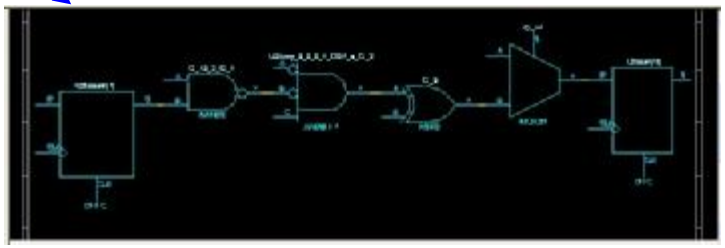


1. Layout 후 timer 선택
2. Select Clock => design 에 사용된 모든 clock 표시
   timming 분석이 필요한 clock 선택
3. 선택된 clock에 대해서 최대 performance 표시
4. 요구 clock 성능 지정 가능

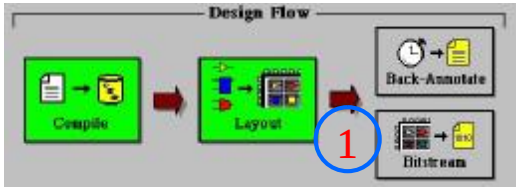


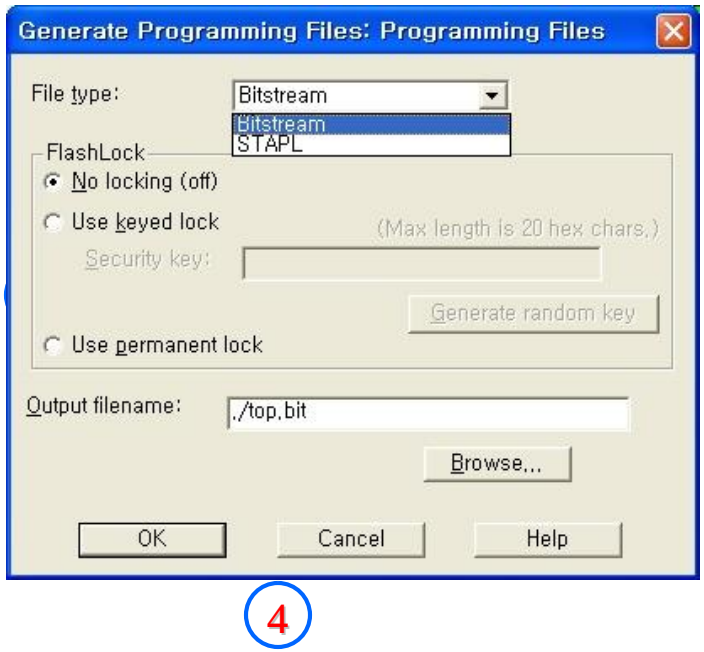


5. Paths 선택하여 timing
   분석 가능
=> 각각의 path 별로 timing 분석
1) input – register
2) register – register
3) register – output
4) input - output



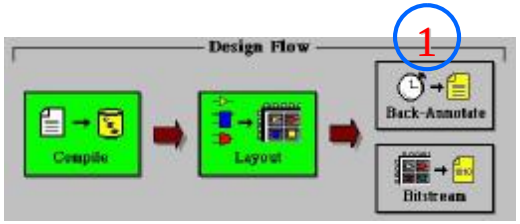
Designer Series – Bitstream

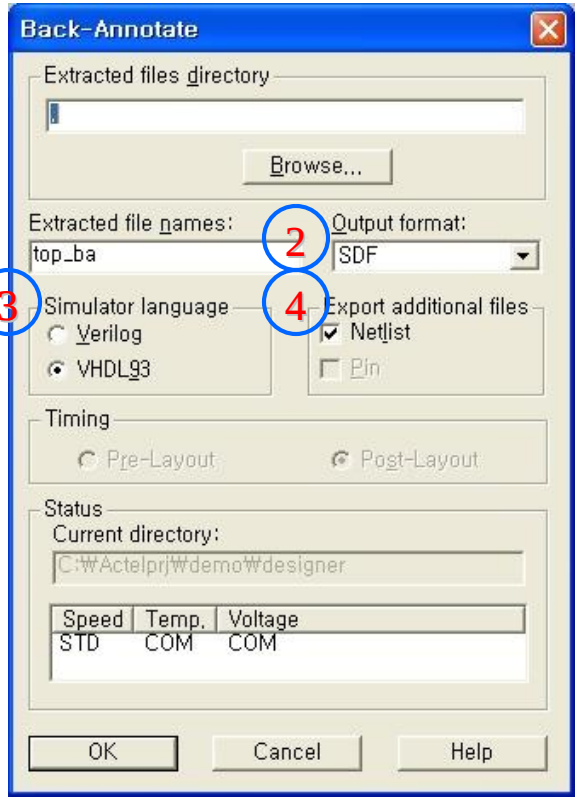




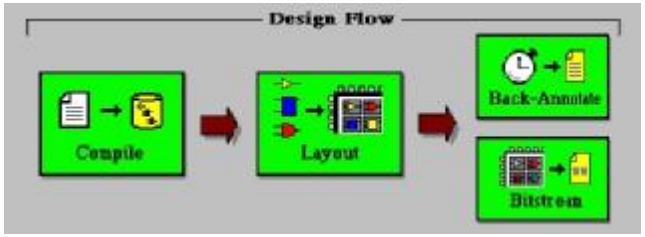
1. Bitstream icon 선택
2. File Type 선택
   BitStream : silicon sculptor를 이용한 programming
   STAPL : flash pro and flahs pro lite 를 이용한 programming
3. FlashLock 입력시 program, erase 불가
   => Use Keyed Lock 선택시 device에 따라 19~65 자리의 hex 값지정가능
   => Use permanent lock 선택시 random 값 지정
4. Filename 지정
5. Programming file 생성

Designer Series – Backannotation

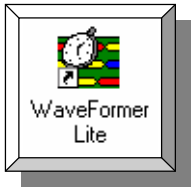




1. Back-Annotate 선택
2. SDF(Standard Delay Format) 출력
3. Timing simulation을 위한 HDL 선택
   Verilog / VHDL93
   # libero에서는 VHDL87 은 사용하지 않음
4. Pin file / Netlist 출력 가능
5. 모든 과정이 끝난 후 저장
   => .adb 파일로 저장되며 이 파일하나로 design의 모든 정보를 내포하고 있어 관리 요망.



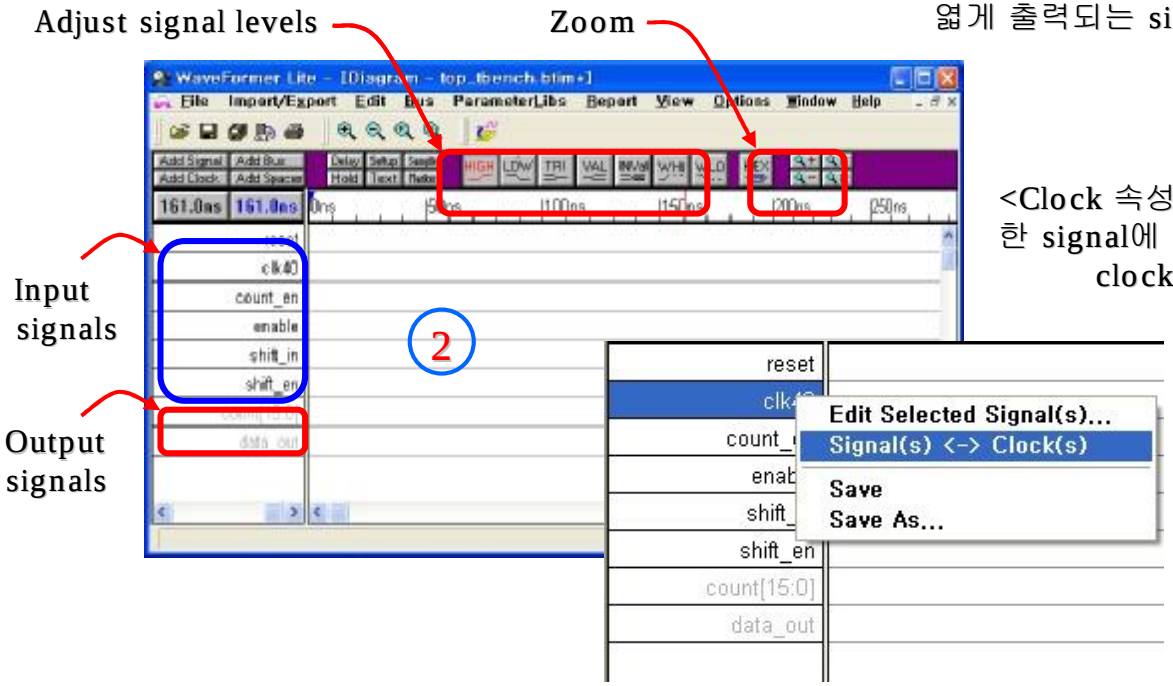
Stimulus – Wave Former Lite



Waveformer Lite

1. Design Flow에서 Stimulus 실행
2. WaveFormer Lite의 초기 화면

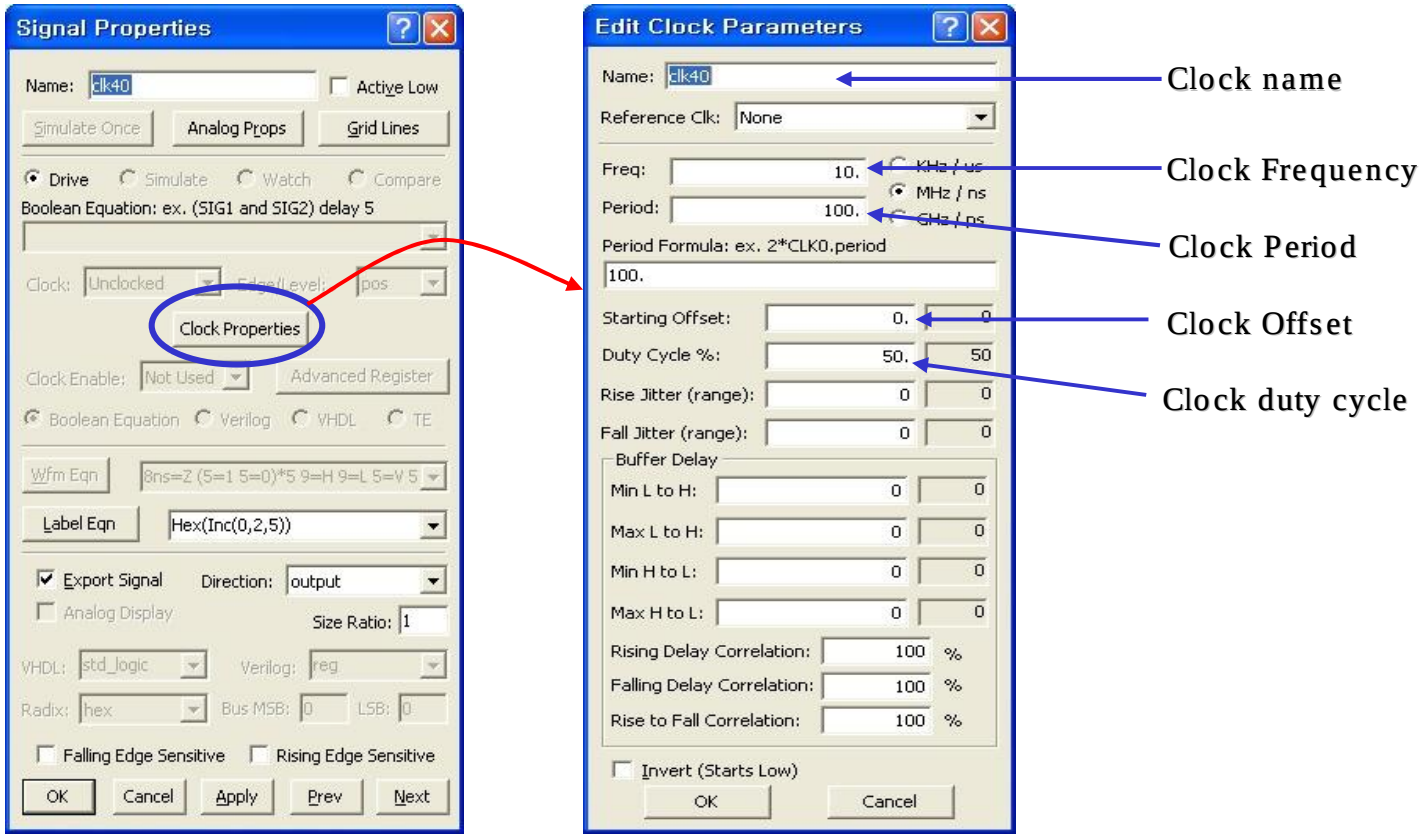
<note> signal list에서 진하게 출력되는 signal 은 input을 나타내고
엷게 출력되는 signal은 output을 나타낸다.



<Clock 속성 부여>
한 signal에 clock의 속성을 부여하기 위해서
clock을 선택 우클릭을 이용하여 속성 부여

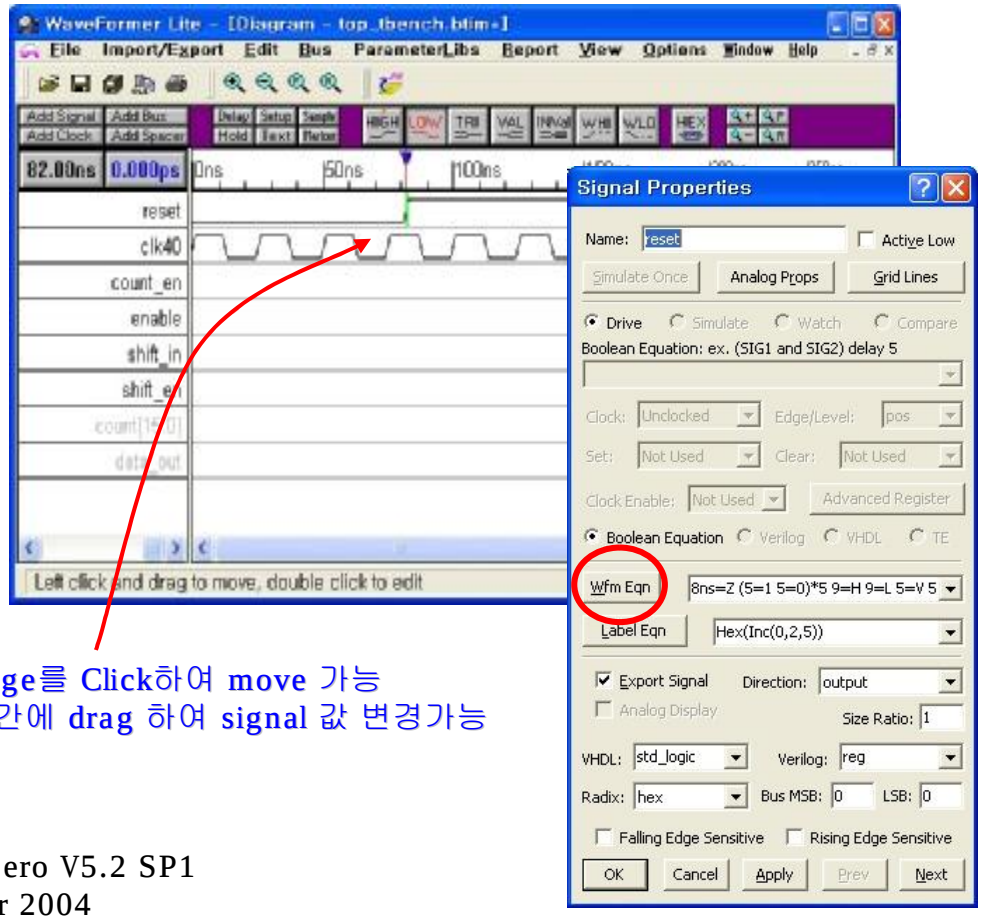
Waveformer Lite – Clock Properties

<Clock 속성 수정>
Clk40을 더블클릭 하면 Properties 창이 활성화 됨



Waveformer Lite – Edit Signal

1. 일반 1 bit signal 의 입력은 high/low icon을 이용하여 입력
   => 왼쪽그림에서 보면 현재 입력은 low로 선택되어 있으며
   한번 low 입력을 주면 자동으로 high로 전환됨



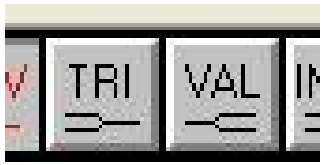
Edge를 Click하여 move 가능
중간에 drag 하여 signal 값 변경가능

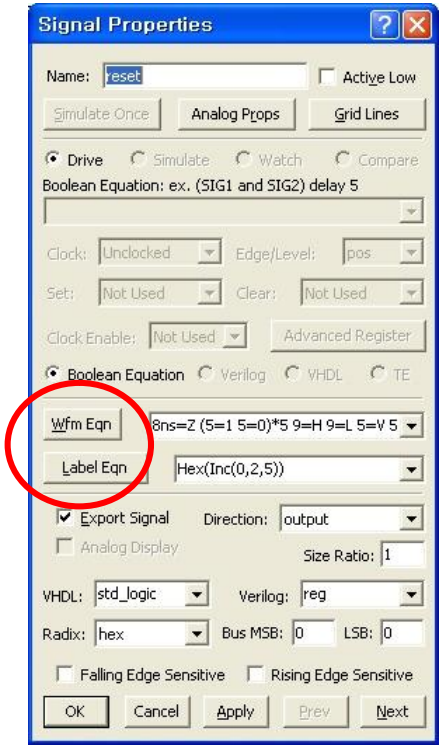
2.Signal 속성으로 주는방법 =>
signal을 더블클릭하여 속성창을
open Wfm Eqn 을 이용

예) 0ns=1 100ns=0 200ns=1
(50ns=1 50ns=0)*10

Libero V5.2 SP1
Apr 2004
Ver52

Waveformer Lite – Edit bus





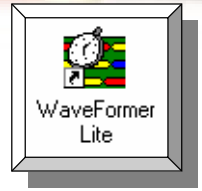
1. Bus signal에 입력을 주기 위한 icon 사용
   => 왼쪽그림에서
      TRI : Tri-state
      VAL: Valid Bus

2.Signal 속성으로 주는방법 => bus signal을 더블클릭하여 속성창을 open Wfm Eqn/Label Eqn 을 이용

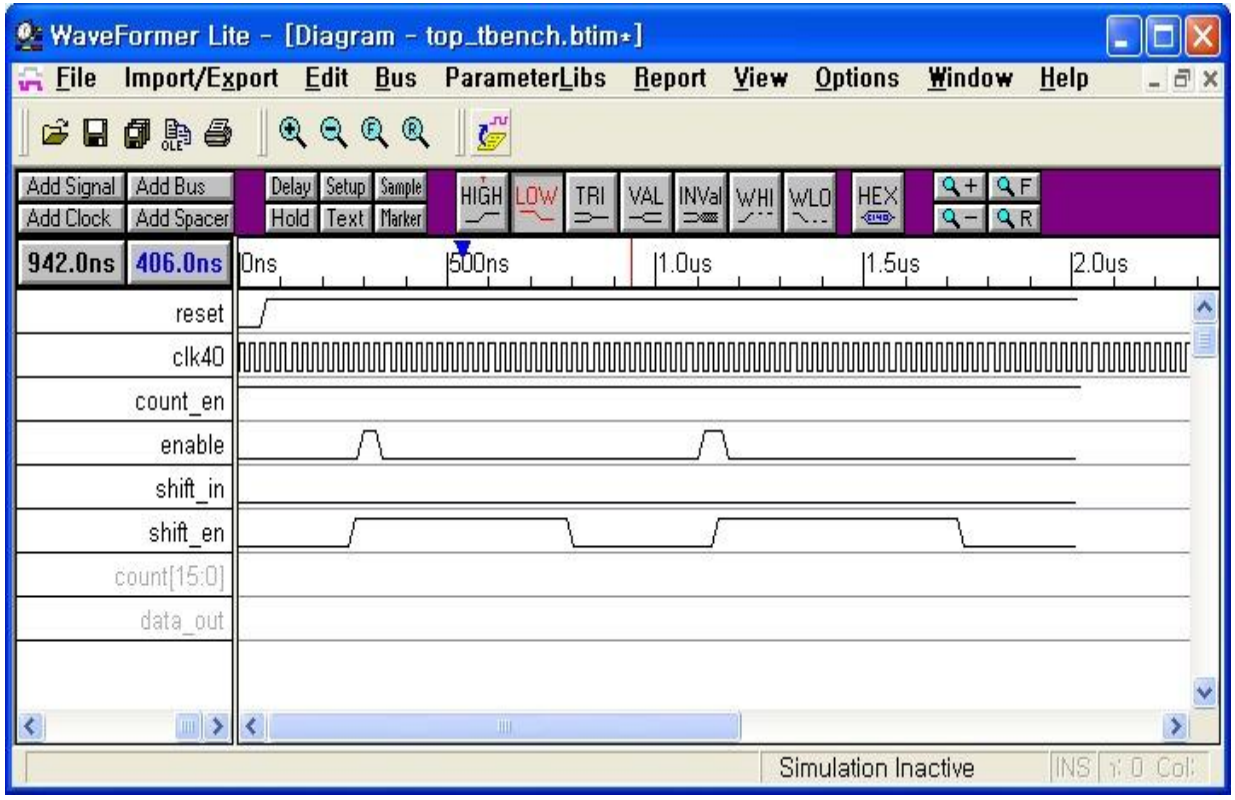
   예) wfm eqn => (100ns=V)*10
      : 100ns 간격으로 10개의 bus 입력
   Label eqn => Hex(Inc(0,2,10))
      : 초기값 0, 증가치 2, 데이터 개수 10

   File("vectors.txt") => 파일에서 입력
   Bin(RandInt(20,255)) => 랜덤값 입력

Waveformer Lite – Export files 1

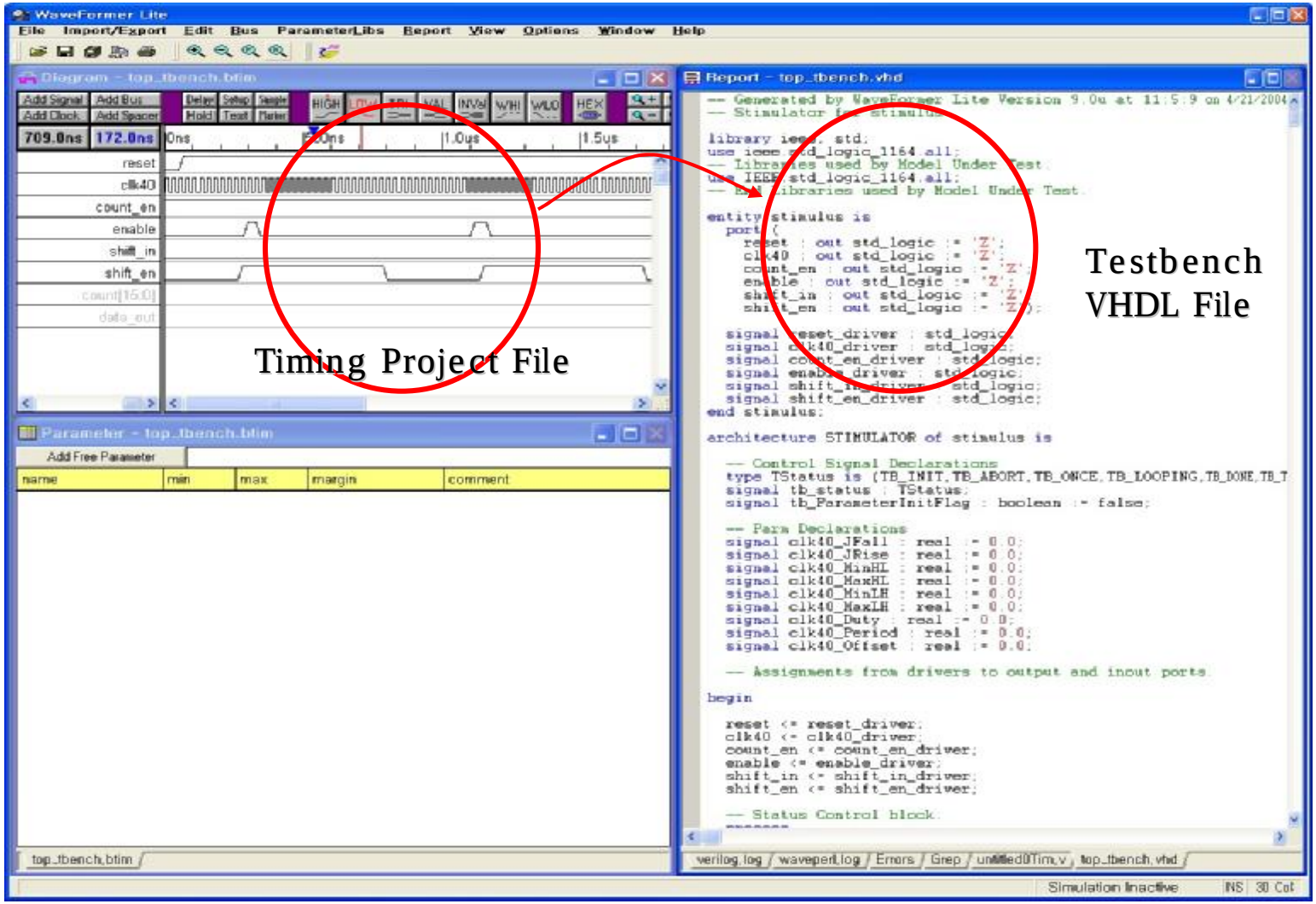


1. 파일 저장은 btim 형식과 VHDL 형식으로 두가지 모두 출력
   save as => Timing Project => top_tbench.btim
   save as => Testbench VHDL=> top_tbench.vhd
   => VHDL Wait with Top Level TestBench 을 선택

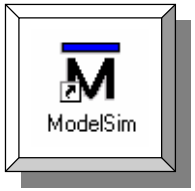


Libero V5.2 SP1
Apr 2004
Ver52

Waveformer Lite – Export files 2

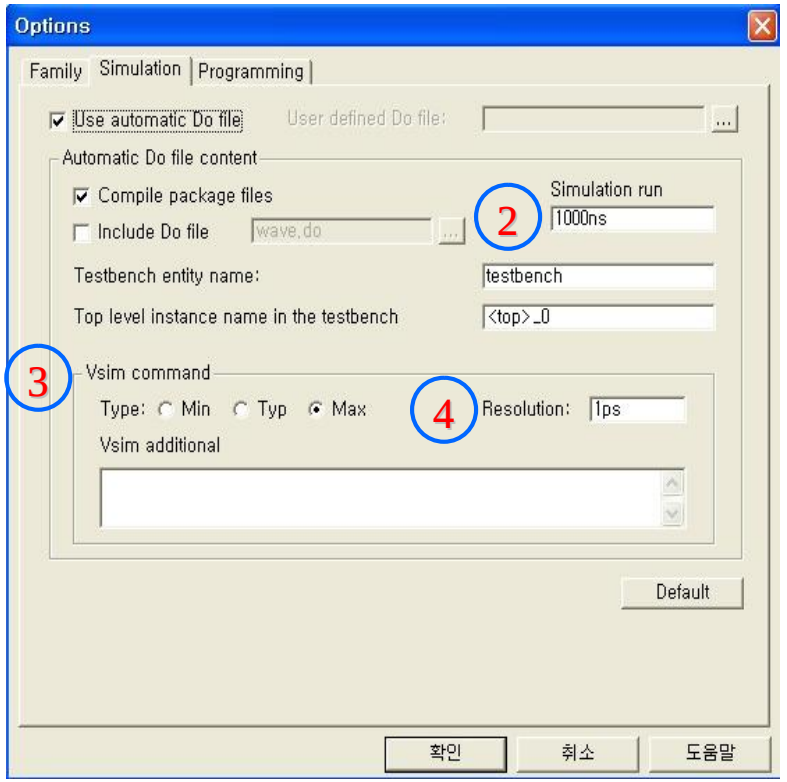


Simulation - Modelsim

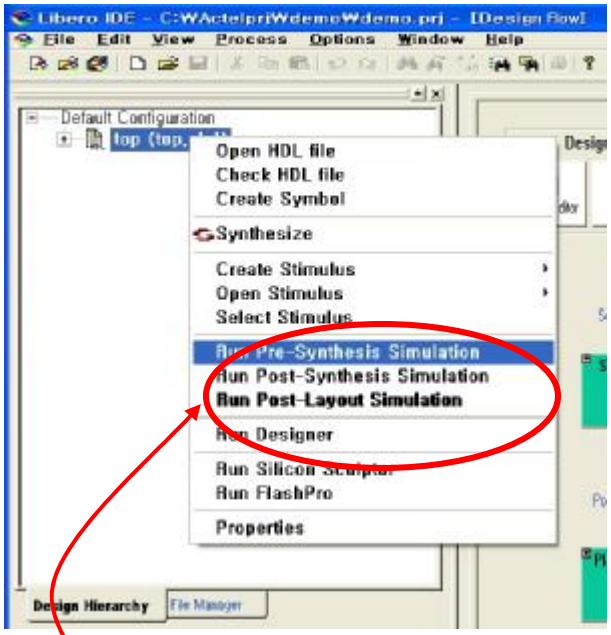


Libero Options – Simulation

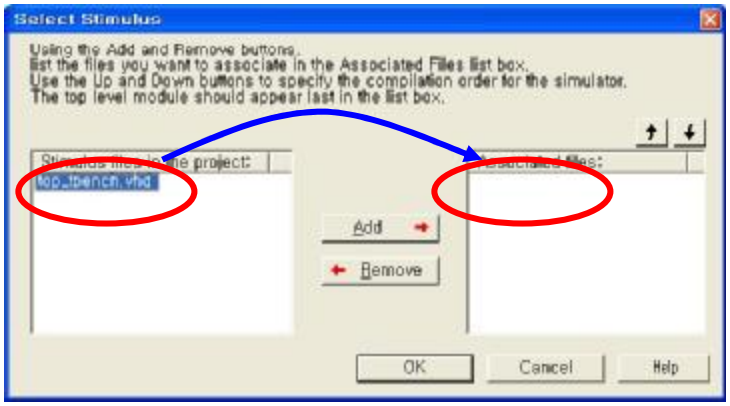
1. LiberoIDE 에서 options=>project settings 선택
   => Simulation 선택
2. Simulation end time 설정
3. Simulation type 설정
   MIN : best case sim
   TYP : typical case sim
   MAX : worst case sim
4. Resolution : simulation sampling time 설정
   정확한 simulation 결과를 얻기 위해서는 ps 단위로 설정



ModelSim – Select Stimulus



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

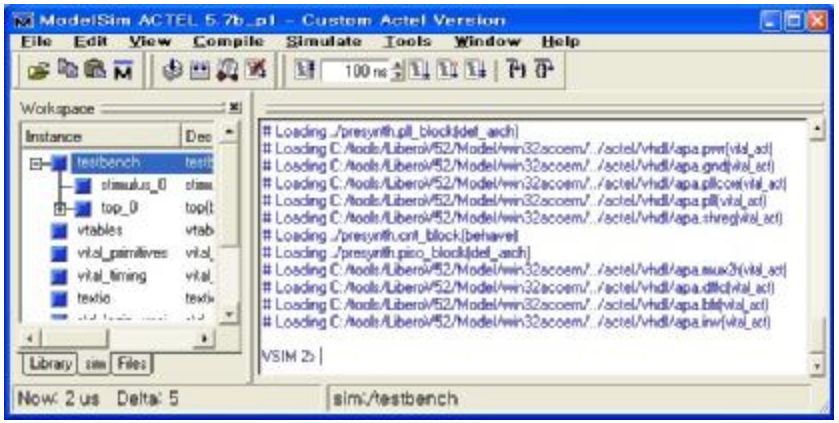


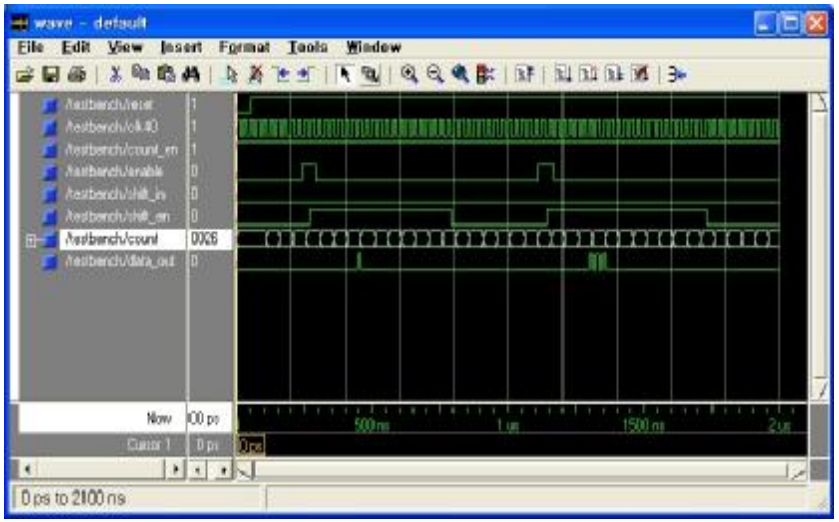
Run Pre_Synthesis Simulation : Function simulation
Run Post_Synthesis Simulation : Gate simulation
Run Post_Layout Simulation : Timing simulation

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

Libero V5.2 SP1
Apr 2004
Ver52

ModelSim – Select Stimulus





1. ModelSim 이 열리면서 자동적으로 compile, load design, simulation 실행

<Note> macro file 관리

modelsim에서는 세가지 simulation 모두같은 run.do 의 macro를 생성하기 때문에 이름을 따로 지정하여 관리하는 것이 편리함.

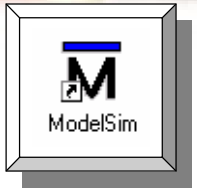
예) func.do, gate.do, time.do

<macro file>

menu 에서 tools -> macro -> file 선택하여 실행 가능

Libero V5.2 SP1
Apr 2004
Ver52

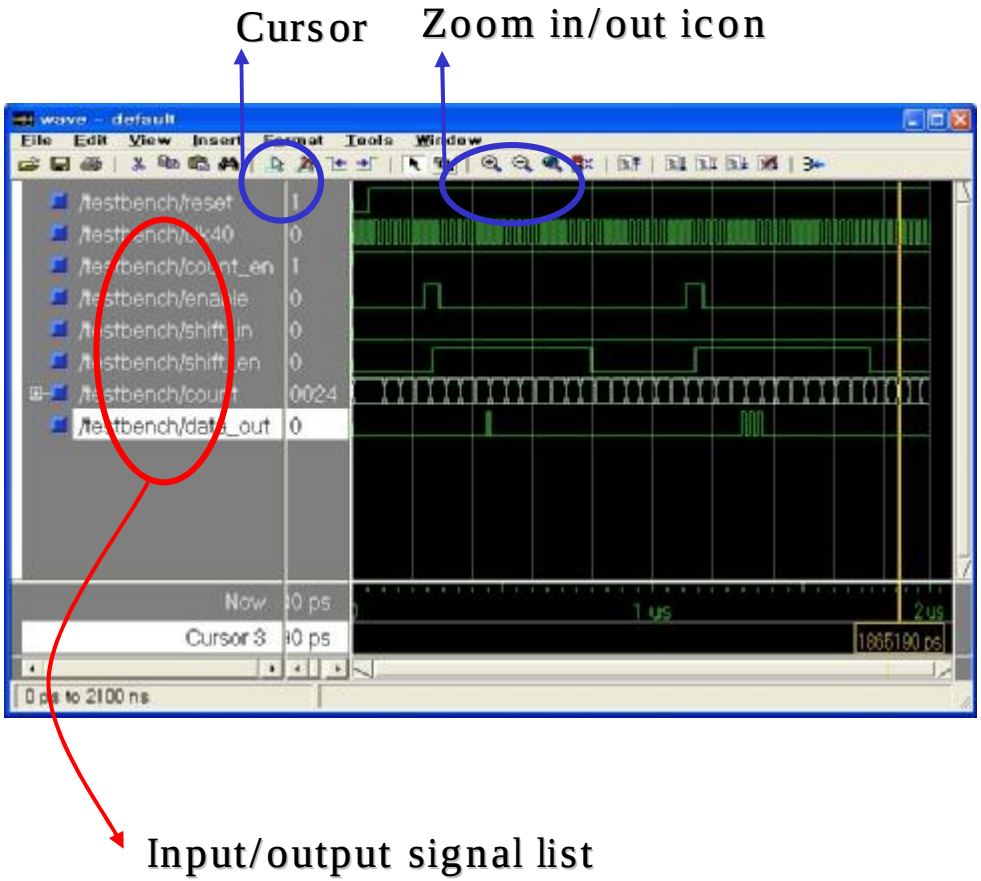
ModelSim – macro file




```
# function simulation
vlib presynth                                  ┐
vmap presynth ./presynth                       ┘   Simulation 의 종류선택
vcom -93 -work presynth ../hdl/pll_block.vhd   ┐
vcom -93 -work presynth ../hdl/cnt_block.vhd   │
vcom -93 -work presynth ../hdl/piso_block.vhd  │   Source VHDL Compile
vcom -93 -work presynth ../hdl/top.vhd         ┘
vcom -93 -work presynth ../stimulus/top_tbench.vhd ——— Testbench VHDL Compile
vsim -t 1ps presynth.testbench ——— Load Design
add wave /testbench/* ——— Testbench의 Signal을 wave 창에 add
run 2000ns ——— 2000ns까지 Simulation 진행

# Timing Simulation
vlib postlayout
vmap postlayout ./postlayout
vcom -93 -work postlayout ../designer/top_ba.vhd
vcom -93 -work postlayout ../stimulus/top_tbench.vhd
vsim -t 1ps -sdfmax /top_0=../designer/top_ba.sdf postlayout.testbench
add wave /testbench/*
run 2000ns
```

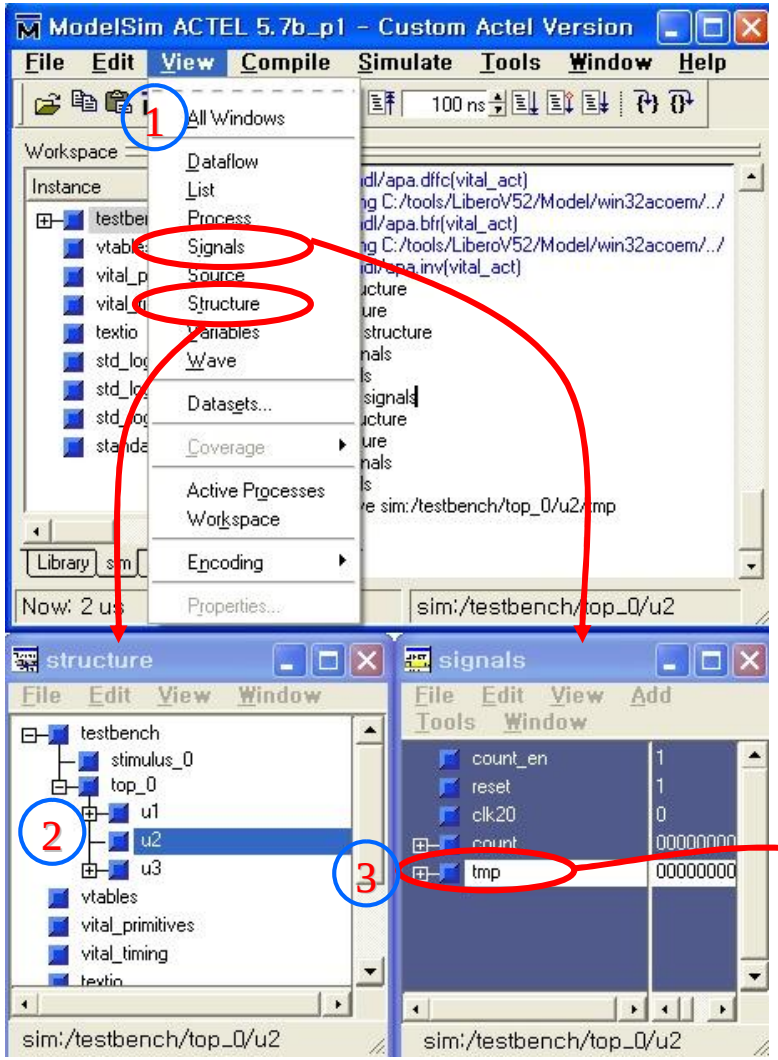
ModelSim – wave



<TIP>

1. 가운데 마우스 키를 이용하여 **zoom** (area, in, out, fix) 가능
2. **Cursor**를 이용한 **timing check**
3. **Data**를 읽기 쉬운 **type** 변형
   binary (default) -> hex or decimal
4. **Signal** 의 분리 및 병합 **(combine)**
5. **Signal** 의 색깔 지정

ModelSim – Internal Signal



<내부 signal을 보기위한 방법>

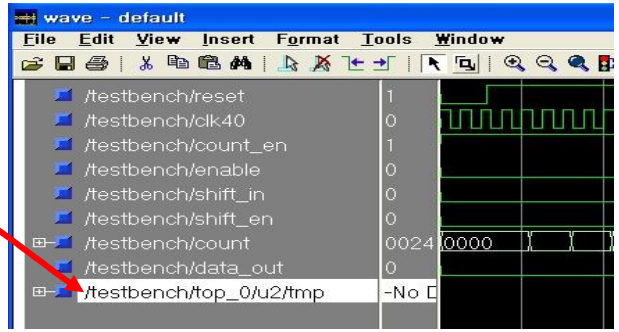
1. View -> Signals 과 Structure 창을open
2. Structure 창에서 sub-block으로 이동
3. Signals 창에서 원하는 signal 을 선택하여 add->wave 를 한다.

<add 방법>

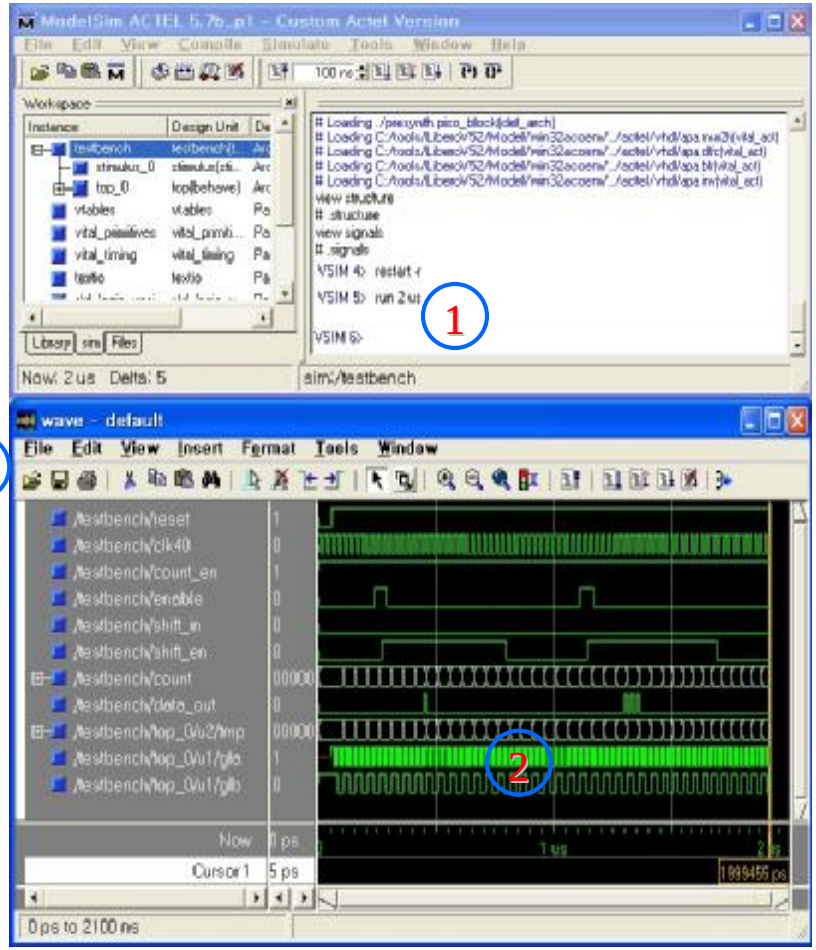
selected signal : 선택된 signal 만 add

selected in region : 선택된 block의 내부의 signal 만 add

selected in design : top block의 이하의 모든 signal add



ModelSim – Save wave.do



<Add 된 signal의 결과값은 나오지 않으므로
simulation을 다시 실행해야 됨>

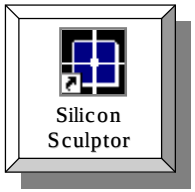
1 Modelsim main windows에서
vsim #> restart –f
vsim #> run 2 us
를 실행

2. Wave 창에서 결과 확인

3. Wave 창에서 wave.do로 저장하게
되면 다음에 이 추가된 signal이
자동으로 올라오게 됨

<NOTE> 저장된 wave.do를 자동 인식하기 위해서는
Libero의 options->project settings.. -> simulation 에서
Include DO file을 check 시 자동으로 불려오게 됨.

Programming – Flash Pro / Silicon Sculptor



Flash Pro – programming

Program 하기 위한 방법
⇒ Flash Pro를 이용한 방법

1. Design Flow에서 FlashPro 선택
2. File -> Connect..
   => Computer와 FlashPro와 연결상태 확인
   File -> Analyze chain
   => board 상의 actel device를 확인
   File -> Open Stapl File
   => Designer 에서 만든 .stp file load
3. Action 에서 PROGRAM 선택
4. Device -> 한 board 안에 2개이상의 device가 있을경우 program 할 device 선택
5. Execute icon 클릭

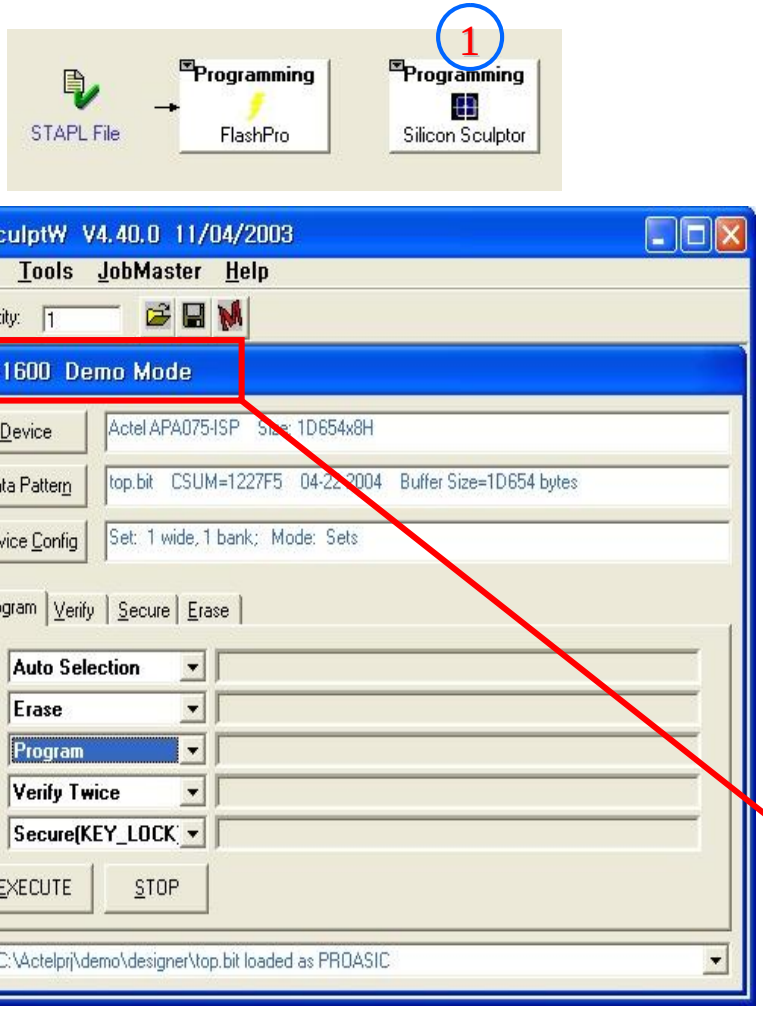
=> Program 완료 후
Pass, exit 0.

Flash Pro – programming

Flash Pro 를 이용한 다수의 device에 programming 순서

1. Flash Pro의 전원을 on
2. Target Board와 Flash Pro 의 케이블 연결
3. File-> connect를 선택, 해당 option check
4. File-> Analyze chain 선택
5. Device 와 Program file 을 선택
6. Program 실행
7. 완료 후, Flash Pro 케이블 분리
8. 다른 Target Board 와 연결
9. 위 4~8 항목을 반복

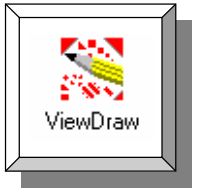
Silicon Sculptor – programming



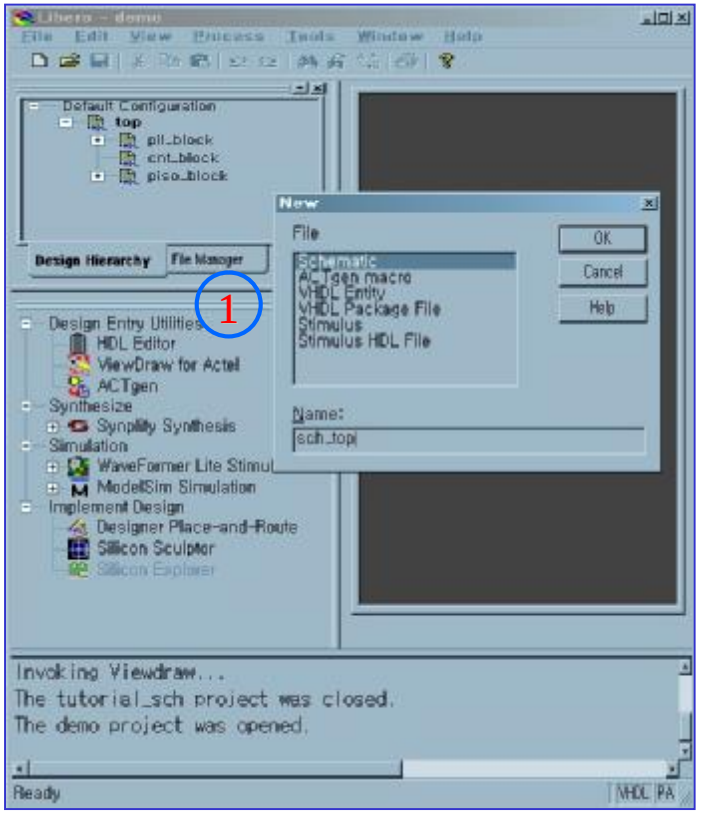
Program 하기 위한 방법
⇒ Silicon Sculptor를 이용한 방법

1. Design Flow에서 Silicon Sculptor 선택
2. Device -> program 할 device 선택
3. Data Pattern -> Designer 에서 만든 .bit 파일 open
4. Device config -> 한 board 안에 2개이상의 device가 있을경우 program 할 device 선택
5. Auto Selection/Erase/Program/Verify Twice 선택
5. Execute icon 클릭
   => Program 완료 후 Pass, exit 0.
   => 장비가 제대로 연결되어있지 않으면 Demo Mode 로 나옴. 정상동작시 program할 adaptor의 명칭이 올라옴.

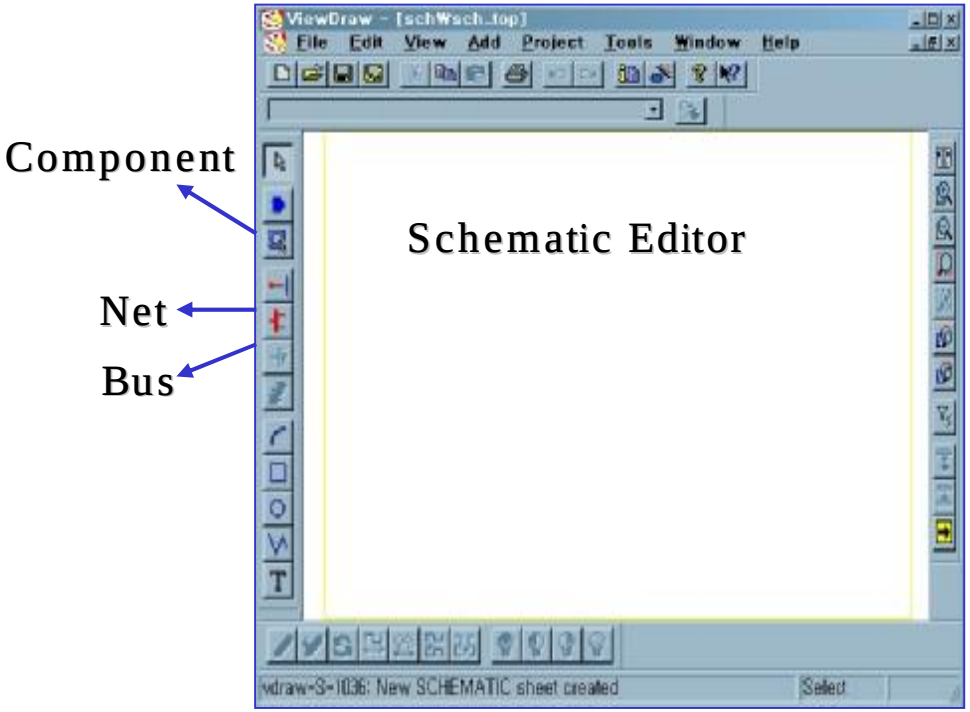
Schematic Design – View Draw



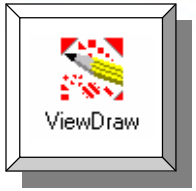
Creat Design – Schematic



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



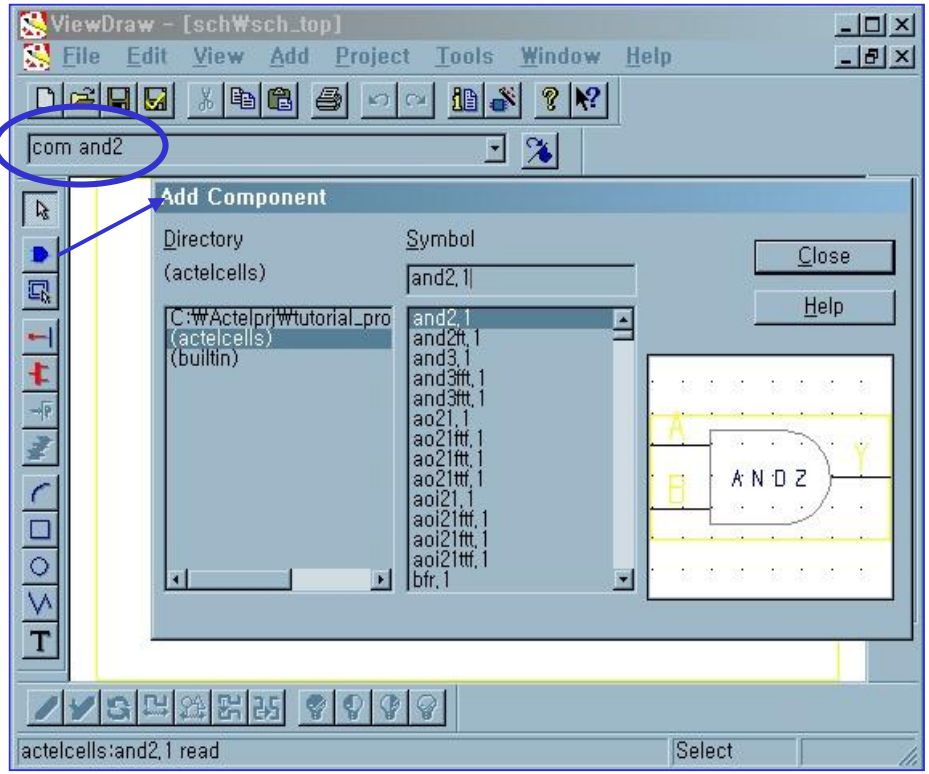
Creat Design – Schematic



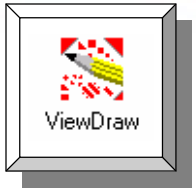
1. Component 선택시 두가지 방법

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

   ii. Add component 에서 원하는 component 를 선택



Creat Design – Schematic



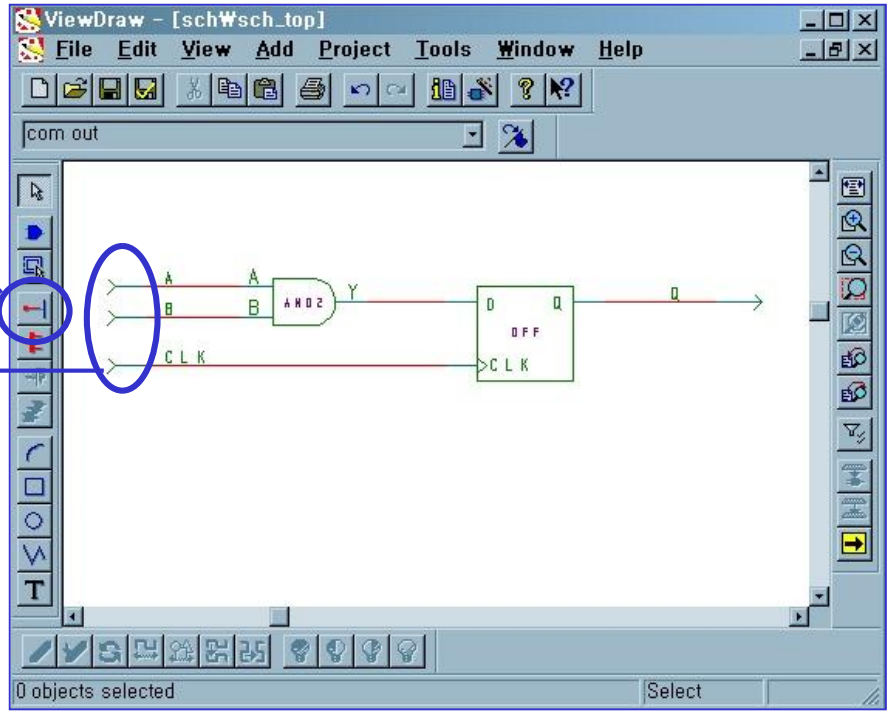
1. Net 연결은 net icon을 이용하여 component 간의 pin을 연결한다.

   i. 단축키 <n>를 이용할 수 있다.

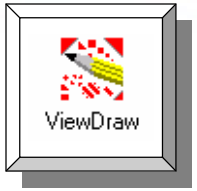
2. Block의 in/out은 component 에서 in 또는 out을 선택하여 net 끝에 연결한다.

3. Net name은 net properties에 label에 이름을 준다.

   i. In/out의 모든 net
   ii. 모든 bus 에는 이름을 주어야 한다.



Creat Design – Schematic

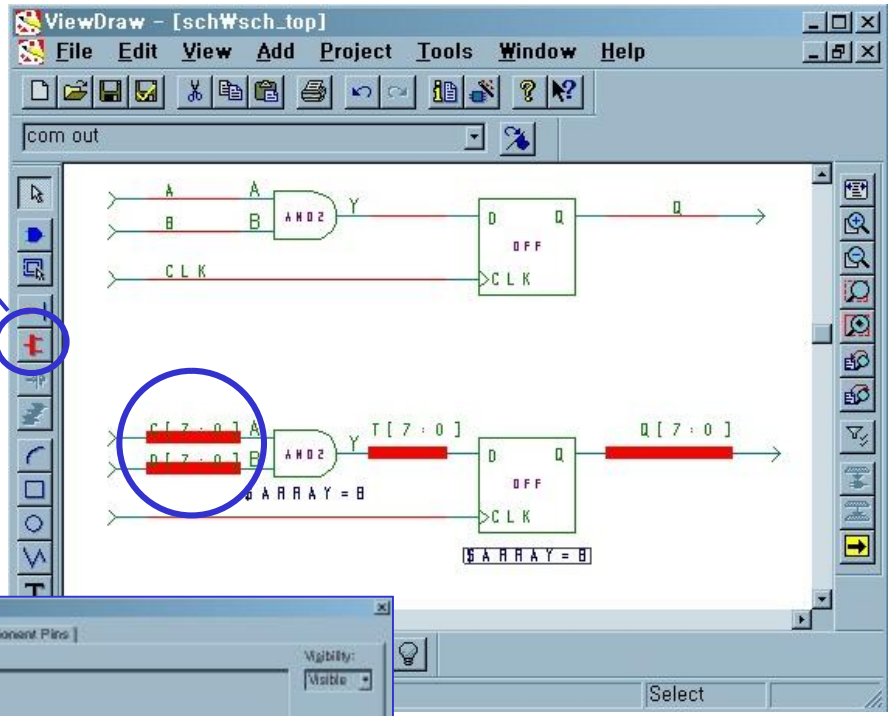


1. Bus 연결은 bus icon을 이용하여 component 간의 pin을 연결한다.
   i. 단축키 <b>를 이용할 수 있다.

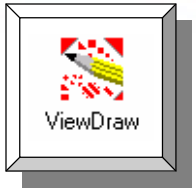
2. Bus name은 Bus properties에 label에 name을 준다.

3. Index를 위하여 [7:0] 의형태를 갖는다.

4. 배열을 주기위해서 component properties에서 array option을 이용한다.

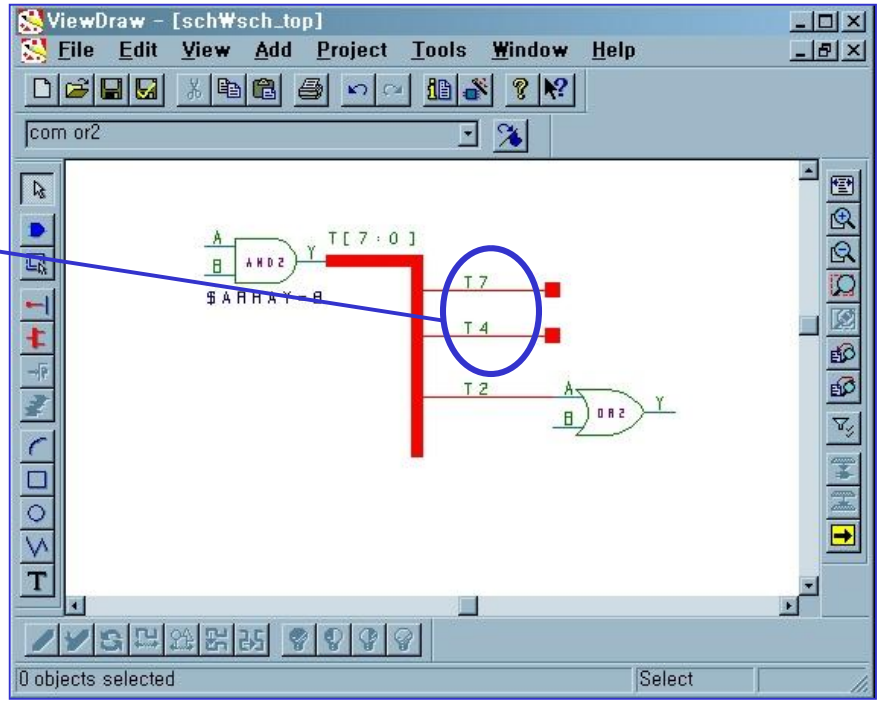


Creat Design – Schematic

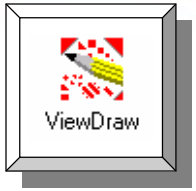


1. Bus 에서 bit 의 분리는 bus라인에서 net 바로 연결하여 사용한다.

2. Index를 bus name 에 바로 붙여사용

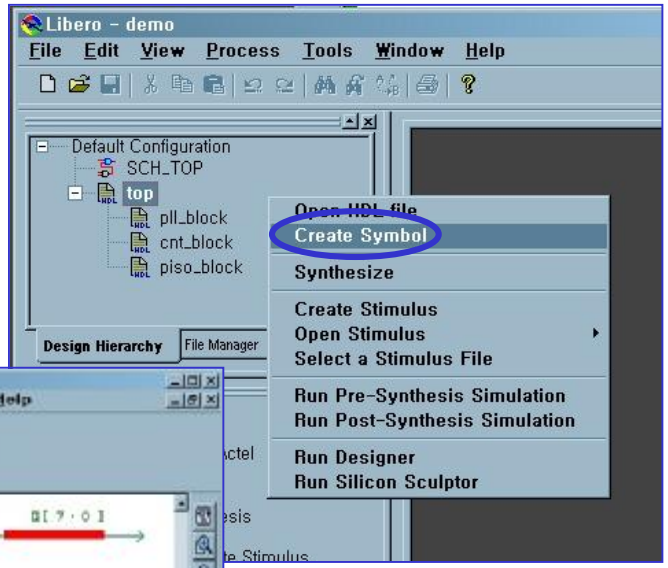


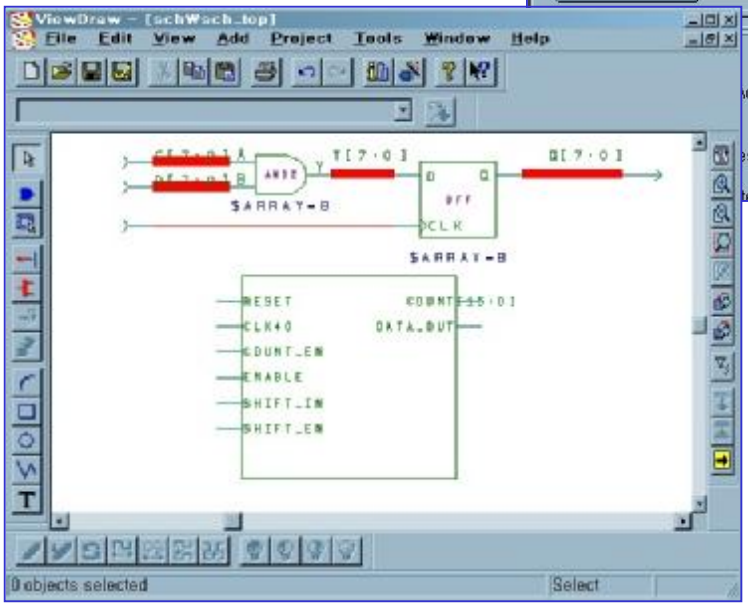
Creat Design – Schematic



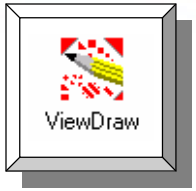
1. Vhdl 로 구성된 top design을 schematic의 sub block으로 만들기 위해서 create symbol

2. Add component에서 top design을 import 하여 net를 연결한다.





Creat Design – Schematic



1. Vhdl top design과 schematic 과의

2. Add component에서 top design을 import 하여 net를 연결한다.