

QuickWorks 사용 절차

1. 개요.

기본적인 QuickLogic Design 절차는 그림 1-1. 과 같이 Schematic Editor(회로도) 또는 Turbo writer(HDL) 를 이용한 회로 설계작업 후 Automatic Place and Route 를 통한 Design Implementation 작업 , SILOS simulator, Physical Viewer, Delay Modeler를 이용한 Design Verification 작업 그리고 마지막으로 Programmer, Tester 를 이용한 Device Programming 작업으로 완성된다.

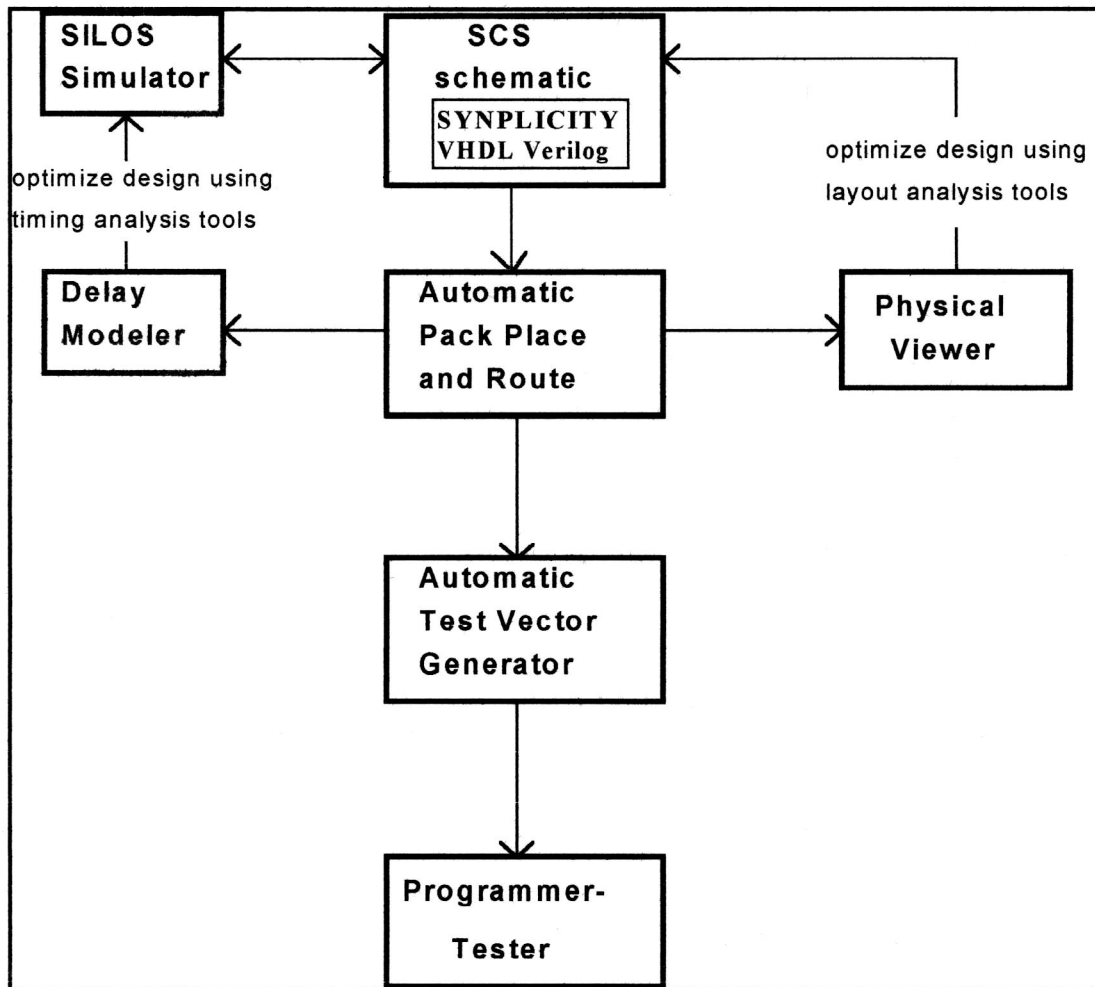


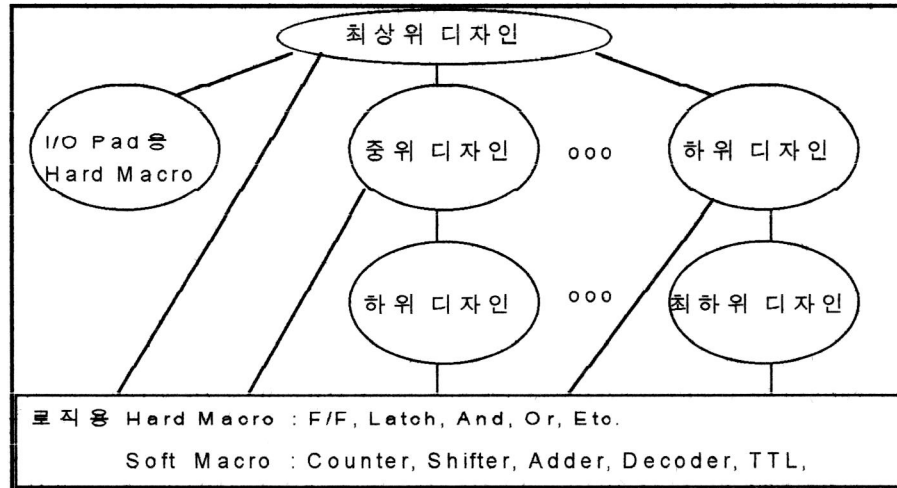
그림 1-1. Schematic Capture Design Flow

- 1). Windows 로 들어감 (DOS Prompt 상에서 "win" enter)
- 2). Windows 에서 SpDE icon 을 Double click.

2. Schematic Editor 를 이용한 Design Entry 과정.

QuickLogic 이 제안하는 기본적 Schematic Design 은 Bottom-Up 계층적 설계로 한다. 즉 Sub-Design 을 그리고 나서 해당 심볼을 생성시킨 다음, 상위 Design 에서는 하위 Design 에서 생성시킨 심볼들을 이용해 계층적 설계를 마무리 한다.

그림 2-1. Schematic Editor 를 이용한 Hierarchy Design 구조.



* 최상위 디자인상에 PAD종류를 사용해 실제 chip상의 pin 과 mapping 시켜줌으로써 Schematic을 완성한다.

다음 소개되는 디자인 절차는 4 비트 카운터의 Hierarchy Design 을 위한 것으로
서 ..\pasic\design\tutorial\schem 아래의 bbit0.*, bbit1.*,bbit3.* (Sub-Design)등의 화일들과 지금부터
만들 bbit2.sch, bbit2.sym을 이용해 Top Design인 bcount4.sch 화일을 만들어 보는것이다.

1). SpDE Main Menu 중 "Design" Field를 Pop-up, "Edit Schematic"을 Click. 또는 메뉴필드하단의
Schematic Editor Tool Bar를 Mouse 왼쪽버튼으로 Click.

2). "Launch Schematic Editor" Box 内 Filename 에다 수정하고자 하는 File 을 등록한 후 "Open"
을Click 하거나, 새로운 디자인을 하기위해서는 "New" Button 을 Click. 해 Schematic Editor를
Open 한다. 먼저 bbit2.sch그림을 완성하기위해 "New" Button 을 Click.

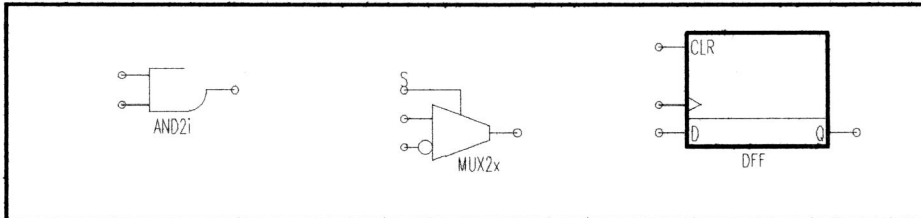
Editing Command Manu

ADD	SYMBOL	(F2)	
	WIRE	(F3)	
	NET NAME	(F4)	
EDIT	DELETE	(F5)	
	MOVE	(F7)	wire, symbol segment, region move
	DRAG	(F8)	rubber band move
	UNDO	(F9)	unlimeted undo
	COPY		wire, symbol, region
FILE	SAVE		
	Matching Symbol creating default symbol		
	Consistency Check		drawing fault check

3). "Schematic Editor" 화면의 "View" Field를Pop-up "Zoom-in","Zoom- out"을이용해 회로설계
에 적절한 화면을 만듬("Z" Curser 를 이용해서 일반적으로 가장자리의 위치 표시 번호가 보일
때까지 Zoom-in 함)

4). "Add" Field 를 Pop-up해서 "Symbol" Field를 Click, (또는 F2 키를 친 후) Mouse의 오른쪽 버튼을 Click하면 "Symbol Libraries" Box가 나오는데 Box 내 "Library" Field 의 여러가지 (and, nor, nand, etc) Catagory 에서 한가지를 선택하면 "Symbol" Field 에 해당 하는 심볼들이 나열된다. 이 중에서 원하는 것을 마우스 왼쪽 버튼으로 클릭, 그리고 작업화면의 적절한 위치에 선택된 심볼을 두고 마우스의 왼쪽 버튼을 클릭한다.

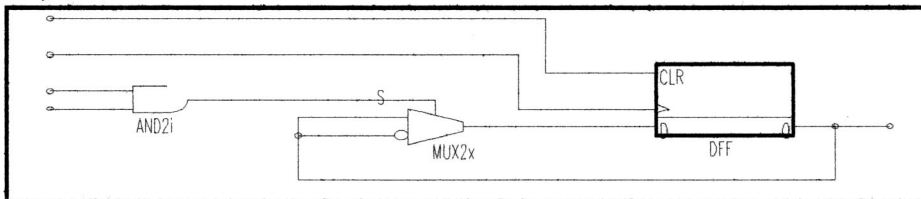
그림 2-2.



5). 6)번과 같은 방식으로 원하는 모든 라이브러리를 불러온 후 Net 연결 작업으로 들어감.

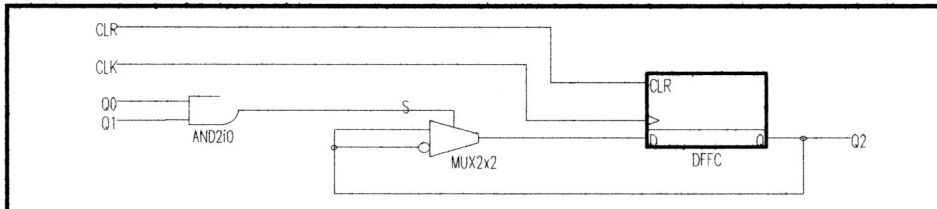
6). Net 연결은 "Add"Field상의 "Wire"Field를 Click 한 후(또는 F3 키를 친 후) 마우스의 왼쪽 버튼으로 연결 하고자 하는 심볼의 핀 끝을 click 하고서 다른쪽의 심볼 핀 끝으로 마우스를 옮겨 왼쪽 버튼을 클릭 한다.

그림 2-3.



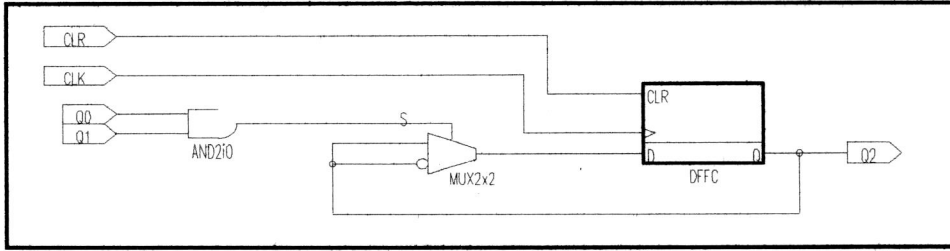
7). 각 Net들에 Net Name을 붙인다. ("Add" Feild 상의 "Net Name"Field를 Click 또는 F4 키를 이용 한다.) 먼저 붙이고자 하는 net 명을 기입한 후 enter 키를 치면 화면상에 그 이름이 올라오는데 마우스를 이용해 원하는 net의 끝의 작은 점까지 이름을 이동한 후 왼쪽 버튼을 클릭한다. 이때 십자형커서의 중심과 작은점의 중심을 일치시켜주는것이 중요하다. 또 여러 net 에 이름들을 붙이려면 net 명들을예와 같이 (clr,clk,q0,q1,q2) 차례로 기입한 후 enter 키를 치면 화면상에 그 이름들이 나타난다. 이때 마우스의 오른쪽 버튼을 클릭하면 첫번째 기입된 net 명만 남게 된다. 역시 이이름을 위와 같이 net 에 붙이면 다음 이름이 나타나고, 같은 방법으로 차례로 net 들에 붙여주면 된다.

그림 2-4.



net 명 모두를 붙인 후 "I/O Marker"를 net 명 위에 붙인다. 방법은 "Add" Field 내의 "I/O Marker"를 Click 하면 "I/O Marker" Box 가 나오는데 "Input, Output, Bidir" Field를 클릭해 해당하는 net 의 끝에 두어서 마우스 왼쪽 버튼을 클릭 한다.

그림 2-5.



8). Schematic Design 을 마친 후 이 디자인이 수정된 화일인 경우는 "File" Field상의 "Matching Symbol"을 Click해 해당 디자인의 심볼을 만든다. 이때 심볼은 자동적으로 생성되어 *.SYM 화일로 overwrie된다. 그리고 새로이 생성된 화일인 경우 화일을 Save 한 후 "Matching Symol"한다.

9). "FILE" Field 상의 "Consistency Check" Field를 Click해 회로상의 오류를 검사. 여기서 이야기하는 회로상의 오류는 Net연결이 불완전한 부분이 존재하는지를 의미하며Error 가있을때 Error Message Box에 Error가 나열된다. 마우스 왼쪽버튼으로 Error Message상을 Click 하면 일반적으로 Schematic상의 각 Net끝의 작은점이 존재하는곳을 지시하게된다. 만일 지시된부분의 Net가 사용자의 의도에의해 사용되지않는것일때 무시를해도된다.

10). "File"Field" 상의 "Save" Field를 Click해 작업된 화일을 저장.

11). Top Design 인 bcount4.sch 을 불러와서 위에서 만든 bbit2.sym 을 4). 과 같은 방식으로 Symbol ADD를 해서 그림 2-6 과 같이 bcount4.sch를 마무리한다.

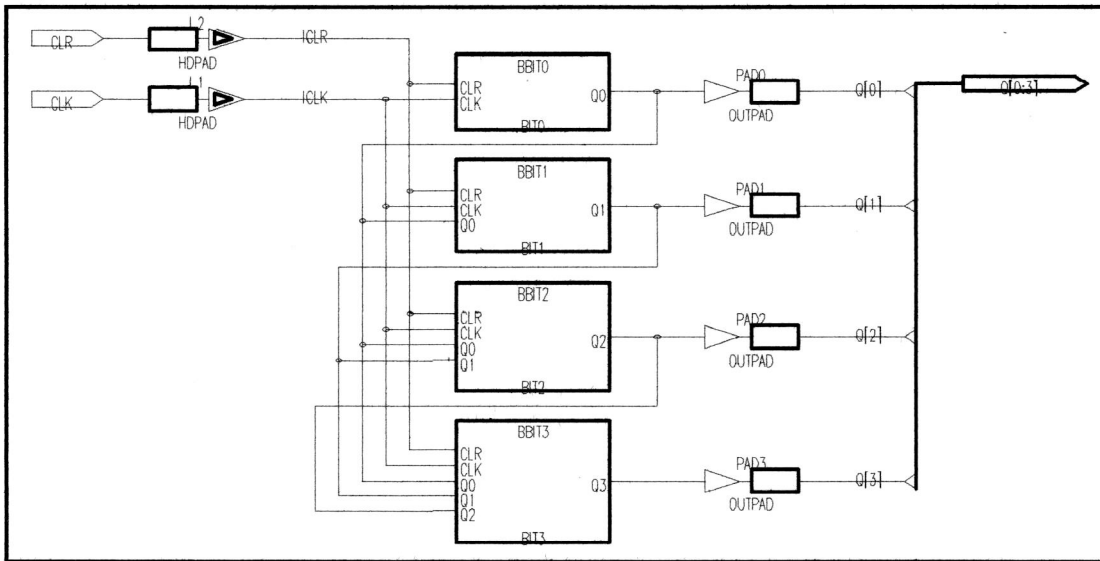
이때 앞에서 그려본 BBIT2.SCH 화일과 다른 것은이 같은 TOP DESIGN 에서는 HDPAD, OUTPAD와 같은 I/O PAD를 사용한다는 점과 "Q[0:3]"과 같은 BUS 를 만드는 것이다. I/O PAD에는 다음과같이 용도에 따라 구분해서 사용한다.

- * INPAD: 실제 디바이스 상의 I/O PIN 에INPUT으로 ASSIGN된다. 드라이브 가능 FANOUT 은 13 개.
 - * HDPAD: 실제 디바이스 상의 INPUT PIN에 ASSIGN된다. 드라이브 가능 FANOUT은 24개. 단 HDPAD는 Quick Logic 모든 Part에있어 공히 6개있으며 최대 4개까지 PCB상에서 묶어줌으로써 96개의 FANOUT을 얻을수 있다.
 - * CKPAD: 실제 디바이스 상의 CLOCK PIN에 ASSIGN 된다. 드라이브 가능 FANOUT은 해당 디바이스의 MACRO CELL 수 와 동일함. 단 CKPAD는 Quick Logic 모든 Part에있어 공히 2개 있으며 F/F의Clock, Reset, Preset으로만 연결 가능함. 만일 CKPAD를 HDPAD용도로 사용하려면 CKtPAD를 이용해 3Port중 맨 위 , 맨 아래의것을 사용하면된다. 그리고 가운데 것은 CKPAD의 OUTPUT 와 같다.
 - * OUTPAD: 실제 디바이스 상에 I/O PIN에 OUTPUT 으로 ASSIGN 된다.
- 상기의 각종 PAD는 DATABOOK의 pASIC MACRO LIBRARY부분을 참조.

BUS를 만드는 방법은 일단 WIRE(NET)를 그린 후 "ADD" FIELD 내의 "BUS TAB" FIELD를 CLICK 하고 그려 는 WIRE에서 부터 원하는 심볼의 PIN STUB(PORT) 까지 마우스 왼쪽 버턴 을 이용해서 그린다. 단 BUS TAB은 수평 또는 수직 으로만 그릴 수 있다. BUS TAB이 붙게 되면 WIRE의 색이 빨간색으로 변하게 되는데 이 WIRE 상에 BUS NAME을 예와 같이 (예: DATA[0:7]) ASSIGN 한다. 그리고 BUS TAB 상의 NET NAME은 BUS 를 마우스 왼쪽 버턴으로 선택 하면 BUS NAME이 자동적으로 나타나고 다시 마우스 오른쪽 버턴 을 한번 CLICK하면 BUS 각각의 이름이 예와 같이 (예: DATA[0], DATA[1],DATA[7]) 순차적으로 나타나게 되는데 각 이름을 적절한 BUS TAB상에다 ASSIGN한다.

* 본 Schematic Editor에서는 VCC, GND가 Symbol이 아닌 Net Name으로 Assign되므로 VCC, GND가 필요할때 ADD NET NAME을통해 사용한다.

그림 2-6.



12). "Schematic Editor"Shell 을 Minimize.

3. Hierarchy Navigator를 이용한 QDF (Quick Logic Netlist Format) Extraction.

1). "SpDE" Shell 에서 "Navigate Hierarchy"Field 를 Click, "Launch Hierarchy Navigator" Box를 띄워 "New"Field를 Click 해 Save된 *.SCH file (예를들어 만들어는 BCOUNT4.SCH)을 Double Click 한다.

2). 1)번의 결과로, Building Hierarchy Block이 작동되고 작업된 파일의 "Hierarchy Navigator" 화면이 나오는데 "F2=Push/Pop" Field 를 Click해 Design Hierarchy 구조를 살펴볼 수 있다. "Process" Field상의 "Export QuickLogic"를Click해 "Netlist Information" Box가 올라오면 "Netlist Format" field 에 생성하고자 하는netlist format (예를 들어 pre layout simulation을 하고자할때 "Verilog"를 선택하고 post layout simulation 또는 chip 으로 구현하고자 할때 "QDIF"를 선택) "Current Part", "Current Package" Field에 원하는 사항을 등록한후 "OK"를 Click.QDF File 을 생성하고, "File"Field상의 "Save"Field를 Click해 *.TRE File을 저장.

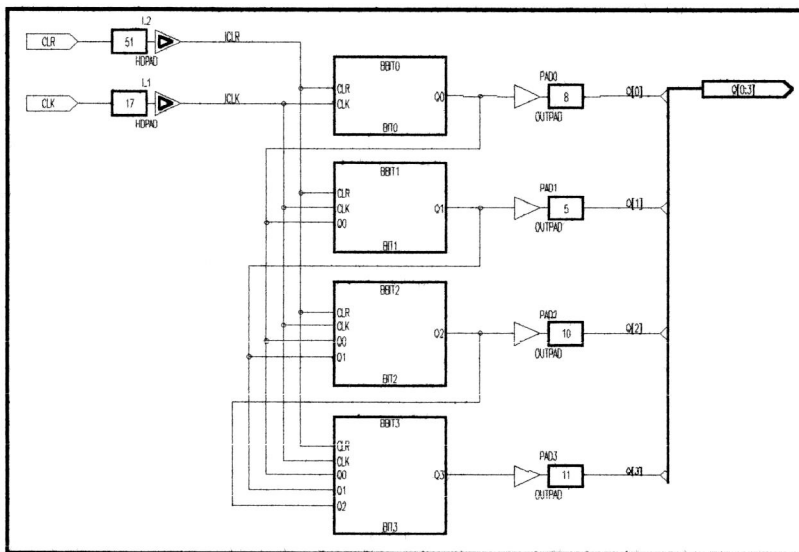
3). "Hierarchy Navigator" Shell을 Minimize.

4. Design Implementation (Place & Route).

- 1). "SpDE" Shell상에서 "File"field 상의 "Import"를 Click "QDIF"를 Click해 "Select diectory and existing file:"Box를올린다. 이 Box상에서 "File Name"Field 에 원하는 화일을 등록 후 "OK"를 Click.
- 2). 1)번의 결과로 Pin number만 Display 된 Physical Viewer가 "SpDE" Shell상에 올라온다. 이 상태에서 "Tools"field상의 "Run All Tools"를 Click해 실행시키면 "Level 2 Logic Optimizer", "Placer", "Router", "Delay Modeler", "Back-Annotation", "Sequencer", "ATVG" tool들이 자동으로 실행된다. 실행 완료후 "OK"를 Click하면 화면상에 Place & Route 결과가 Display 된다. "Zoom-in", "Zoom-out"를 이용해 화면을 검색해 보고 "File" Field상의 "Save"Field를 Click해 *.ATR, *.CHP file을 저장한다.
- 3). "SpDE"화면을 minimize.
- 4). "Hierarchy Navigator" Shell 을 maximize 시켜 "File"field상의 "Back Annotate"를 click한후 "View"field상의 "Redraw"를 click 하면 I/O pad상에 pin number가 Display된다.

지금까지의 작업이 회로 Design 및 Place & Route 과정이고 다음으로 Simulation 작업에 들어 간다.

그림 4-1.



5. SILOS III 을 이용한 Design Simulation 및 Verification 과정.

- 1). "Hierarchy Navigator" Icon을 Maximize.
- 2). "SILOS III"Field상의 "Waveform Edit"Field를 Click해서 "Waveform Editing Tool" Shell 과 "Noting Selected" Box를 띄운다.
- 3). "Waveform Editing Tool" Shell에서 "Option"Field상의 "Setup Options..."Field를 Click해 "Setup Option" Box를 띄운다.
- 4). "Time units" Block내에서 적절한 (예를들어 "1.0" 과 "ns"를 Click해서 1 ns 의) Time Unit를 정한다. "Simulation Time:"Field에 적절한 시간을 적고 (예를들어 10,000 ns), "Export On Save" Option을 Click해서 Enable시킨다. "Setup Option" Box를 닫는다.

5). "**Waveform Editing Tool**" Shell로 가서 Display되어진 Input node name(CLK, CLR)을 확인한 후 각 Input node들에 대한 Input Stimulus 생성 작업을 한다.

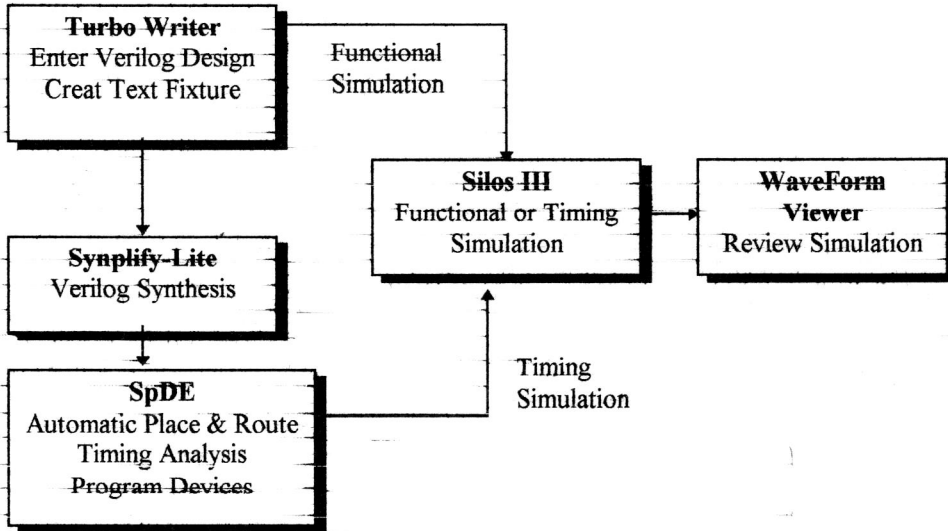
작업방법은 다음과 같다. 우선 작업할 공간에 적절한 한 Input node 를 Click해 선택한 다음 , Mouse를 원하는 시점에서 Click하면 **Nothing Selected** block상단에 현재 선택된 Node 명이 나타나고 state field 에는 high, duration field에는 editing된 시간을 보여주며 waveform editing tool의 작업공간에 자동적으로 High 로 wave가 그려진다. 이때 그려진 wave상하단에 bold line이 현재 선택된 wave를 지시한다. 그리고 다음 원하는 시점에서 Click 하면 Low로 Toggle되는 방식으로 계속 반복 작업을 수행하면서 Wave를 그려 나간다. 또 클릭과 같은 주기적으로 반복되는 Node는 한주기만을 edit하고 마우스 왼쪽버튼으로 한주기의 waveform을 선택하면 **selected...** block내의 repeat field상에 반복할 횟수 또는 forever field를 enable시켜 repeat시킨다. 모든 Input node에 대해 원하는 Stimulus 생성 작업을 한다. "**File**" Field상의 "**Save**" Field를 Click해 작업한 ***.WET** 파일을 저장한다. "**Waveform Editing Tool**" Shell을 닫는다.

6). 다시 "**Hierarchy Navigator**" Shell로 가서 "**SILOS III**" Field상의 "**setup simulation options..**" Field를 Click해 그 block을 띄운다. 먼저 simulation type을 Post Layout로 setting하고 corner는 Typ로 한다. 그리고 text fixture, top level design, sdf file field상 에 적절한 file명을 기입한다. (여기서는 bcount4.tf, bcount4.vq, bcount4.sdf) 그리고 **OK** click. " Hierarchy Navigator" menu로 가서 "SILOS III" field상 run simulation 을 click "SILOS III" shell을 띄운다. logic sim menu 에서 run logic simulator를 click "simulate" block을 띄운다. simulation하고자 하는 time을 T1(start), T2(end)field에 기입한다.(예를들어 T2는 10US) **simulate** button을 click. 하면 simulation process를 100% 완성한다. 다시 navigator의 SILOS III menu 에서 review silos result field를 click하면 simulation 결과를 보여주는 **Waveform Viewer**" Shell이 올라온다. 이때 input output bidir node 의 simulation 결과 wave들이 자동display 되는데 추가로 internal node 의 simulation 결과를 보고자 할 땐 navigator shell의 Tools menu 에서 probe item field를 click해서 현재 navigator 상의 node들을 마우스 왼쪽 버튼을 click하면 그 node의 simulation 결과가 waveform viewer상에 display된다. 한화면상에 "**Hierarchy Navigator**", "**SILOS III**", "**Waveform Viewer**"를 항상 볼 수 있도록 각 Shell 들을 적절한 크기로 배치한다. zoom in, zoom out을 이용해 timing 분석을 한다.

Simulation 작업이 끝난 후 "**File**" Field상의 "**Save**"를 Click해 Simulation결과를 저장한다. 지금까지 기술한 simulation방법은 실제 device 의 delay가 고려된 Timing Simulation이며 function simulation 을 하려면 다시 "**Hierarchy Navigator**" Shell로 가서 "**SILOS III**" Field상의 "**setup simulation options..**" Field를 Click해 그 block을 띄운다. 먼저 simulation type을 Pre Layout로 setting하고 text fixture, top level design field상 에 적절한 file명을 기입한다. (여기서는 bcount4.tf, bcount4.v) 그리고 **OK** click. ******.v** file은 export quicklogic 을 실행할때 Netlist format을 Verilog 로 하면 자동 생성된다.)

6. Verilog HDL 을 이용한 설계절차

그림 6-1. Verilog Tutorial Design Flow



여기서는 Turbo Writer 를 이용해 간단한 Verilog HDL 화일 latch4.v 와 Text Fixture화일 latch4.tf 을 만들어 Function Simulation , Synthesis , 및 Timing Simulation 을 하는 연습을 해본다.

- 1) Spde main memu상의 “Design”아래 “Text Editor”를 click 해서 “Turbo Writer”를 띄운다.
상단의 menu에서 “File” 아래 “Change Directory”를 click해 작업하고자 하는 directory를 지정하고 “New” field를 click 작업할 화일명을 기입(여기서는 vcount4.v)해 화일 을 open한다.
작업 sheet 에 다음과 같은 Verilog 화일을 write한다.

그림 6-2 latch4.v

```

module count16(TRIOUT,ten,load,clk,laten);
output [15:0] TRIOUT;
input ten, load, clk, laten;
reg [15:0] data;
wire [15:0] q;
always @(posedge clk)
begin
    if (!load)
        data = 16'b1011000010000000;
    else
        data = data + 1;
end

assign {q[15:0]} = laten ? {data[15:0]} : {q[15:0]};
assign {TRIOUT[15:0]} = !ten ? {q[15:0]} : 16'bz;

endmodule
  
```

- 2) 위와 같이 write한후 menu “HDL”아래 “Synplicity Verilog Syntax Checker”를 click해서 화일의 syntax 오류를 검사한다. 검사 결과는 Turbo Writer window 하단에 display된다.

3) 오류 검사에서 “no more error” 메세지 가 display 되면 menu “HDL”아래 “Generate Verilog Test Bench” 를 click 하면 다음과 같은 default file(latch4.tf)가 display 된다.

그림 6-3. 자동 생성된 latch4.tf

```

/*****
**
** Automatic Verilog Test Fixture Generation
** =====
**
** Generated by HDL Turbo Writer
** Version 2.0b (c) SAROS Technology Ltd 1994
**
*****/

`timescale 1ns/1ns

module t;

wire [15:0] TRIOUT;
reg load, clk, laten, ten;

    count16 m (. TRIOUT(TRIOUT),.ten(ten),.load(load),.clk(clk),.laten(laten));

    // Enter fixture code here

endmodule // t
```

위와 같이 자동 생성된 latch4.tf화일에 사용자의 Stimulus Fixture Code 를 “// Enter fixture code here” 라인 다음에 다음과 같이 edit 한다.

그림 6-4. 완성된 latch4.tf

```

/*****
**
** Automatic Verilog Test Fixture Generation
** =====
**
** Generated by HDL Turbo Writer
** Version 2.0b (c) SAROS Technology Ltd 1994
**
*****/

`timescale 1ns/1ns

module t;

wire [15:0] TRIOUT;
reg load, clk, laten, ten;

    count16 m (. TRIOUT(TRIOUT),.ten(ten),.load(load),.clk(clk),.laten(laten));

    // Enter fixture code here
initial begin // 'load'
    load = 0; #60;
    forever
    begin // Patt_2
        load = 0; #18;
        load = 1; #2302;
    end
end
```

```

    load = 0; #160;
end
end // load

initial begin // 'clk'
    forever
    begin // Patt_4
        clk = 1; #20;
        clk = 0; #20;
    end
end // clk

initial begin // 'laten'
    forever
    begin // Patt_6
        laten = 0; #270;
        laten = 1; #70;
    end
end // laten

initial begin // 'ten'
    forever
    begin // Patt_8
        ten = 1; #2450;
        ten = 0; #2650;
    end
end // ten
endmodule // t

```

4) 앞에서 준비된 latch4.v, latch4.tf를 가지고 Function simulation을 해본다.

Function Simulation 을 하려면 “SpDE” main menu에서 “Design”아래 “Silos III Simulator”를 click 해 “Silos III Simulator Options” window 를 띄운다. 그리고 다음과 같이 option setting을 한 후 OK를 click.

```

“Test Fixture” ..\latch4.tf
“Top Level Module” ..\latch4.v
“Simulation Type” Pre-Layout

```

“QuickWorks SILOS III” window가 뜨고 준비 완료되면 menu “Logic Sim”아래 “Run Logic Simulation...”을 click 해서 “Simulate” window 를 띄운다. 그리고 5장에서 설명된 Silos III simulator 이용법과 동일한 절차를 수행한다.

5) Simulation 결과는 main menu “Design”아래 “Waveform Viewer”를 click해 보고자하는 simulation 파일 (여기서는 latch4.sim)을 click 해서 보면 “Waveform Viewer - Silos -latch4” window가 뜬다. 여기 menu “Edit”아래 “Show”를 click “Hierarchal Name List” window가 뜨고 “t”로 display된 부분을 double click 또는 click 한 후 “Push”를 click 해 보면 다음과같은 signal name 이 나온다.

TRIOUT[0] .. TRIOUT[15], \clk<c>, \laten<c>, \load<c>, \ten<c>, clk, laten, load, ten 등
verilog source file에서 설정된 모든 signal 이 나오면 TRIOUT[0] 부터 TRIOUT[15]까지 mouse drag해 “Add to Bus”를 click bus로 묶어 “Show”click 나머지 signal은 각각 선택 한 후 “Add Wave” 를 click “Waveform Viewer - Silos -latch4” window 에 display 시켜 logic fuction 을 검정 한다.

6) logic 의 function 을 검정 한 후 Quick Logic device로 구현 작업 및 Timing Simulation 을 한다.
먼저 Quick Logic device로 구현 작업은 다음과 같다.

SpDE main menu “File” 아래 “Import” 아래 “Verilog”를 click synthesis 할 화일 을 지정 하면 “Synplify_Lite” window가 뜬다.

“Verilog Source File” field에 ..\latch4.v 를 “QDIF Result File” field 에 ..\latch4.tf 를 “part” 및 “Package” 에 적절한 device , package를 지정 한 후 “Run”버튼을 click 하면 “compile” , “.mapper”가 동작하고 작업 완료 후 생성 된 latch4.qdf 화 일 을 자동 “SpDE”로 load 한다. 4장 에서 설명된 design implematation 절차를 진행한다.

7) 이렇게 구현된 latch4.chp 파일에 대한 Timing Simulation을 해본다.

Timing Simulation 을 하려면 “SpDE” main menu에서 “Design”아래 “Silos III Simulator”를 click 해 “Silos III Simulator Options” window 를 띄운다. 그리고 다음과 같이 option setting을 한 후 OK를 click.

“Test Fixture” ..\latch4.tf
“Top Level Module” ..\latch4.vq
“SDF File” ..\latch4.sdf
“Simulation Type” Post-Layout
“Corner” Typ

“QuickWorks SILOS III” window가 뜨고 준비 완료되면 menu “Logic Sim”아래 “Run Logic Simulation...” 을 click 해서 “Simulate” window 를 띄운다. 그리고 5장 에서 설명된 Silos III simulator 이용법과 동일한 절차를 수행한다.

Timing Simulation 검증은 Function Simulation 의 결과를 검증하는 방법과 거의 동일하나 “t” 또는 “m” 를 push 해 interner signal 도 “Waveform Viewer”에 display할 수 있다.