

# Verilog HDL icarus Simulator 설치

# Verilog HDL의 역사

- HDL(Hardware Description Language) 등장 배경
  - Gate Count 및 디자인의 복잡성 증가
    - 대형회로를 설계시 기존의 설계 방법 비효율성(시간/비용 증가)
- 복잡한 하드웨어를 쉽게 기술하기 위하여 언어를 사용하는 방법 연구
  - Programming 언어 : 회로의 지연처리, 합성 등의 문제점
  - HDL : VHDL, Verilog HDL, AHPL, IDL, ISPS, UDL/I 등이 연구, 사용
- HDL의 종류
  - VHDL(Very High Speed IC Hardware Description Language)
    - 미국 국방성 중심으로 1987년 표준화
    - 1993년 IEEE 1076-1993으로 불리는 IEEE 표준 HDL new version 탄생
    - 표현 형식이 엄격하여 학계에서 주로 사용
  - Verilog HDL
    - C와 비슷한 syntax
    - Gateway Design System 사에서 개발 Cadence로 흡수
    - 약 70%이상의 기업체에서 사용
    - 실습 수업에 사용할 것은 **Verilog HDL!**

# Verilog HDL과 C 언어

- C 언어와 유사점
  - 문법은 매우 유사
  - 절차형(procedural) 및 순차형 실행은 동일
  - if~else문 , case 문 , for loop 문 등 사용
  - 연산자 : 논리 (&, |, ^..), 산술( +, -, \* /..), 비교 (<, >=, ==, ..)
- HDL 언어의 특성
  - 동시성(concurrent)
  - 병렬성 (parallel)
  - 추상화 (abstraction)

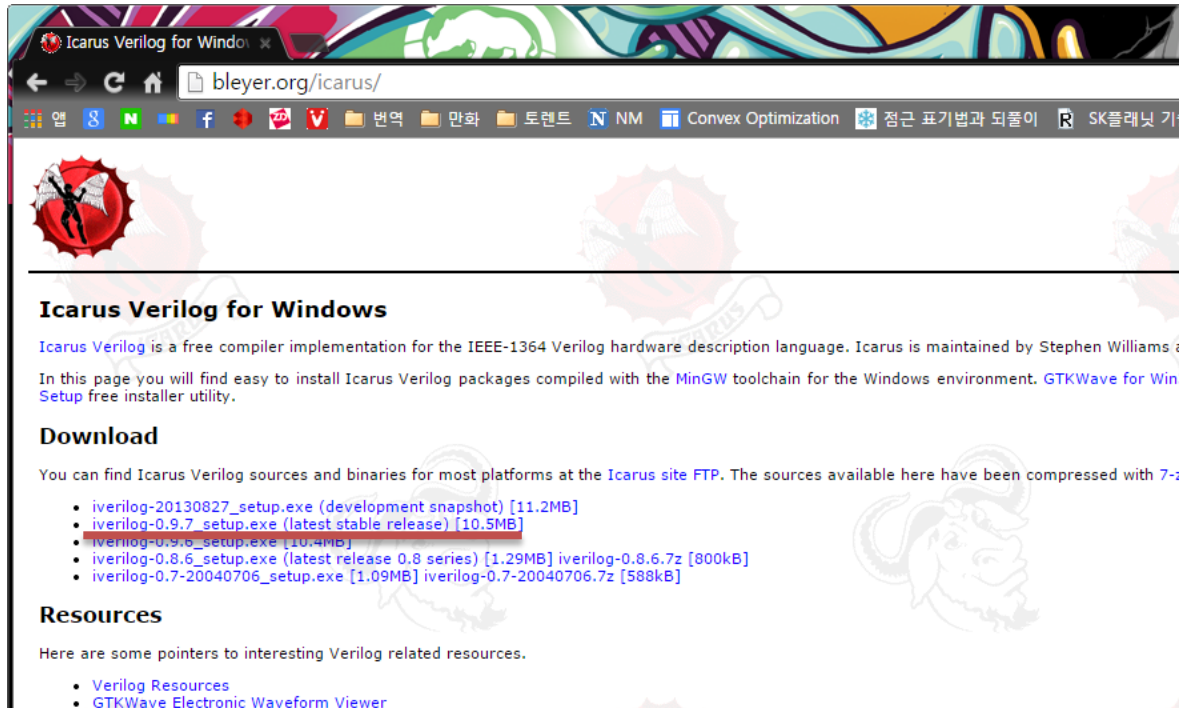
# HDL의 3가지 Modeling

- 구조적 (Structural) 모델링
  - 논리 게이트, 플립플롭 등을 사용한 연결도 표현
  - 기존 설계한 회로를 포함한 네트리스트(netlist) 사용
- 데이터플로우 (dataflow) 모델링
  - 데이터 이동을 표현
  - 연산자를 사용한 연속할당문
- 동작적(behavioral) 모델링
  - if~else, case, while, for 등과 같은 구문 사용
  - 인간의 사고에 가장 근접한 표현

# Simulator 설치

- 설치해야 할 것
  - 1. Icarus Verilog Simulator
  - 2. GTKWave
  - 3. Verilog Syntax Highlight가 되는 Editor

# Icarus Verilog Simulator 설치



- <http://bleyer.org/icarus/>
  - iverilog-0.9.7\_setup.exe (latest stable release)

# 환경변수 설정하기

- C:\liverilog\bin <--여기에 set\_path.bat 파일 만듦

```
@call :icarus_setup
```

```
@call :prompt_setup
```

```
@goto end
```

```
:icarus_setup
```

```
@echo Setting environment for Icarus Verilog Simulator
```

```
@set PATH=%PATH%;C:\Wiverilog\Wbin
```

```
@exit /B 0
```

```
:prompt_setup
```

```
@rem
```

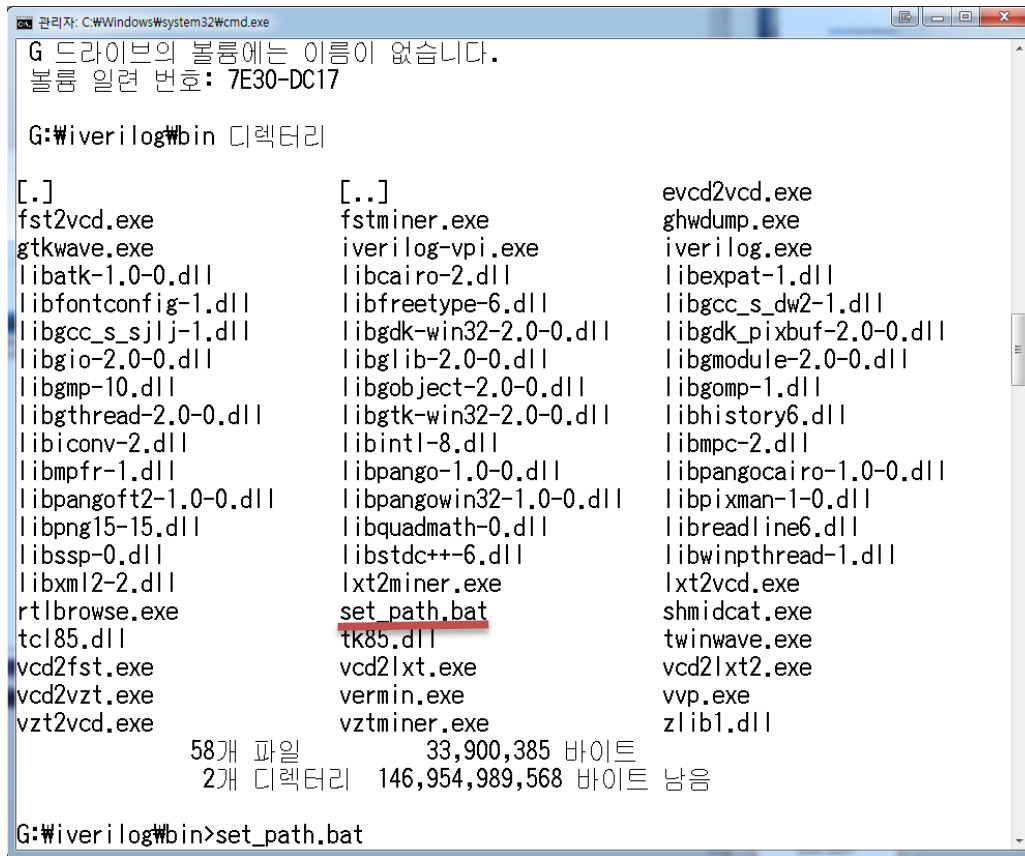
```
@rem PROMPT $P$_$+$S$G$$
```

```
@exit /B 0
```

```
:end
```

# 환경변수 설정하기

- cmd.exe 실행 후 set\_path.bat file 실행



```
관리자: C:\Windows\system32\cmd.exe
G 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 7E30-DC17

G:\iverilog\bin 디렉터리

[.]
fst2vcd.exe
gtkwave.exe
libatk-1.0-0.dll
libfontconfig-1.dll
libgcc_s_sjlj-1.dll
libgio-2.0-0.dll
libgmp-10.dll
libgthread-2.0-0.dll
libiconv-2.dll
libmpfr-1.dll
libpangoft2-1.0-0.dll
libpng15-15.dll
libssp-0.dll
libxml2-2.dll
rtlbrowse.exe
tcl85.dll
vcd2fst.exe
vcd2vzt.exe
vzt2vcd.exe

[.]
fstminer.exe
iverilog-vpi.exe
libcairo-2.dll
libfreetype-6.dll
libgdk-win32-2.0-0.dll
libglib-2.0-0.dll
libgobject-2.0-0.dll
libgtk-win32-2.0-0.dll
libintl-8.dll
libpango-1.0-0.dll
libpangowin32-1.0-0.dll
libquadmath-0.dll
libstdc++-6.dll
lxt2miner.exe
set_path.bat
tk85.dll
vcd2lxt.exe
vermin.exe
vztminer.exe

evcd2vcd.exe
ghwdump.exe
iverilog.exe
libexpat-1.dll
libgcc_s_dw2-1.dll
libgdk_pixbuf-2.0-0.dll
libgmodule-2.0-0.dll
libgomp-1.dll
libhistory6.dll
libmpc-2.dll
libpangocairo-1.0-0.dll
libpixmap-1-0.dll
libreadline6.dll
libwinpthread-1.dll
lxt2vcd.exe
shmidcat.exe
twinwave.exe
vcd2lxt2.exe
vvp.exe
zlib1.dll

58개 파일 33,900,385 바이트
2개 디렉터리 146,954,989,568 바이트 남음

G:\iverilog\bin>set_path.bat
```

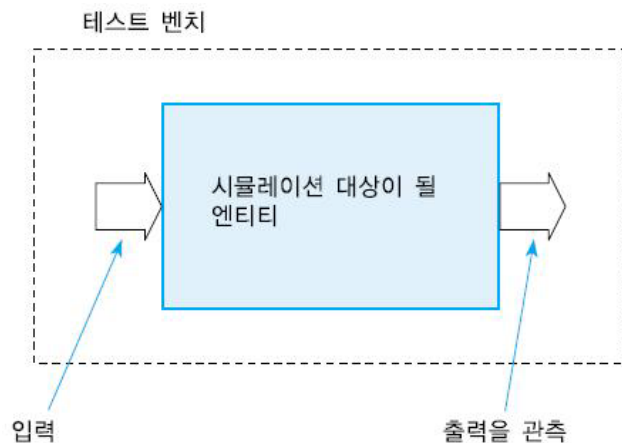


- 시뮬레이션(simulation)

- [홍내내는 것, 모의 실험]이라는 의미로, 여기에서는 회로 동작에 필요한 입력을 주고 이에 따른 출력의 변화를 관찰하는 것
- 각 엔티티에 입력을 주기 위해 필요한 기술

- 테스트 벤치 설정

- VHDL로 시뮬레이션을 대상 회로에 입력을 주거나 대상 회로의 출력을 관측하기 위한 테스트 벤치(test bench)(또는 테스트 벡터(test vector))를 설정



# Code 작성하기

- Add.v 라는 이름으로 적당한 editor를 사용해서 작성

```
1 module add(  
2     iX, // first input  
3     iY, // second input  
4     iC, // carry input  
5     oSUM, // sum output  
6     oC // carry output  
7 );  
8  
9 // input declaration  
10 input iX;  
11 input iY;  
12 input iC;  
13  
14 // output declaration  
15 output oSUM;  
16 output oC;  
17  
18 // port data type  
19 wire iX;  
20 wire iY;  
21 wire iC;  
22 wire oSUM;  
23 wire oC;  
24  
25 //code starts here!!  
26 assign {oC, oSUM} = iX + iY + iC;  
27  
28 endmodule  
29  
30
```

# Code 작성하기

- Testbench
  - testbench.v 라는 이름으로 코드 작성

```
1 `include "add.v"
2 module testbench;
3
4 reg x, y, ic;
5 wire sum, oc;
6
7 initial begin
8     $dumpfile("wave.vcd");
9     $dumpvars(0, testbench);
10    $monitor("x=%d y=%b ic=%b, sum=%b oc=%b", x,y,ic,sum,oc);
11
12    x=0;
13    y=0;
14    ic=0;
15
16    #5 x=0; y=0; ic=0;
17    #5 x=0; y=0; ic=1;
18    #5 x=0; y=1; ic=0;
19    #5 x=0; y=1; ic=1;
20    #5 x=1; y=0; ic=0;
21    #5 x=1; y=0; ic=1;
22    #5 x=1; y=1; ic=0;
23    #5 x=1; y=1; ic=1;
24 end
25
26 add ADD0 (
27     .iX(x),
28     .iY(y),
29     .iC(ic),
30     .oSUM(sum),
31     .oC(oc)
32 );
33
34 endmodule
35
```

# Code 작성하기

관리자: C:\Windows\system32\cmd.exe

```
D:\#VHDL#test1>iverilog testbench.v -o testbench
```

```
D:\#VHDL#test1>vvp testbench
```

```
VCD info: dumpfile wave.vcd opened for output.
```

```
x=0 y=0 ic=0, sum=0 oc=0
```

```
x=0 y=0 ic=1, sum=1 oc=0
```

```
x=0 y=1 ic=0, sum=1 oc=0
```

```
x=0 y=1 ic=1, sum=0 oc=1
```

```
x=1 y=0 ic=0, sum=1 oc=0
```

```
x=1 y=0 ic=1, sum=0 oc=1
```

```
x=1 y=1 ic=0, sum=0 oc=1
```

```
x=1 y=1 ic=1, sum=1 oc=1
```

```
D:\#VHDL#test1>
```

# Code 작성하기

- wave form 보기 (gtkwave 사용)

gtkwave wave.vcd

