

Verilog HDL 개요(3)

Kyung-Wook Shin
kwshin@kumoh.ac.kr

School of Electronic Eng.,
Kumoh National Institute of Technology

Verilog HDL

Verilog HDL 개요

K.W. SHIN

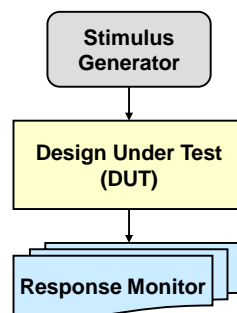
Verilog 모델링 예

2

□ 테스트벤치 모듈

❖ HDL 모델을 시뮬레이션하기 위한 Verilog 모듈

- DUT에 인가될 시뮬레이션 입력(stimulus)을 생성하는 구문
- 시뮬레이션 대상이 되는 모듈(Design Under Test; DUT)의 인스턴스
- 시뮬레이션 입력에 대한 DUT의 반응(response)을 관찰하는 구문



Verilog HDL

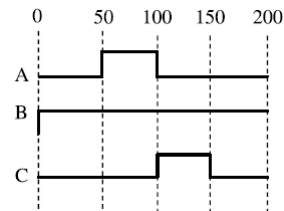
Verilog HDL 개요

K.W. SHIN

Verilog 모델링 예

3

```
module test_fix ();  
    reg A, B, C;  
  
    circuit c1(A, B, C, Out);  
  
    initial begin  
        A=0; B=1; C=0;  
        #50 A=1;  
        #50 A=0; C=1;  
        #50 C=0;  
        #50 $finish;  
    end  
endmodule
```



Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 모델링 예

4

```
module tb_ha ;  
    reg a, b;  
  
    half_adder U0_half_adder(a, b, sum, cout);  
  
    initial begin  
        a = 0; b = 0;  
        #10 a = 1;  
        #10 a = 0;  
        b = 1;  
        #10 a = 1;  
        #10 a = 0;  
        b = 0;  
    end  
endmodule
```

반가산기의 테스트벤치 예

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 모델링 예

5

```
module tb_dff_sr ;  
    reg set, reset, clock, din;  
  
    dff_sr U0_dff_sr (set, reset, clock, din, q, qb);  
  
    initial begin  
        clock = 1'b0;  
        forever begin  
            #10 clock = ~clock;  
        end  
    end  
  
    initial begin  
        forever begin  
            din = 0; reset = 1; set = 1;  
            #15 din = 1;  
            #30 reset = 0;  
            #10 reset = 1;  
            #25 din = 0;  
            #20 set = 0;  
            #10 set = 1;  
            #30 din = 0;  
        end  
    end  
endmodule
```

모듈 dff_sr의 테스트벤치 예

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 어휘 규칙

6

□ 어휘 토큰 (lexical tokens)

- ❖ 여백(white space)
- ❖ 주석(comment)
- ❖ 연산자(operator)
- ❖ 수(number)
- ❖ 문자열(string)
- ❖ 식별자(identifier)
- ❖ 키워드(keyword)

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 어휘 규칙

7

□ 여백(white space)

- ❖ 빈칸(space), 탭(tab), 줄바꿈
- ❖ 어휘 토큰들을 분리하기 위해 사용되는 경우를 제외하고는 무시
- ❖ 공백(blank)과 탭은 문자열에서 의미 있게 취급

□ 주석(comment)

- ❖ HDL 소스코드의 설명을 위해 사용되며, 컴파일과정에서 무시됨
- ❖ 단일 라인 주석문; `//` 로 시작되어 해당 라인의 끝까지
- ❖ 블록 주석문; `/* ~ */` 로 표시
 - 블록 주석문은 내포(nested)될 수 없음

□ 연산자(operator)

- ❖ 단항 연산자, 2항 연산자, 3항 연산자

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 어휘 규칙

8

□ 수 표현 (number representation)

- ❖ 정수형(integer) ; 10진수, 16진수, 8진수, 2진수

- ❖ 형식 : `[size_constant]`<base_format> <number_value>`

- `[size_constant]` : 값의 비트 크기를 나타내는 상수
 - ✓ 0이 아닌 unsigned 10진수가 사용되며, 생략될 수 있음
 - ✓ **unsigned 수** (즉, 단순 10진수 또는 비트 크기가 지정되지 않은 수)는 **32비트로 표현됨**
 - ✓ 상위 비트가 x(unknown) 또는 z(high-impedance)인 unsigned unsigned 상수는 그 상수가 사용되는 수식의 비트 크기만큼 확장됨
- `'base_format` : 밑수(base)를 지정하는 문자(d, D, h, H, o, O, b, B)
 - ✓ signed를 나타내기 위해 문자 s 또는 S가 함께 사용될 수 있음
- `number_value` : unsigned 숫자를 사용하여 값을 표현
 - ✓ 'base_format에 적합한 숫자들로 구성
- base_format과 number_value 사이에 + 또는 - 부호 **사용 불가**

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 어휘 규칙

9

□ 수 표현 (number representation)

- ❖ 비트 크기와 밑수를 갖지 않는 단순 10진수는 **signed** 정수로 취급
- ❖ 부호 지정자 없이 밑수만 지정되면 **unsigned** 정수로 취급
- ❖ 밑수 지정자와 부호 지정자 **s**가 함께 사용되면 **signed** 정수로 취급
 - 부호 지정자 **s**는 비트 패턴에는 영향을 미치지 않으며, 비트 패턴의 해석에만 영향을 미침
- ❖ 음수는 2의 보수(**2's complementary**) 형식으로 표현됨
- ❖ 지정된 비트 크기보다 **unsigned** 수의 크기가 작은 경우
 - **MSB** 왼쪽에 0이 삽입
 - **MSB**가 **x** 또는 **z**이면, **x** 또는 **z**가 왼쪽에 삽입
- ❖ 값에 물음표(?)가 사용되면 **z**로 취급
- ❖ 첫번째 문자를 제외하고는 밑줄(**underscore**)이 사용될 수 있으며, 이는 수의 가독성(**readability**)을 좋게 함

Verilog 어휘 규칙

10

Number	# of Bits	Base	Dec. Equiv.	Stored
10	32	Decimal	10	00....01010
2'b10	2	Binary	2	10
3'd5	3	Decimal	5	101
8'o5	8	Octal	5	00000101
8'ha	8	Hex	10	00001010
3'b5	Invalid!			
3'b01x	3	Binary	-	01x
12'hx	12	Hex	-	xxxxxxxxxxxx
8'b0000_0001	8	Binary	1	00000001
8'bx01	8	Binary	-	xxxxxx01
'bz	Unsize	Binary	-	zz...zz(32bits)
8'HAD	8	Hex	173	10101101

Verilog 어휘 규칙

11

예 1.4.1 unsized 상수

```
659          // a decimal number
'h837FF      // a hexadecimal number
'o7460       // an octal number
4af          // illegal (hexadecimal format requires 'h)
```

예 1.4.2 sized 상수

```
4'b1001      // a 4-bit binary number
5'D3         // a 5-bit decimal number
3'b01x       // a 3-bit number with the LSB unknown
12'hx        // a 12-bit unknown number
16'hz        // a 16-bit high-impedance number
```

Verilog 어휘 규칙

12

예 1.4.4 밑줄을 사용한 수의 표현

```
27_195_000
16'b0011_0101_0001_1111
32'h 12ab_f001
```

예 1.4.5 MSB 자동 삽입

```
reg [11:0] a, b, c, d;
initial begin
    a = 'hx;      // yields xxx
    b = 'h3x;     // yields 03x
    c = 'hz3;     // yields zz3
    d = 'h0z3;    // yields 0z3
end
```

```
wire [3:0] neg, sig;
assign neg = -4'b0101; // 1011
assign sig = -5;       // 1011
```

Verilog 어휘 규칙

13

- ❖ 실수형(real) ; IEEE Std. 754-1985(IEEE standard for double-precision floating-point number)

예 1.4.6

```
1.2
2394.26331
1.2E12
1.30e-2
0.1e-0
29E-2
236.123_763_e-12 (underscores are ignored)
```

예 1.4.7

문법적 오류

```
.12
9.
4.E3
.2e-7
```

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 어휘 규칙

14

□ 문자열(string)

- ❖ 이중 인용 부호(" ") 사이에 있는 일련의 문자들
- ❖ 단일 라인에 존재해야 하며, 여러 라인에 걸친 문자열은 사용 불가
- ❖ 8비트 ASCII 값으로 표현되는 **unsigned** 정수형 상수로 취급
- ❖ 문자열 변수는 **reg**형의 변수이며, 문자열 내의 문자 수에 8을 곱한 크기의 비트 폭을 가짐

예 1.4.8

```
reg [8*12:1] string_var;

initial begin
    string_var = "Hello world!";
end
```

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 어휘 규칙

15

□ 식별자(identifier)

- ❖ 객체에 고유의 이름을 지정하기 위해 사용
- ❖ 대소문자를 구별하여 인식
- ❖ 가독성을 위해 밑줄 사용 가능
- ❖ 단순 식별자; Alphabet, 숫자, 기호 \$, 밑줄 등으로 구성
 - 첫번째 문자는 숫자나 기호 \$ 사용 불가, 문자 또는 밑줄만 사용
- ❖ 확장 식별자(escaped identifier);
 - \ (back slash)로 시작되며, 여백(빈칸, 탭, 줄바꿈) 등으로 끝남
 - 프린트 가능한 ASCII 문자들을 식별자에 포함시키는 수단을 제공

□ 키워드(keyword)

- ❖ Verilog 구성 요소를 정의하기 위해 미리 정의된 식별자
- ❖ 확장문자가 포함된 키워드는 키워드로 인식되지 않음

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 어휘 규칙

16

예 1.4.10

유효한 식별자의 예

```
shiftreg_a
busa_index
error_condition
merge_ab
_bus3
n$657
```

예 1.4.11

확장 식별자의 예

```
\busa+index
\~clock
\***error-condition***
\net1/\net2
\{a,b}
\ a*(b+c)
```

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 어휘 규칙

17

Verilog keyword (일부)

always	if	release
and	ifnone	repeat
assign	incdir	rnmos
automatic	include	rpmos
begin	initial	rtran
buf	inout	rtranif0
bufif0	input	rtranif1
bufif1	instance	scalared
case	integer	showcancelled
casex	join	signed
casez	large	small
cell	liblist	specify
cmos	library	specparam
config	localparam	strong0
deassign	macromodule	strong1
default	medium	supply0
defparam	module	supply1
design	nand	table
disable	negedge	task
edge	nmos	time
else	nor	tran
end	not	tranif0
endcase	notif0	tranif1

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Verilog 어휘 규칙

18

시스템 task와 시스템 함수

- ❖ 문자 \$로 시작되는 이름은 시스템 task 또는 시스템 함수

컴파일러 지시어(compiler directive)

- ❖ 문자 ' (ASCII 값 60, accent grave 또는 open quote)로 시작
- ❖ Verilog 소스코드의 컴파일 과정에 영향을 미침
- ❖ 특정 파일에 있는 컴파일러 지시어는 다수 파일의 컴파일 동작을 제어할 수 있음

```
$display("display a message");  
$finish;
```

```
`define wordsize 8
```

Verilog HDL

Verilog HDL 개요

K.W. SHIN

Keep in mind

19

