

구조적 모델링(1)

Kyung-Wook Shin
kwshin@kumoh.ac.kr

School of Electronic Eng.,
Kumoh National Institute of Technology

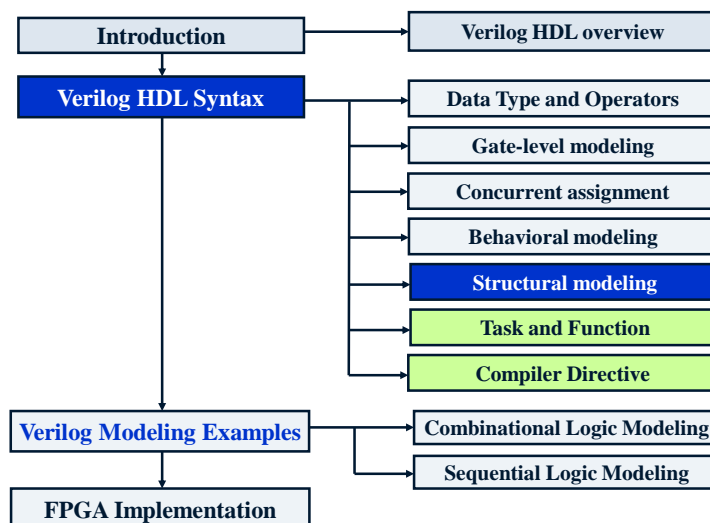
Verilog HDL

구조적 모델링

K. W. SHIN

Learning Map

2



Verilog HDL

구조적 모델링

K. W. SHIN

6.1 모듈

3

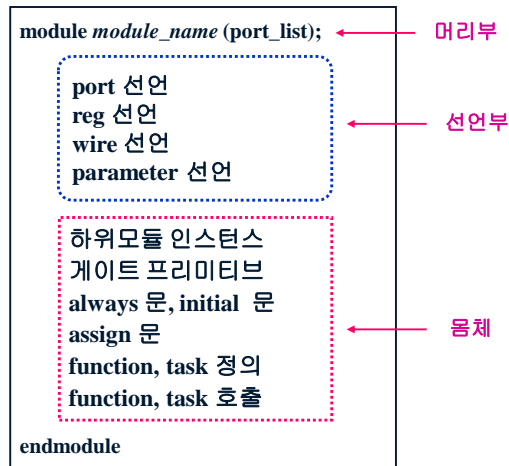


그림 6.1 Verilog 모듈의 구성

Verilog HDL

구조적 모델링

K. W. SHIN

6.1.1 모듈 포트 선언

4

□ 모듈 포트 선언

- ❖ 포트목록에 나열된 포트들은 선언부에서 포트선언을 통해 input, output, inout(양방향)으로 선언
- ❖ signed와 범위에 대한 정의를 포함한 포트에 관한 모든 다른 속성들이 포트선언에 포함될 수 있음

```
inout_declaration ::=
    inout [ net_type ][ signed ][ range ] list_of_port_identifiers
input_declaration ::=
    input [ net_type ][ signed ][ range ] list_of_port_identifiers
output_declaration ::=
    output [ net_type ][ signed ][ range ] list_of_port_identifiers
    | output [ reg ][ signed ][ range ] list_of_port_identifiers
    | output reg [ signed ][ range ] list_of_variable_port_identifiers
    | output [ output_variable_type ] list_of_port_identifiers
    | output output_variable_type list_of_variable_port_identifiers
list_of_port_identifiers ::=
    port_identifier { , port_identifier }
```

Verilog HDL

구조적 모델링

K. W. SHIN

6.1.1 모듈 포트 선언

5

```
input  aport;    // First declaration - okay.
input  aport;    // Error - multiple port declaration
output aport;    // Error - multiple port declaration
```

예 6.1.1 기본적인 모듈 포트선언의 예

```
module test(a, b, e, f);
  input      [7:0] a;
  input signed [7:0] b;
  output      [7:0] e;
  output signed [7:0] f;
  reg         [7:0] e;
```

6.1.2 모듈포트 선언목록

6

□ 포트 선언목록

- ❖ 포트 선언목록에서 포트를 선언
- ❖ 포트 선언목록으로 선언된 포트들은 모듈의 선언부에서 재선언되지 않음

```
module test( input [7:0] a,
             input signed [7:0] b,
             output reg [7:0] e,
             output signed [7:0] f );
// illegal to redeclare any ports of the module in the body of the module.
```

[코드 6.1] 포트선언목록에 의한 모듈포트 선언

6.2 모듈 인스턴스

7

□ 구조적 모델링

```
Module_name [#(param)] Instance_name [[N:M]] (port_mapping);
```

- ❖ 다른 모듈의 인스턴스와 포트 매핑을 통한 모델링
- ❖ 범위 지정을 통한 인스턴스 배열의 생성 가능
- ❖ 모듈 인스턴스 이름은 생략할 수 없음
 - 게이트 프리미티브의 인스턴스 이름은 생략 가능

□ 포트순서에 의한 포트 매핑

- ❖ 모듈의 포트목록에 나열된 포트 순서와 1 : 1로 대응되어 연결
- ❖ 포트에 연결되는 신호가 없는 경우에는 해당 위치를 빈칸으로 남겨 둔다

□ 포트이름에 의한 포트 매핑

- ❖ 포트 이름과 그 포트에 연결되는 신호 이름을 명시적으로 지정
- ❖ 포트의 비트 선택, 부분 선택, 결합 등을 사용할 수 없음

```
.port_name([expression])
```

Verilog HDL

구조적 모델링

K. W. SHIN

6.2 모듈 인스턴스

8

예 6.2.1

순서에 의한 포트 매핑

```
module topmod;
    wire [1:0] v, w;

    modB b1(v[0], v[1], w[0], w[1]);
endmodule
```

코드 6.2

```
module modB(wa, wb, c, d);
    inout wa, wb;
    input c, d;

    tranif1      g1(wa, wb, cinvert);
    not #(2, 6) n1(cinvert, int);
    and #(6, 5) g2(int, c, d);
endmodule
```

예 6.2.3

이름에 의한 포트 매핑

```
module topmod;
    wire [1:0] v, w;

    modB b1(.wb(v[1]), .wa(v[0]), .d(w[1]), .c(w[0]));
endmodule
```

코드 6.3

Verilog HDL

구조적 모델링

K. W. SHIN

6.2 모듈 인스턴스

9

예 6.2.4

동일 포트에 대한 다중 연결 오류

```
module test;
  a U0(.i(a),
      .i(b), // illegal connection of input port twice.
      .o(c),
      .o(d), // illegal connection of output port twice.
      .e(e),
      .e(f) // illegal connection of inout port twice.
  );
endmodule
```

코드 6.4

Verilog HDL

구조적 모델링

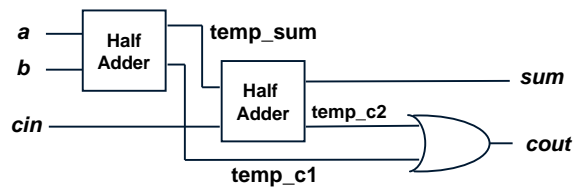
K. W. SHIN

6.2 모듈 인스턴스

10

예 6.2.5

1비트 전가산기 회로의 구조적 모델링



```
module half_adder(a, b, sum, cout);
  input a, b;
  output sum, cout;
  wire cout_bar; //생략 가능 (1-bit wire)

  xor U0 (sum, a, b);
  nand (cout_bar, a, b); // 인스턴스 이름 생략 가능 (gate primitive)
  not U1 (cout, cout_bar);
endmodule
```

반가산기 모듈

코드 6.5-(a)

Verilog HDL

구조적 모델링

K. W. SHIN

6.2 모듈 인스턴스

11

예 6.2.5 1비트 전가산기 회로의 구조적 모델링

```
module full_adder(a, b, cin, sum, cout);
    input a, b, cin;
    output sum, cout;
    wire temp_sum, temp_c1, temp_c2; //생략 가능

    // half_adder 모듈의 instantiation
    half_adder u0(a, b, temp_sum, temp_c1); // 순서에 의한 포트 연결
    half_adder u1(.a(temp_sum),
                  .b(cin),
                  .sum(sum),
                  .cout(temp_c2) ); // 이름에 의한 포트 연결
    or u2(cout, temp_c1, temp_c2); // 게이트 프리미티브 인스턴스

endmodule
```

코드 6.5-(b)

Verilog HDL

구조적 모델링

K. W. SHIN

6.2 모듈 인스턴스

12

예 6.2.5 1비트 전가산기 회로의 구조적 모델링

```
// 1비트 full adder 모듈의 시뮬레이션 testbench
module tb_full_adder ;
    reg a, b, cin; // initial 블록에서 값을 받으므로 reg로 선언
    integer k;

    // full_adder 모듈의 instantiation
    full_adder U0(a, b, cin, sum, cout);

    // 시뮬레이션을 위한 파형 생성
    initial begin
        forever
            for(k = 0; k < 8; k = k+1) begin
                cin = k/4;
                b = (k%4)/2;
                a = k%2;
                #10;
            end
        end
    end
endmodule
```

[코드 6.6] [코드 6.5]의 시뮬레이션 테스트벤치

코드 6.6

Verilog HDL

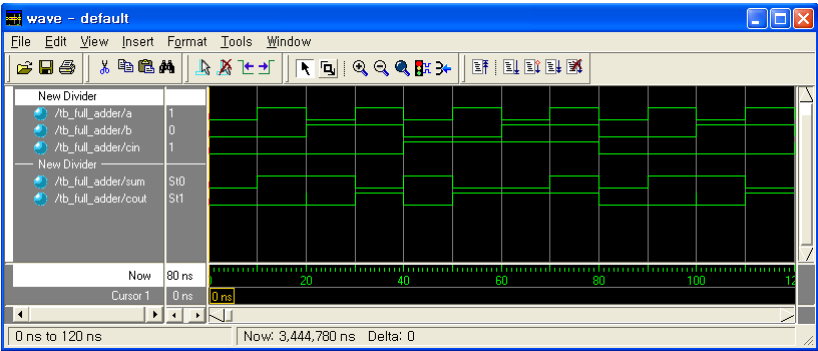
구조적 모델링

K. W. SHIN

6.2 모듈 인스턴스

13

예 6.2.5 1비트 전가산기 회로의 구조적 모델링



[그림 6.6] [코드 6.6]의 시뮬레이션 결과

6.2 모듈 인스턴스

14

예 6.2.6 인스턴스 배열을 이용한 구조적 모델링

```
module bus_driver(busin, bushigh, buslow, enh, enl);
    input [15:0] busin;
    input enh, enl;
    output [7:0] bushigh, buslow;

    driver busar3(busin[15:12], bushigh[7:4], enh);
    driver busar2(busin[11:8], bushigh[3:0], enh);
    driver busar1(busin[7:4], buslow[7:4], enl);
    driver busar0(busin[3:0], buslow[3:0], enl);
endmodule
```

코드 6.7-(a)

```
driver busar[3:0] (.in(busin),
    .out({bushigh, buslow}),
    .en({enh, enh, enl, enl}));
endmodule
```

모듈 인스턴스의 배열을 이용

코드 6.7-(b)

6.2 모듈 인스턴스

15

□ 실수형 값의 포트 연결

- ❖ **real** 자료형은 직접 포트에 연결될 수 없음
- ❖ 시스템 함수인 **\$realtobits**와 **\$bittoreal**를 통해 **real** 자료형을 비트 자료형으로 변환한 후 모듈 포트에 적용

예 6.2.7

```
module driver(net_r);  
    output net_r;  
    real r;  
    wire [64:1] net_r = $realtobits(r);  
endmodule
```

코드 6.8

```
module receiver(net_r);  
    input net_r;  
    wire [64:1] net_r;  
    real r;  
  
    initial  
        r = $bittoreal(net_r);  
endmodule
```

코드 6.9