

Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

#73

학생 : 김시윤

강사 : Innova Lee(이 상훈)

VHDL 기본개념

구분어	& ‘ () * + , - . / : ; < = > ` [] ? @
복합 구분어	=> ** := /= <= <> ?? ?= ?/= ?< ?<= ?> ?>= << >>

예제 해석

```
process( A , B ) – a,b 순차적으로 접근한다.  
begin  
  
    S <= A xor B; – S 의 결과는 A xor B 이다.  
    C <= A and B; – C 의 결과는 A and B 이다.  
end process;
```

예약어

Process begin abs access after alias all and architecture configuration exit else if in for
등 여러가지가 있다.

식별어

식별어는 엔티티나 아키텍처의 이름, 신호나 변수 상수의 이름 등 설계자가 명명해야 할 다양한 문법 구조의 이름이나 예약어로 사용된다.

ex)

COUNT	A	carry_out	DCT	Encoder
FrameCount	x9	ADD_A_B	X	

리터럴

상수 선언에서 사용되는 상수 값 자체를 리터럴 이라고 부른다. VHDL 언어에서는 프로그램 내에 사용되는 상수 값을 선언하는 방식을 정의해놓고 있으며 정수, 실수, 문자, 문자열, 비트열 등 다양한 종류의 상수 값을 정의할 수 있다.

ex)

```
constant HTOTAL_WIDTH : integer := 512; //정수  
constant MAX_BIN : string := B"1111_1111_1111"; // 비트열  
constant MAX_HEX : BIT_VECTOR := X"FFFF"; //비트단위 변수  
type VOLTAGE_RANGE is range -15.0to 15.0; //전압범위?
```

추상 리터럴

1)

추상 리터럴은 실수형 리터럴(real literal) 과 정수형 리터럴(integer literal)로 구분할 수 있다
실수형 리터럴은 소숫점을 포함하고 정수형 리터럴은 소숫점이 없는 추상적 리터럴을 의미한다.

2)

추상리터럴은 10 진수 리터럴(decimal literal) 과 2 진수에서 16 진수까지의 숫자 형태로 표기가 가능한 진수 리터럴(based literal)로 구분이 가능하다

자릿수를 일치시키거나 가독성을 위해서 숫자 앞부분에 0 을 추가할 수도 있으며, 밑줄을 사용하여 자릿수가 많은 수의 해독성을 높일 수 있다. 공백문자는 분리아이므로 추상리터럴에서 사용하지 말것!

십진수 리터럴

ex)

15 01E6 123_456_789 – 정수형 리터럴의 예
15.0 0.0 0.123 3.141_592 – 실수형 리터럴 예
1.234E-15 1.2e+6 5.0E3 --지수 부분을 포함한 실수형 리터럴 예

ex)

```
constant HALF_PERIOD_IN : time := 5.0E3ns;
```

진수 리터럴

– 십진수 254 와 같은 값
2#1111_1110# : 이진수 11111110 = 254
16#FE# : 16 진수 FE = 254
016#0fe# : 16 진수 FE = 254

– 십진수 실수 4095.0 과 같은 값
16#F.FF#E+2
2#1.1111_1111_111#E11

문자 리터럴

문자 리터럴의 예 :

‘a’ – 영문자 소문자
‘A’ – 영문자 대문자
‘@’ – commercial at 기호
‘ ‘ ‘ – apostrophe 기호
‘ ‘ – space
“” – 숫자

문자 리터럴 사용 예

```
when "1001" => result := '9';  
when "1010" => result := 'A';
```

문자열 리터럴의 사용 예

```
report "File Processing is completed!";
```

비트열 리터럴의 사용 예

```
constant c1: STRING := B"1111_1111_1111";  
constant c2: BIT_VECTOR := X"FFF";
```