



VHDLの基本記述

例題1:AND回路の記述1

例題2:AND回路の記述2

例題1

And_Sample1

論理積 (AND) を行う回路

VHDLの構成

- library部
- entity部

回路を出入りする

信号を定義

- architecture部

回路の動作を記述

VHDL

library部

entity部

architecture部

例題1

library部

```
library IEEE;  
use IEEE.std_logic_1164.all;  
use IEEE.std_logic_arith.all;  
use IEEE.std_logic_signed.all;
```

ライブラリ IEEE : 標準的な各種宣言

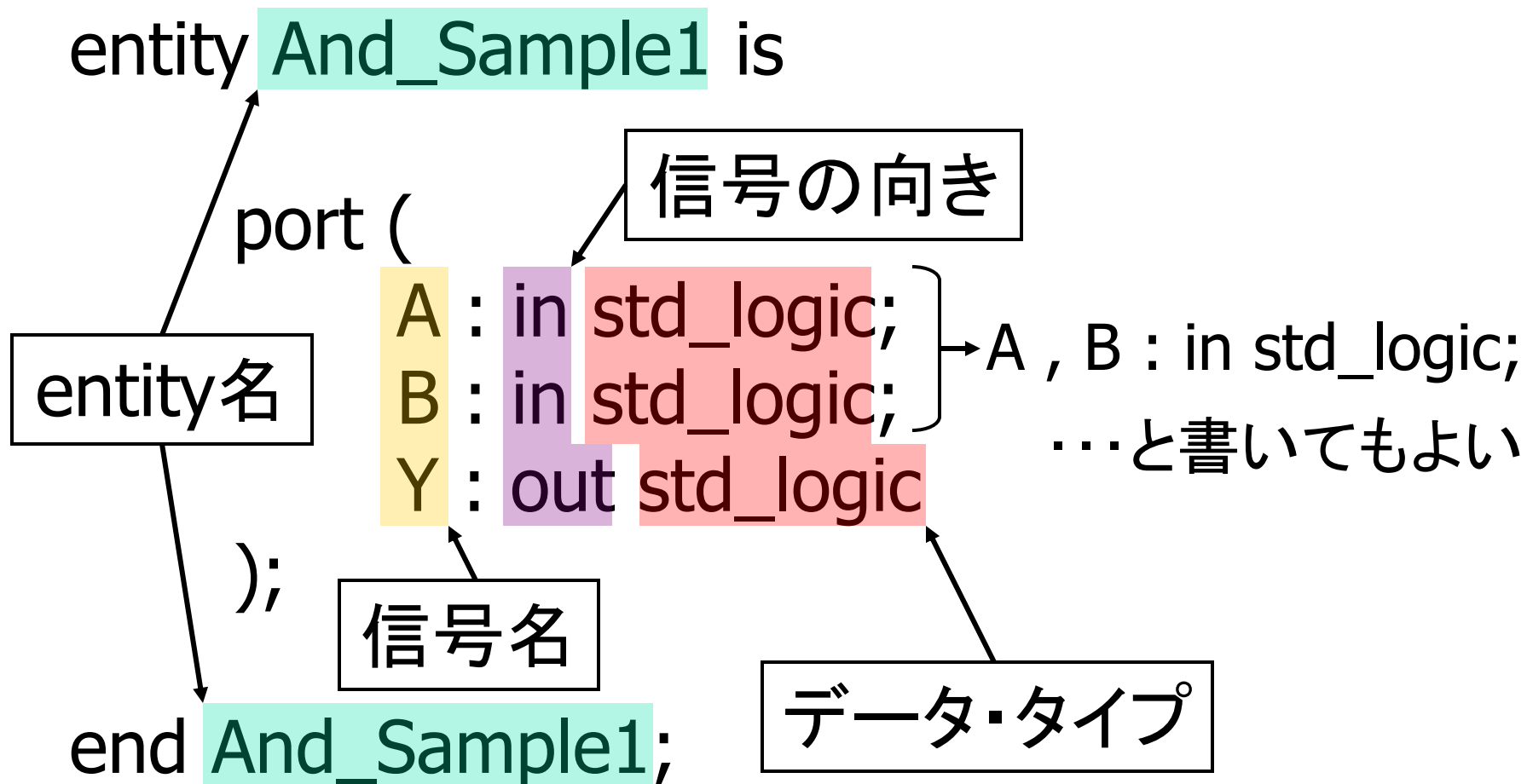
std_logic_1164 : 基本関数

std_logic_arith : 算術演算

std_logic_signed : 符号付き演算

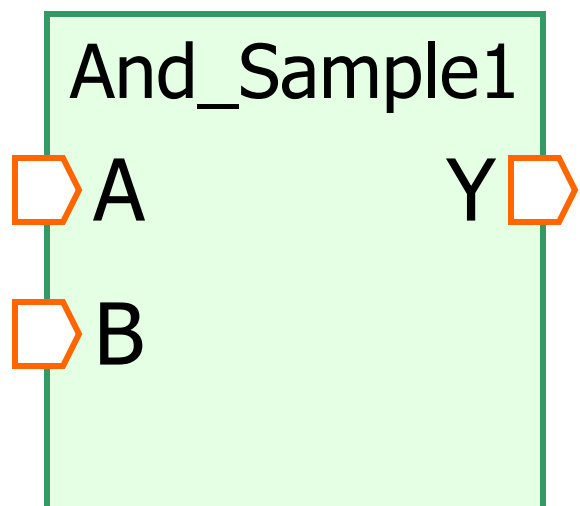
例題1

entity部 (VHDL)



例題1

entity部(モデル)



entity宣言イメージ

何の機能をする回路かは

この段階では不明だが

回路の名前は何で、

どのような型の信号が

入力・出力・入出力されるか
などを定義する

例題1

architecture部 (VHDL)

architecture RTL of And_Sample1 is

begin

Y <= A and B;

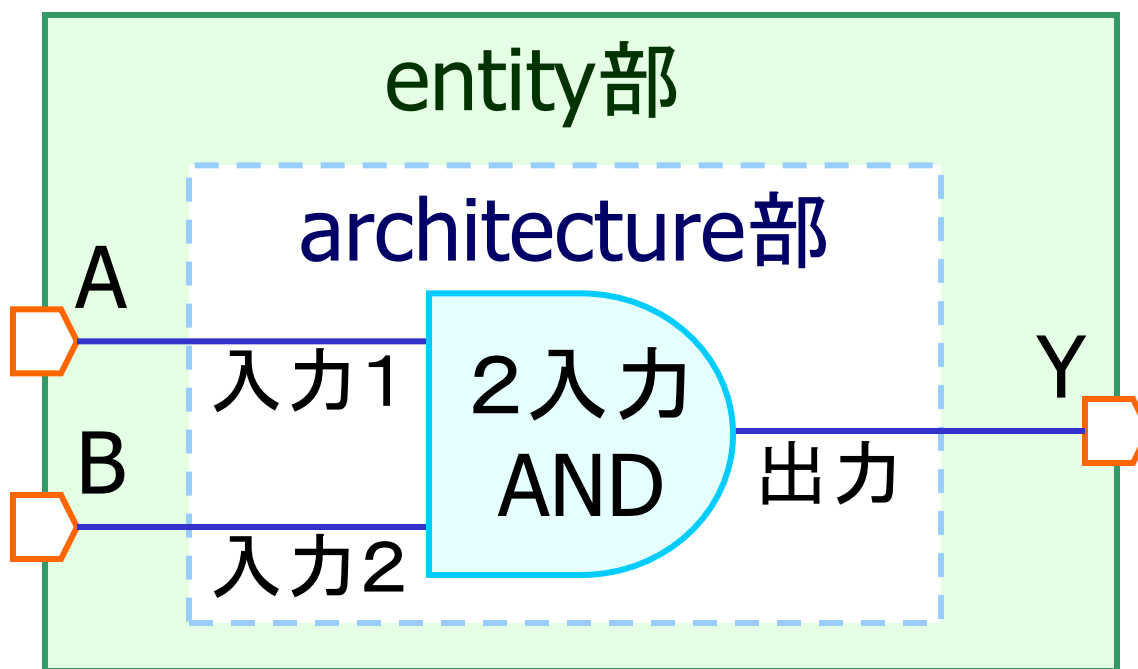
end RTL;

entity名

architecture名

例題1

architecture部 (モデル)



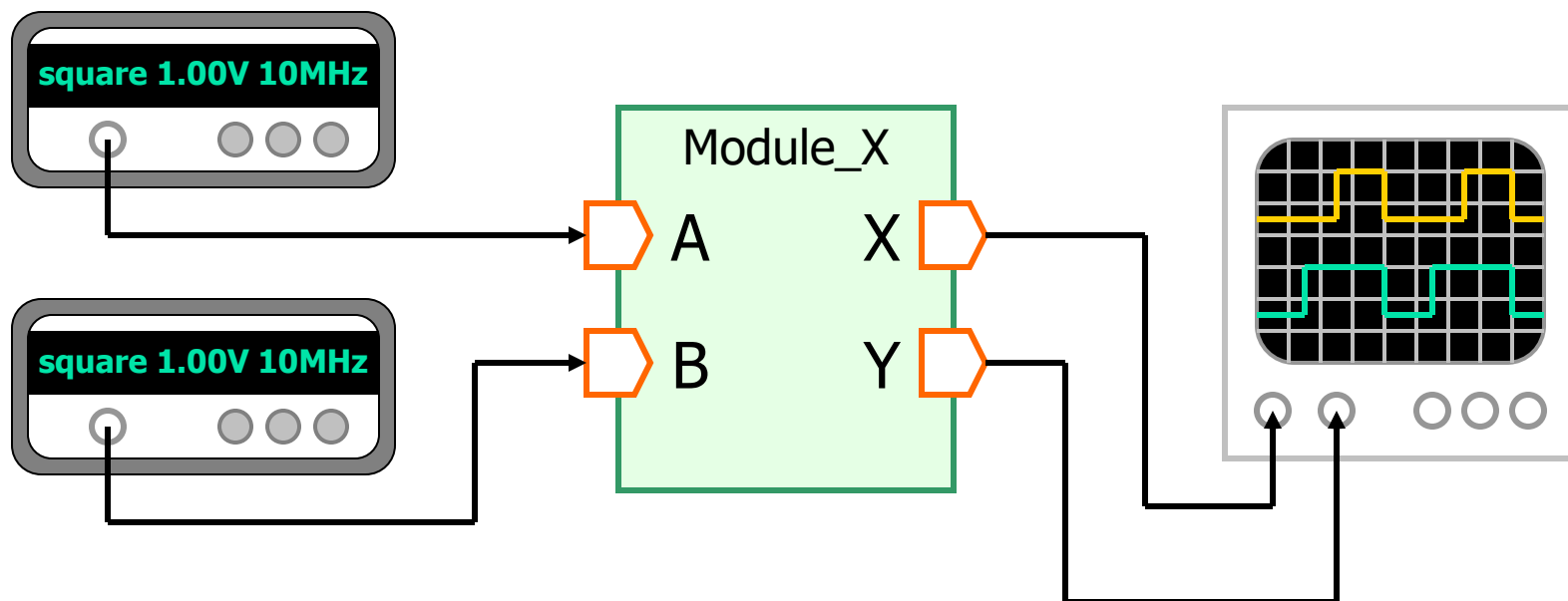
回路としてどのような中身になっているか、
どのような働きをするのかを定義する

例題1

テスト・ベンチ (tb_〇〇)

シミュレーションのための、

「回路に信号を入力するための」回路
シミュレーション用で、FPGAには搭載しない



例題2

And_Sample2

論理積 (AND) を行う回路

- 回路から出力される結果は

And_Sample1.vhdと同じ

- 異なるのは回路動作の記述方法

2の方の記述方法: 動作記述

- 回路がどのように振舞うかを

文章的 (プログラムの) に表す

- 効率的に回路の動作を表すことができる

例題2

architecture部 : process文

process文ラベル

LOGICAL_AND : process(A, B)

begin

センシティブティ・リスト

if ((A = '1') and (B = '1')) then

.....

end if;

end process;

if文などの条件分岐文
が利用可能になる

例題2

余談(process文:順次処理)

```
INPUT_A : process  
begin
```

```
    if done = false then
```

```
        a <= '0';
```

```
        wait for 100 ns;
```

```
        a <= '1';
```

```
        wait for 100 ns;
```

```
    else
```

```
        wait;
```

```
    end if;
```

```
end process;
```

センシティブティ・リストなし※

上から順に
処理される

※ センシティブティ・リストと
wait文は同時使用できない

例題2

動作記述

```
if ((A = '1') and (B = '1')) then  
    Y <= '1';
```

```
else  
    Y <= '0';  
end if;
```

真理値表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

その他の場合

信号A、Bの
両方が'1'の場合

例題2

動作記述(コメント・アウト部)

```
-- if ((A = '0') or (B = '0')) then
--     Y <= '0';
-- else
--     Y <= '1';
-- end if;
```

コメント

その他の場合

真理値表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

信号A、Bのうち
片方が'0'の場
合