



階層構造記述

例題4: SW状態('0'、'1')表示回路の

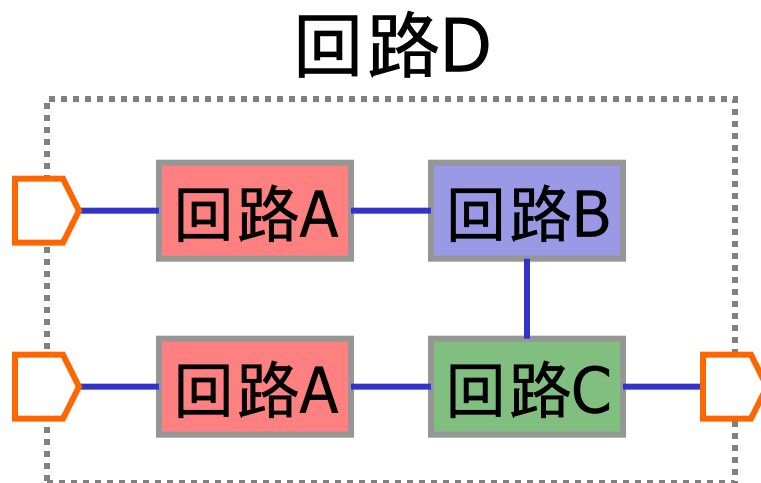
階層構造記述

練習4: 加算結果出力回路の

階層構造記述・実装

階層構造記述とは

- これまで作成した回路など、
ある機能を有する回路を
1つのモジュールとして扱い、
他の回路に組込む記述方法



回路D:

最上位層モジュール

回路A、B、C:

下位層モジュール

※ 何層でも作れる

例題4

Logical_01

- 最上位層モジュール

Logical_01: スイッチのON/OFF状態を
7セグメントLEDの0/1表示であらわす回路

- 下位層モジュール

Logic_to_Int: 論理 '0' / '1' (std_logic) を
数字 (integer: 整数型) の0/1に変換する

Int_to_Seg7:

数字0/1を7セグメントLEDに表示

例題4

architecture部1

```
component Logic_to_Int
```

```
port (  
    ...  
);
```

```
end component;
```

```
component Int_to_Seg7
```

```
port (  
    ...  
);
```

```
end component;
```

```
signal data_intd : integer range 0 to 1;
```

component名

信号宣言: signal
signal宣言の後に
信号名、(:、)
データ・タイプ
を記述する

componentのport記述は
entityと一致させる

整数型(32bitまで)、**範囲**はsignal宣言時に必須

例題4

architecture部2

```
LTI : Logic_to_Int port map (  
    SW => SW,  
    DATA => data_intd
```

```
);
```

インスタンス名

component名

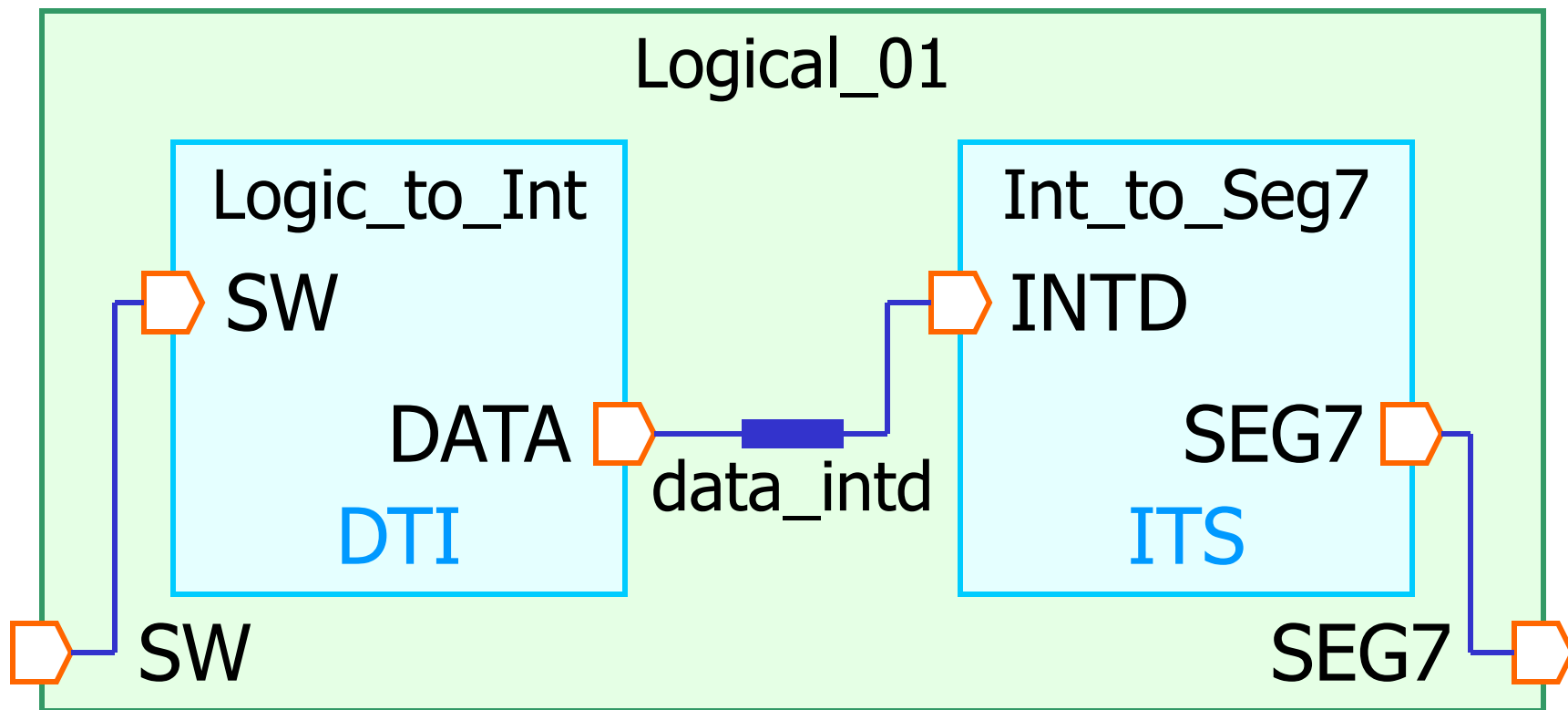
```
ITS : Int_to_Seg7 port map (  
    INTD => data_intd,  
    SEG7 => SEG7
```

```
);
```

接続の記述

左: インスタンスの入出力
右: インスタンス外の信号

例題4 回路モデル



練習4

Calc_Add(1)

2つで1組のスイッチであらわされる2信号の
和を7セグメントLEDで表示する回路

- 最上位層モジュール

Calc_Add: 上記を実現するよう組合わせる

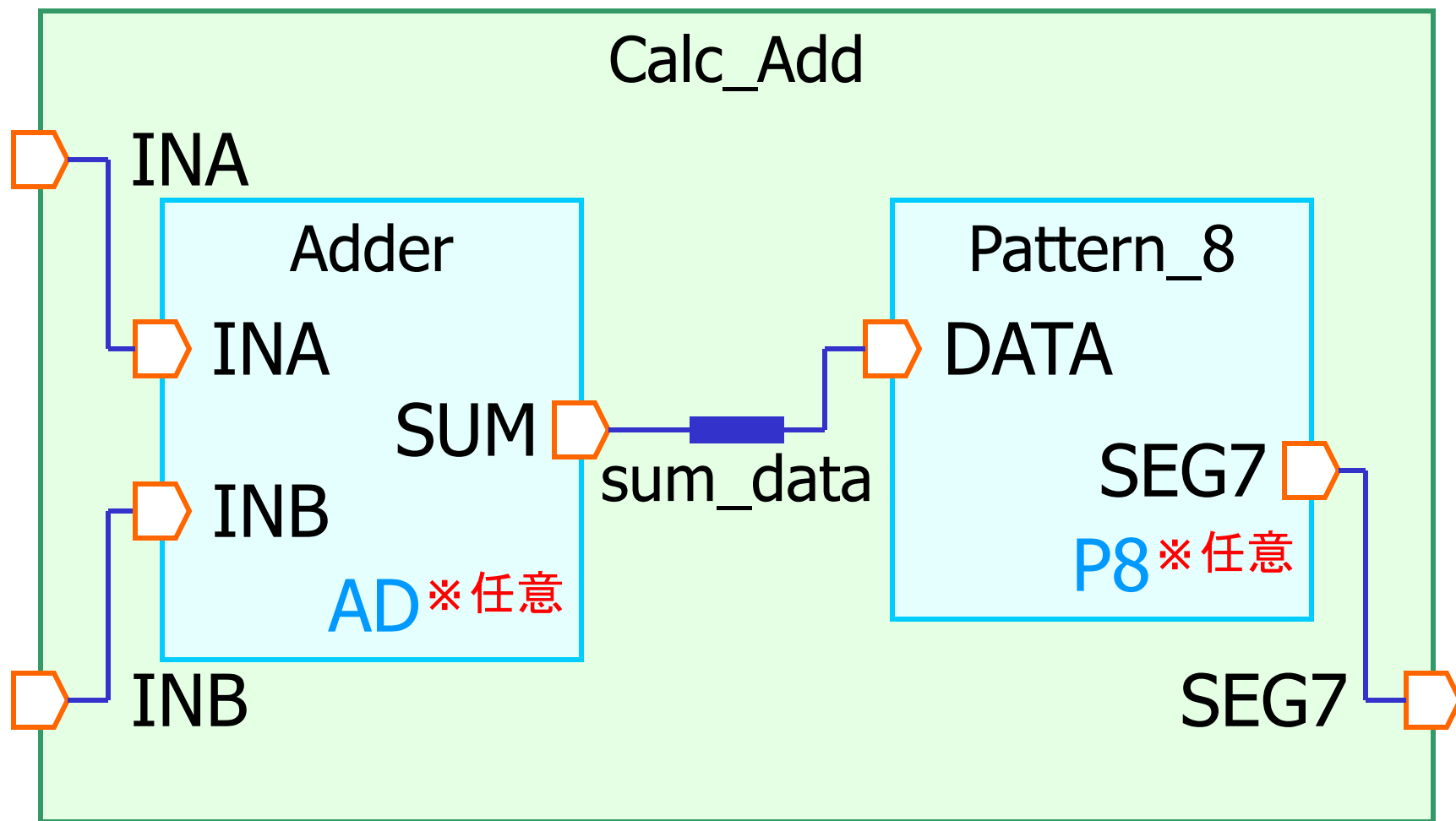
- 下位層モジュール

Adder: 2信号の加算を行う回路(完成済)

Pattern_8: 練習3の回路をコピー&編集
(扱う信号の型を変更しなければならない)

練習4

Calc_Add(2)



練習4

余談(テスト・ベンチ)

