本科目の設計製作フローと、VHDLデザイン

↓：手掛けるフロー

ALTERA社製PLD用
EDA/CADツール

Quartus II 9.1
Web Edition

Windows

接続ケーブル

ALTERA社製PLD
●EPM7128SLC84-15
●EP2C70F896

EDA/CADツール

ハードウェア仕様記述（UML、C言語）
ビヘイビア（動作）記述（HDL、C言語）
動作合成
RTL（機能）記述（ブロック図、HDL、C言語）
論理回路・式記述（論理回路図、HDL）
設計
記述チェック
機能検証 sim
論理合成
ゲートレベル記述（ネットリスト
配置・配線
レイアウトレベル
（コンフィグレーシ
タイミング検証 sim
コンフィグレーション
動作確認（ボード）

VHDL（プログラミング）演習
『ディジタル回路を設計する』

1. デザインの段階（抽象レベル）
   ・メイン：RTL
   ・サブ：ゲートレベル②
2. デザインの方法
   HDL → VHDL
3. モジュール化デザイン
   容易

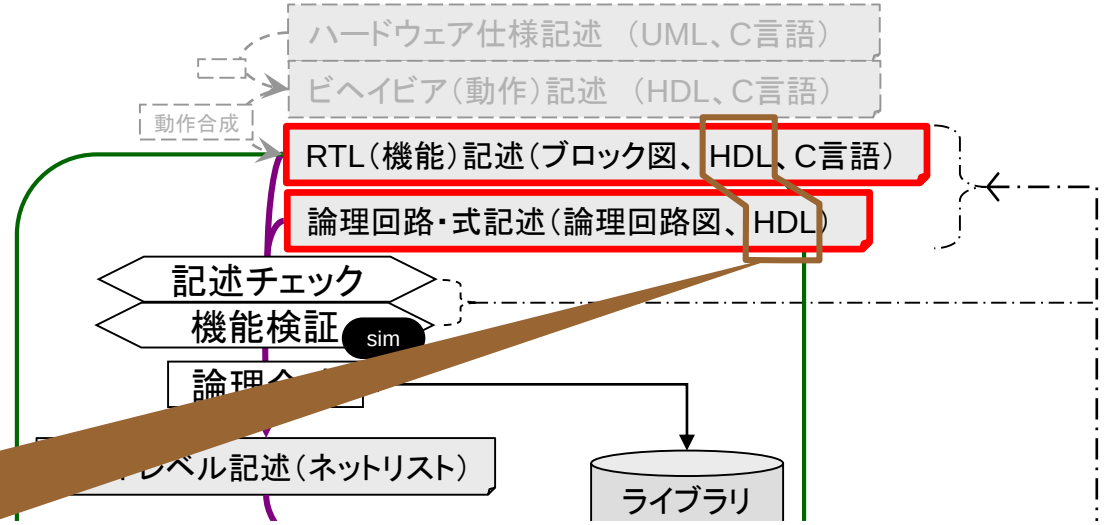
PLD

PLD用開発ボード
●ISPプログラマ＆チェッカ
●DE2-70

2大HDL

本科目では
VHDLを取り上げる。

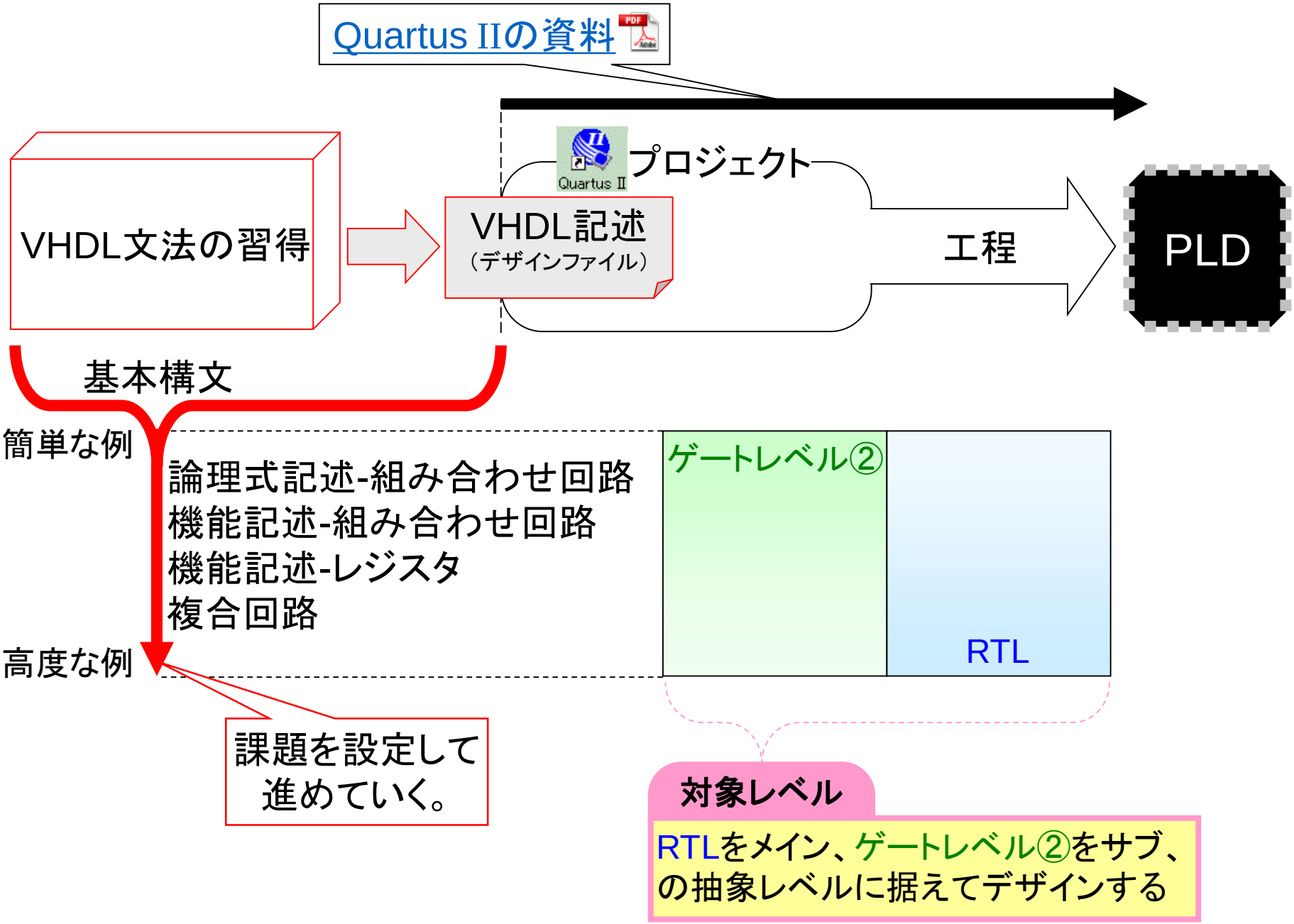
（本質的には、
VHDLとVerilog HDLとに大差はない。）



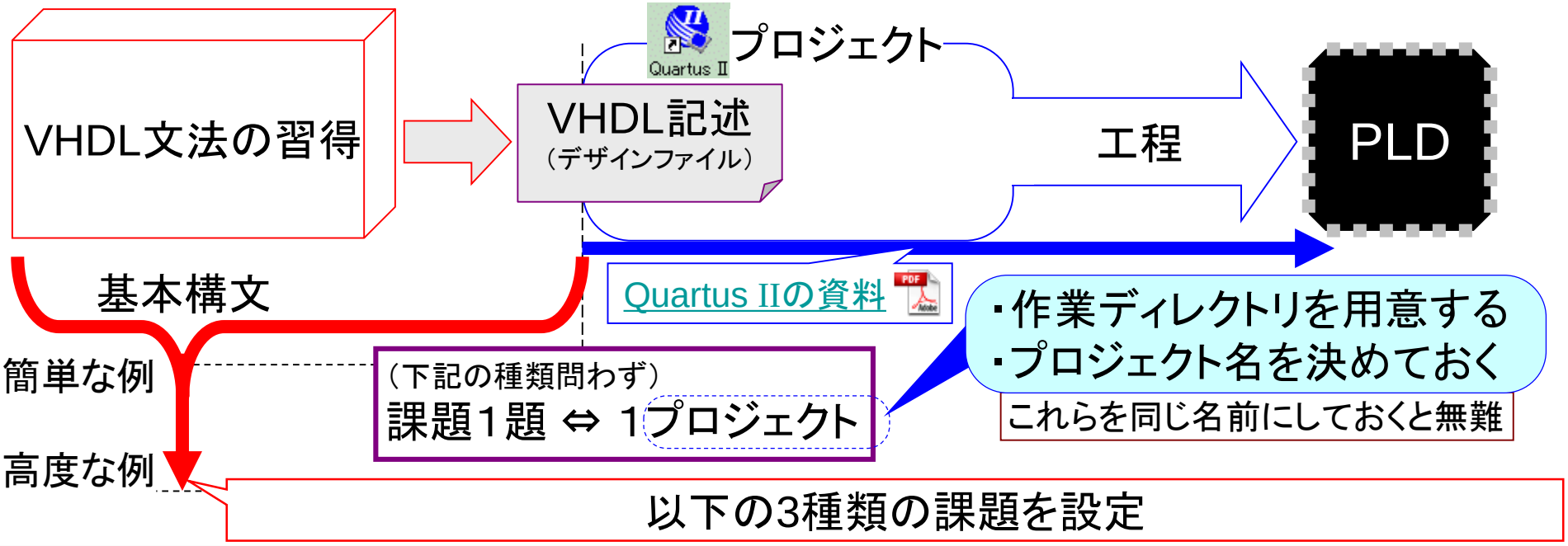
VHDL (Very high speed integrated circuit Hardware Description Language)		Verilog HDL
米国国防総省 VHSICプロジェクトにて開発 （1987年）	誕生	米国Gateway Design Automation社（現Cadence Design Systems社）の回路シミュレータVerilog-XLの専用言語として開発（1984年）
VHDL’87(IEEE-1076-1987) VHDL’93(IEEE-1076-1993)	標準化	IEEE-1364-1995

【注】VHSIC：Very High Speed Integrated Circuit

VHDL文法の習得　{ゲートレベル②→RTL}×{Quartus IIの工程}



課題の設定、課題とプロジェクト



	●練習課題●	■必須課題■	▲追加課題▲
作業タイミング	一斉に	適宜	適宜
レポートとの関係	無し	この中から、レポート課題を出題する。（後日、別スライドで出題する。）	レポート課題に加えて、追加課題をレポートに含めた場合、＋αの評点を付与する。
課題の設定目的	設計製作フロー全体をひと通り押さえ、以降の課題を円滑に行う準備をする。	HDLを用いたディジタル回路設計における基本的内容。	時間的余裕や学習意欲に応える。

VHDLデザインファイルの基本構文

エンティティ名.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity エンティティ名 is
    ジェネリック宣言の記述
    入出力ポートの記述
end エンティティ名;

architecture RTL of エンティティ名 is
各種宣言の記述
begin
    回路機能の記述
end RTL;
```

◆ライブラリ／パッケージ部
特別な事がない限りこの通りに記述する。

◆エンティティ部
デザインの外部インターフェースを記述する。
- ジェネリック宣言の記述（必要ならば）
- 入出力ポートの記述（メイン）

◆アーキテクチャ部
デザインの内部機能を記述する。
- 各種宣言の記述（必要なものを）
- 回路機能の記述（メイン）

●「エンティティ名」の文字列を記述する所
1. ファイル名（拡張子.vhdが付く）
2. エンティティ部の先頭
3. エンティティ部の末尾
4. アーキテクチャ部の先頭

●コメント
「--」（ハイフン2個）で始め、行末まで有効。

【注】エンティティ名のつけ方
- 先頭文字が半角英文字の半角英数文字列。
- 大小文字を区別しない。
- VHDL予約語は使えない。

VHDLデザインファイルの基本構文のイメージ

エンティティ名.vhd

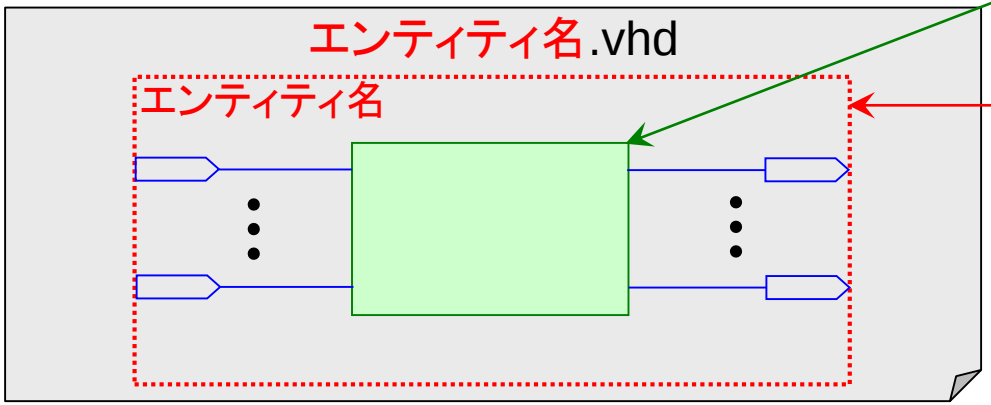
```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity エンティティ名 is
      ジェネリック宣言の記述
      入出力ポートの記述
end エンティティ名;

architecture RTL of エンティティ名 is
各種宣言の記述
begin
      回路機能の記述
end RTL;
```

◆エンティティ部
デザインの外部インターフェースを記述する。
- ジェネリック宣言の記述（必要ならば）
- 入出力ポートの記述（メイン）

◆アーキテクチャ部
デザインの内部機能を記述する。
- 各種宣言の記述（必要なものを）
- 回路機能の記述（メイン）



信号

◆信号 •作図のときに描く配線に相当する概念。
•エンティティ部の入出力ポートの記述で定義される信号を、入出力ポートという。
•アーキテクチャ部の各種宣言の記述で定義される信号を、内部信号という。
•全ての信号は、信号名とデータ型で定義づけられ、9つの論理値をとる。

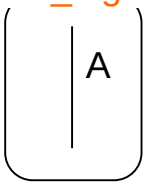
- 先頭文字が半角英文字の半角英数文字列。
- 大小文字を区別しない。
- VHDL予約語は使えない。

【注】
入出力ポートと内部信号の定義の書式は異なる。

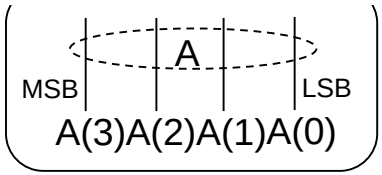
0, 1, X	Logic0, Logic1, Unknown
Z	High Impedance
–	Don't Care
U	Uninitialized
L, H, W	Weak Signals of 0, 1, X

●主な2つのデータ型

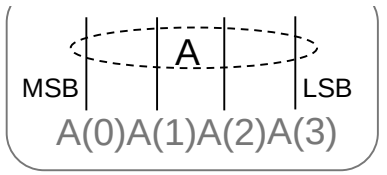
①信号名Aの型:std_logic
（1本の配線）



②信号名Aの型:std_logic_vector(3 downto 0)
（複数本の配線、①の配列型）



std_logic_vector(0 to 3)



●定数表現

①std_logic
'でくくる。
例:
'0','1','Z'

②std_logic_vector
"でくくる。
例:
"1101","00101111"

●基数による定数表現
以下は全て同じ値

"00101111"	Binary
B"00101111"	
O"57"	Octal
X"2F"	heXadecimal

●切り出し(列数指定)と連接(&演算子)
信号名Aの型を
std_logic_vector(3 downto 0)
とするとき、以下は全て同じ意味。

A
A(3)&A(2)&A(1)&A(0)
A(3 downto 2)&A(1 downto 0)
A(3)&A(2 downto 0)

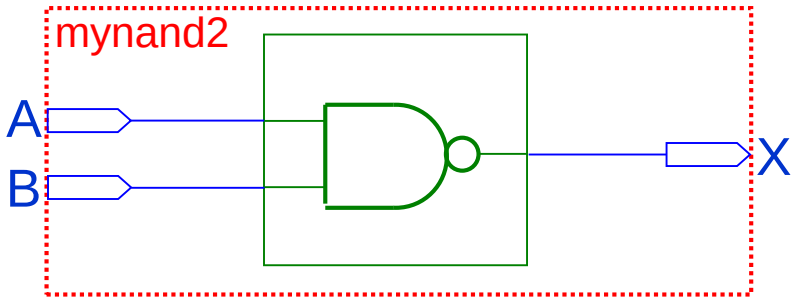
（例）入出力ポート、回路機能

エンティティ名.vhd

```
entity エンティティ名 is
    ジェネリック宣言の記述
    入出力ポートの記述
end エンティティ名;

architecture RTL of エンティティ名 is
各種宣言の記述
begin
    回路機能の記述
end RTL;
```

mynand2.vhd



入出力ポートの記述

```
port(
  A : in std_logic;
  B : in std_logic;
  X : out std_logic
);
```

同義

または

```
port(
  A,B : in std_logic;
  X : out std_logic
);
```

回路機能の記述

```
X <= A nand B;
```

回路ブロック図をイメージしながら記述：エンティティをブロックとしてイメージ

port文、信号代入文、演算子

●ポート文：入出力ポートの記述

書式　`port(ポート1; ･･･ ポートi; ･･･ ポートN);`

信号名,信号名,･･･ ： 方向指定モード データ型

「:」以下が同一である複数の信号名は、コンマで区切り並べて記述できる

in	入力ポートを表わす
out	出力ポートを表わす
inout	双方向ポートを表わす

●信号代入文：回路機能記述の基本

書式　信号名 <= 定数または式;

両辺の型が一致していること

- 信号名（左辺とは別の）
- 演算式

関係演算子を除く

●演算子

【※】信号代入の記号ではない

優先順位別					
not	*	+	=	and
abs	/	-	/=	or
**(冪)	mod	&	<	xor
	rem		>	nand
			<=【※】	nor
			=>	

高 ←― 優先順位 ―→ 低

演算機能別				
論理	関係	連接	算術
not	=	&	+
and	/=		-
or	<		*
xor	>		**(冪)
nand	<=【※】		/
nor	=>		mod
			rem
			abs

項数別	単項	2項
	not	and
	abs	or
	+	xor
	-	+
		など
		その他
書式例		
	not A	B and C

●練習課題●

●プロジェクト「mynand2」
VHDLを用いて記述し、設計製作フローを進めよ。