ハードウェア記述言語による FPGA 設計の基礎技術の修得

第三技術室システム設計班　　松山　幸雄

１．はじめに

ハードウェア記述言語 HDL(Hardware Description Language)の出現によりプログラムによってハードウェア回路（論理回路）を設計できるようになった．また PLD（Programmable Logic Device）の出現により，プログラムにより容易にデジタル IC 回路を専用のオリジナルＩＣに創ることができるようになり,民生品や特殊品などの組み込み機器で使用されている．本研修では，ＦＰＧＡ上でＶＨＤＬ言語による回路作成方法を修得し，応用として 8 ビットＣＰＵソフトマクロ PicoBlaze（KCPSM3）を使用して風速計を作成した．

２．ＦＰＧＡについて

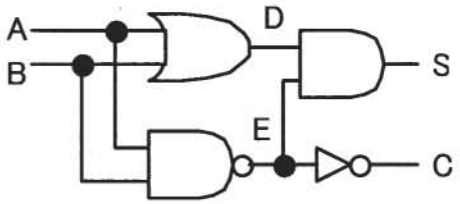
大規模な回路が１つの PLD 内に作成できるものに CPLD と FPGA があり，その位置づけは図１のようになっている．FPGA(Field Programmable Gate Array)は SRAM ベースの LUT(Look Up Table)による基本論理ブロックのアレイ構造セルになっており,例えば LUT に AND ゲートの真理値を書込むことによりセルを AND ゲート化する．高機能，高集積、大規模化するにつれユーザの設計開発が長期化すため，その短縮手段として標準回路機能をコアとしてベンダより提供され，それにはプログラムによるソフトマクロ(USB，小規模 CPU(PicoBlaze(8bit)，MicroBlaze(32bit)など)と FPGA 内臓のハードマクロ(DCM(Digital Clock Module)，乗算器，DSP，大規模 CPU(PowerPC(32bit)など)がある．

・PLD(Programmable Logic Device)
　・PLD ・・・ AND-OR アレイ構造
　　・Simple PLD ・・・ ヒューズ(PAL,GAL・・)
　　・Complex PLD(CPLD) ・・・ EEPROM セル ・・・ altela 社
　・FPGA ・・ SRAM ベースの LUT による基本論理ブロックのアレイ構造
　　SRAM セル,フラッシュメモリ, アンチヒューズ ・・・ xilinx 社
　　電源切断時プログラム消去 → コンフィグ ROM に書込み

［図１］ CPLD と FPGA の位置づけ

３．ＶＨＤＬ言語について

ＨＤＬでよく使われている言語には，記述能力は高くないが記述が簡単な Verilog HDL と記述分野が広く高い記述能力を持つＶＨＤＬ(Very High Speed IC DHL)があり，またＶＨＤＬにはゲート記述を表すゲートレベル，回路生成を表すレジスタトランスファレベル(ＲＴＬ)，状態遷移を表すビヘイビアレベル，システム全体を表すアーキテクチャレベルの４つの記述レベルがある．研修ではＶＨＤＬでロジック回路設計のためにＲＴＬ記述で行った．



【図２】 半加算器回路

３－１．VHDL プログラム記述

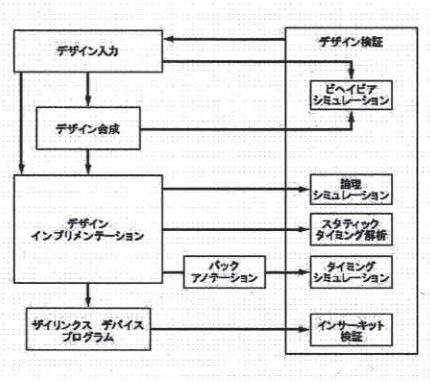
VHDL 言語でのプログラム記述方法を半加算器回路(図 2)を例として図 3 に示す．記述は基本的なデザインデータの集まりでユーザや IEEE 標準のパッケージや算術演算パッケージ等を呼出すためのライブラリ宣言，データの方向，データ型やバス幅等外部との接続を表すエンティティ宣言，内部処理，内部接続や階層間接続など内部動作を表すアーキテクチャ宣言より構成される．エンティティ宣言の port 文で，入力信号 A,B と出力信号(和信号 S,桁上げ信号 C)の方向，型を指定する．アーキテクチャ宣言では，内部信号 D,E のデータ型，幅等を指定し，begin 文では演算を行いその結果を出力 D,E,S,C に代入する signal 文は同時に処理される．またこの半加算器回路をコンポーネント化し全加算器回路の階層下として使用できる．

```
-- ライブラリ宣言
library IEEE;
use IEEE.std_logic_1164.all;
--エンティティ宣言　・・・　外部とのインターフェース部
entity HalfAdder is
   port (
        A,B : in std_logic;
        S,C : out std_logic);
   end HalfAdder;
--アーキテクチャ宣言　・・・ 内部の動作
architecture HalfAdder_body of HalfAdder is
signal D,E : std_logic;
begin
        D <=   A or B;
        E <=   A nand B;
        S <=   D and E;
        C <=   not E;
end HalfAdder_body;
```

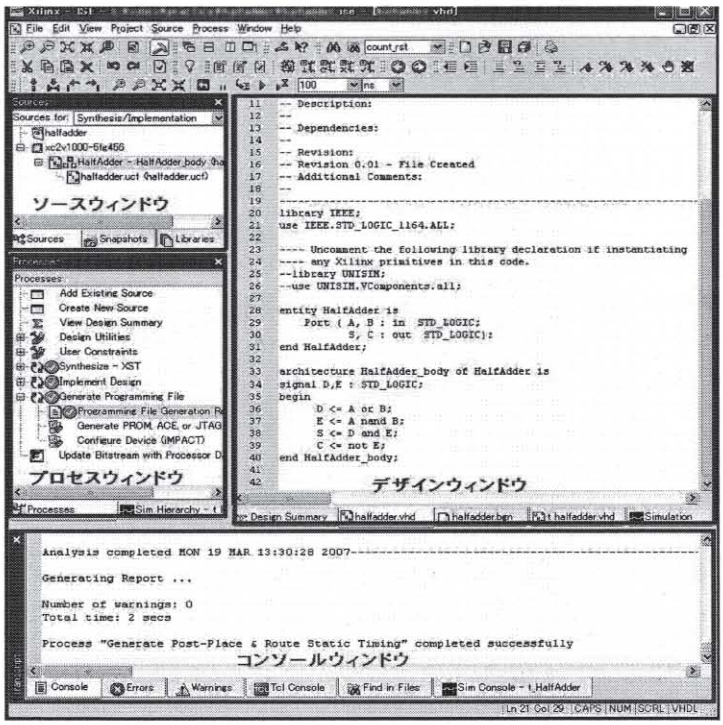
[図3] VHDL プログラム

４．開発ツール

ＦＰＧＡの設計の手順は，図 4 に示した様に回路入力，論理合成やデザイン変換，マップ，配置配線を行うインプリメンテーション，各種シミュレーションで検証しながら完成させる．これらを行う開発ツールは，Xilinx 社の統合開発環境 ISE(Integrated Software Environment)を使用した．これには DHL(回路)入力，論理合成,シミュレーション，デバイスへの回路データ書込等の機能がある．図 5 は ISE 画面でデザインプログラム構成を示すソースウィンドウ，開発手順を表すプロセスウィンドウ，各種レポートやソースプログラム表示をするデザインウィンドウ，各種ログ表示するコンソールウィンドウから構成されている．



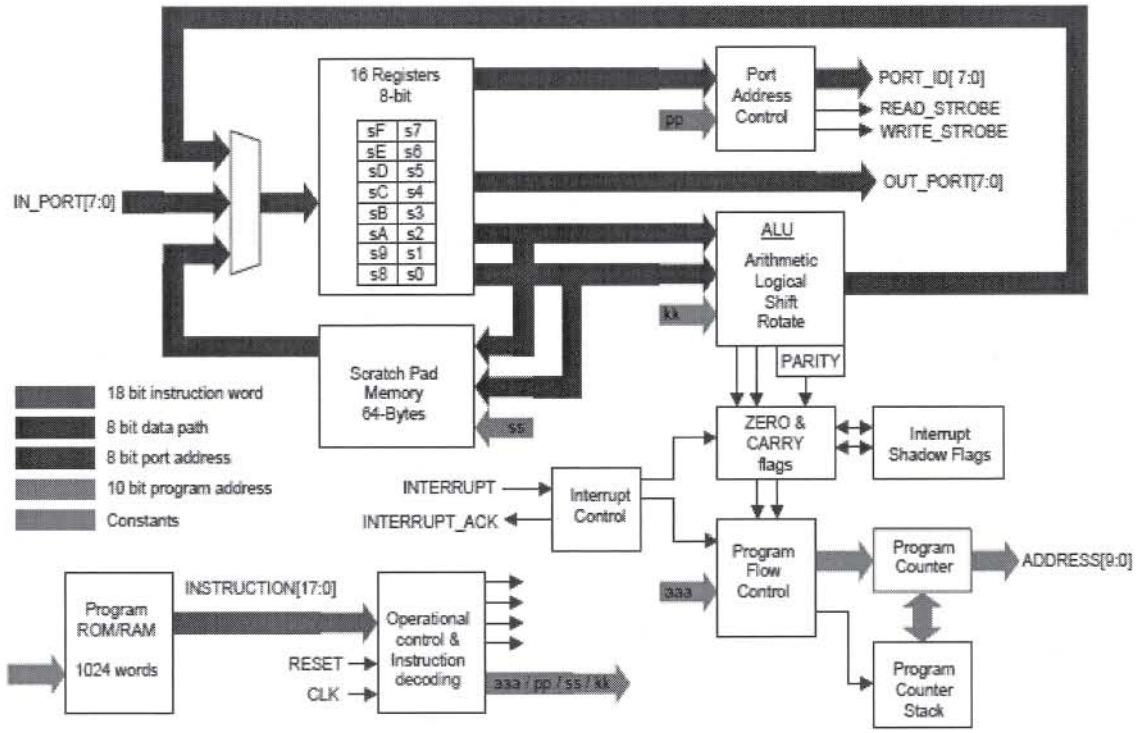
[図 4] 開発手順



[図５] ＩＳＥウィンドウ

５．KCPSM３

KCPSM3((K)constant Coded Programmable State Machine)は，ソフトマクロ PicoBlaze の一つで Xilinx 社の FPGA(Spartan,Vertex-2)に適したシンプルな 8 Bit Micro Controller でる．構成



[図 6] KCPSM3 構成図

```
component kcpsm3
  Port (      address : out std_logic_vector(9 downto 0);
          instruction : in std_logic_vector(17 downto 0);
              port_id : out std_logic_vector(7 downto 0);
         write_strobe : out std_logic;
             out_port : out std_logic_vector(7 downto 0);
          read_strobe : out std_logic;
              in_port : in std_logic_vector(7 downto 0);
            interrupt : in std_logic;
        interrupt_ack : out std_logic;
                reset : in std_logic;
                  clk : in std_logic);
 end component;
```

［図 7］ KCPSM3 のコンポーネント定義(VHDL)

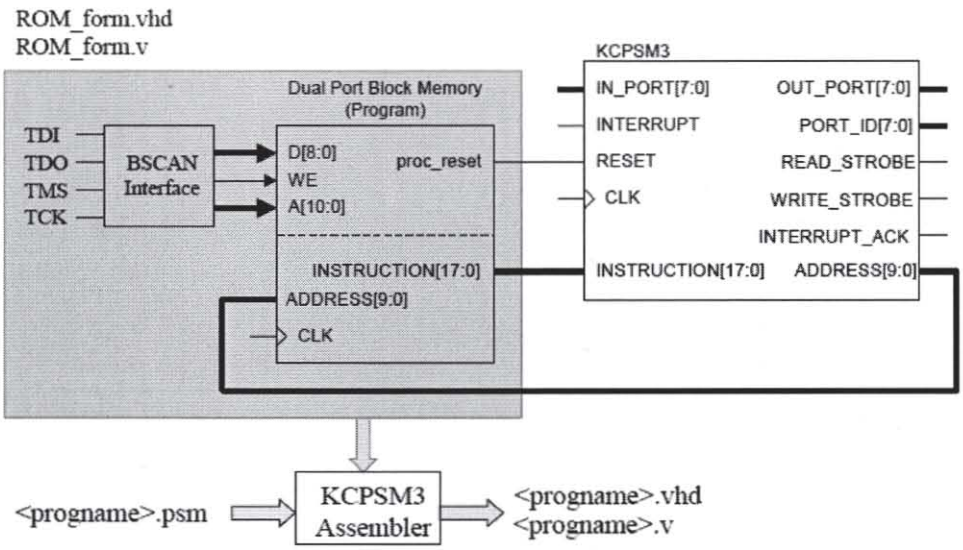
```
component control
  Port (     address : in std_logic_vector(9 downto 0);
         instruction : out std_logic_vector(17 downto 0);
          proc_reset :out std_logic
                 clk : in std_logic);

  end component;
```

[図 8] プログラム ROM のコンポーネント定義

は図 6 のようになっており，命令長：18bit/word，命令数：２１命令，プログラムメモリ：1024wod，レジスタ数：１６(8bit)，入・出力ポート数：各 256，スクラッチ・パッド・メモリ：６４byte(変数領域)，１重割込み等の仕様で，ソフトウェア言語はアセンブラを使用し，DOS でコンパイルする．KCPSM3 マクロは VHDL ソース(kcpsm3.vhd)で供給されており，ユーザ VHDL ソースには KCPSM3 のコンポーネントを定義(図 7)すれば良い．図 8 はプログラム ROM のコンポーネント定義である．

5．1　マイクロプログラムの書込み



[図８]　JTAG プログラムローダー

プログラムメモリへのマイクロプログラム書き込みは１）論理合成時での組込み２）プログラムメモリへの直接書き込みの２とおりがあり，１）の場合，アセンブリコンパイル後作成される VHDL(or VerilogHDL)ファイルを FPGA の論理合成時に組み込まれる．２)の場合，プログラムメモリ ROM はデュアルポート RAM を使用しているので，書込みポートから直接書き込みことができる．図９はコンパイル，直接書き込むためのファイル変換，オンボードの USB インタフェースを使用して JTAG を介して FPGA に書き込むためのバッチファイルで，論理合成する時間が省略される．

```
・kcpsm3 ***.psm          .... compile
・hex2svf ***.hex ***.svf .... serial vector format
・svf2xsvf ***.svf ***.xsvf ... xilinx svf
・setMode -bscan
・setCable –p auto
・addDevice –position 1 –file ***.xsvf   } USB
・play
・quit
```

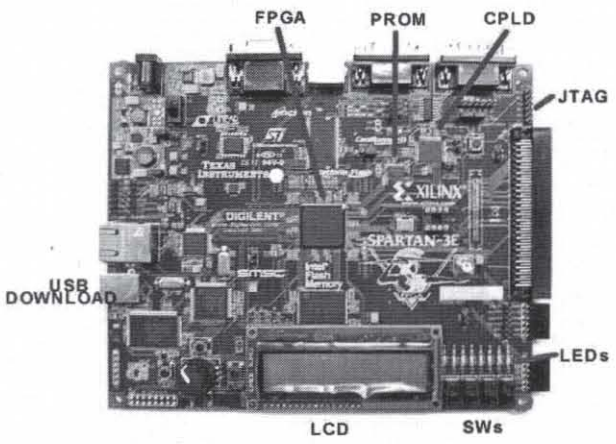
[図９] コンパイル,ファイル変換 USB 書込み

６．風速計の製作

応用として風速計を製作した．これは平成１４年度専門研修で行った風速計でＰＩＣをＦＰＧＡ

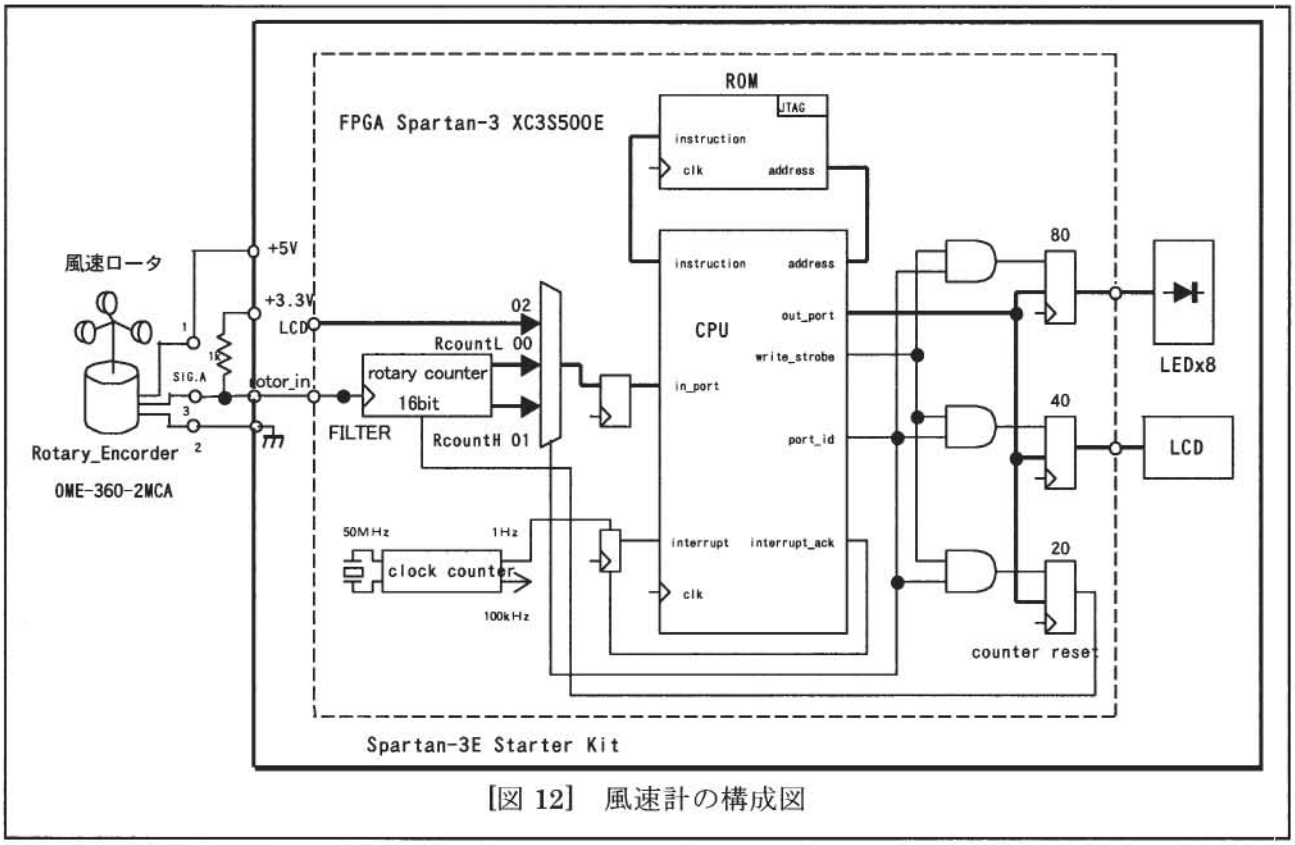
・FPGA：Spartan3E XC3S500E
・CPLD：CoolRunner XC2C64A
・ConfigPROM：XCF04S
・64Mbyte DDR SRAM
・10/100 Ethernet
・PS/2 mouse or keyboard
・VGA display port
・9pin RS-232 ports
・周波数：50MHz
・プログラムダウンロード：USB
・2行x16文字 LCD
・8LEDs
．4スライドスイッチ

[図10] スタータキットボードの主な仕様



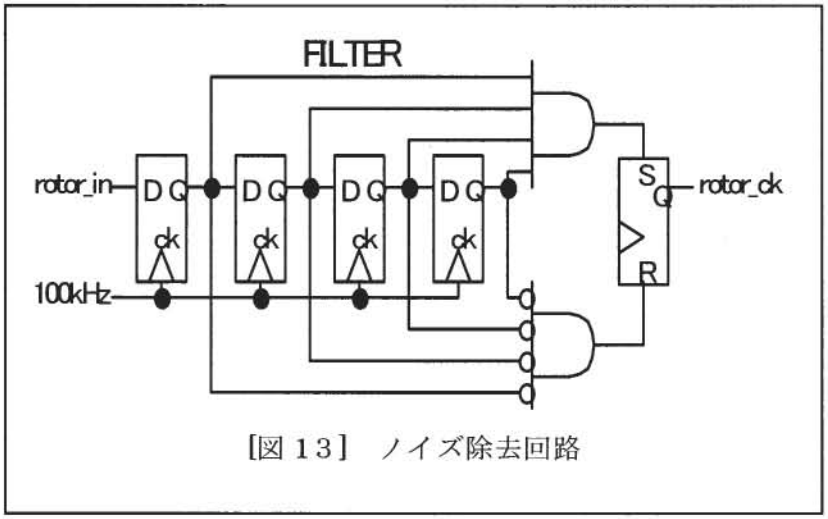
[図 11] Spartan3E スタータキットボード

に置き換えて行った．使用した評価ボードは Xilinx 社の Spartan3E スタータキットボードで FPGA(Spartan 3 E XC3S500E)，CPLD(CoolRunner XC2C64A)を核としたメモリ，VGA ポート，RS232 ポート，LCD など図 10 に示した仕様のものが搭載され演習，実験のみならず，簡易なパソコンを構築することもできるようになっている．なおビットストリームファイルは USB をとおして，FPGA 又はフラッシュメモリへダウンロードする．ボードの概観を図 11 に示す．



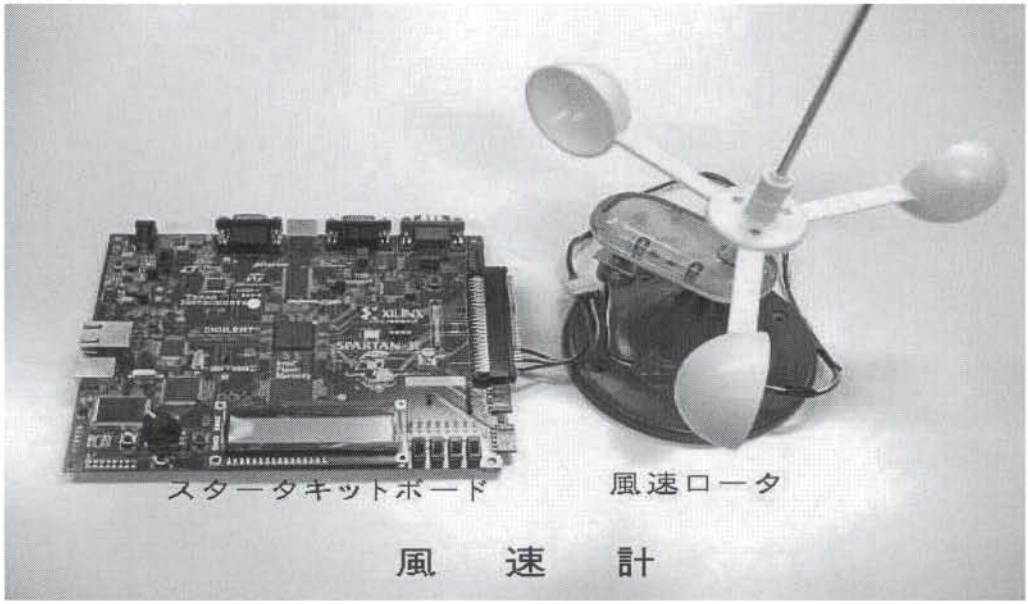
[図 12]　風速計の構成図

風速計の構成図は図１２のようになっており，風速計ロータのパルスをカウンタに入力し，一秒毎に CPU に割り込みをかけ，CPU からカウンタの値を読込み，Anemometer 製品で更正した風速値に換算し，カウンタ値と共に LCD に出力した．また風速計ロータは大和科学教材研究所の理科教材で，それに MEMICON 社のロータリエンコーダ OME-360 を装着し,ロータ出力をパルス出力するように改良した．エンコーダ電源は５V であるが，FPGA 評価ボードの信号入力端子が LVTTL のため，エンコーダのオープンコレクタ出力には３.３V を抵抗でプルアップした．評価ボードでは風速ロータ出力信号にノイズが含まれるとカウンタに誤差が生じるため，カウンタの前（図１２の・FILTER）に図１３のようなノイズを除去するフィルタ回路を設けた．フィルタは 4 ビットのシフトレジスタと AND ゲート，SR レジスタより構成され，シフトレジス



[図 13]　ノイズ除去回路

タに 100kHz のクロックを入力することにより４０μＳまでのノイズを除去した．風速計の概観は下図のようになっており，1 秒間隔で風速値が認識できるように 8 個の LED で左右 4 個を交互(1 秒毎)に点灯するようにした．



7．まとめ

HDL の出現により論理回路がプログラムで記述できライブラリ，マクロ化されるようになってきた．ＦＰＧＡは高機能，大規模化され使用困難に見えたが高機能な開発ツールや IP マクロの充実で使い易くなった．今回の研修では VHDL やＦＰＧＡを十分使い切ることは不可能であったが，それらを使用するための基礎技術は修得できた．最後にこの研修の機会を与えてくださった情報メディア工学科森眞一郎先生に深く感謝します．

[参考文献]

1）長谷川裕恭：改訂「VHDL によるハードウェア設計入門」CQ 出版社
2）益田久喜：「FPGA 活用チュートリアル」 2006/2007 年版　CQ 出版社
3）Spartan-3E スタータキットボードユーザガイド　Xilinx Ltd
4）PicoBlaze KCPSM3 for 8-bit Micro Controller for Spartan-3,Virtex-2 and Virtex-2 PRO
5）PicoBlaze JTAG Loader Quick User Guide　Xilinx Ltd