

# 1 目的

ハードウェア記述言語 VHDL を用いて、ストップウォッチの機能あるいは回路構成を記述し、それを FPGA 上に実現することにより、デジタル回路の動作原理ならびにその設計手順を理解する。また、設計・シミュレーション・実機検証の一連の工程を通じて、FPGA 開発の実践的なスキルを習得し、効率的なデバッグ手法や設計の最適化についても学ぶ。

## 2 原理

### 2.1 FPGA の原理

FPGA (Field Programmable Gate Array) は、ユーザーが回路構成を自由に書き換えられる集積回路である。FPGA 内部には多数のロジックブロック (論理素子) と、それらを相互接続する配線 (インターコネクト)、そして設定情報を保持する構成メモリが組み込まれている。ユーザーは HDL (ハードウェア記述言語) を使って回路設計を行い、そのデータを FPGA に書き込むことで、加算器やカウンタ、プロセッサなど多様なデジタル回路を構成することができる。これにより、専用 IC のようなハードウェアの高速性と、ソフトウェアの柔軟性を両立できる。(1)

FPGA のチップの内では、論理機能を実現することができ、EmbeddedArrayBlock (EAB) および LogicArrayBlock (LAB)、ならびにそれらを接続するための FastTrack 列配線および行配線が、図 2.1 に示すように規則的に配置されている。

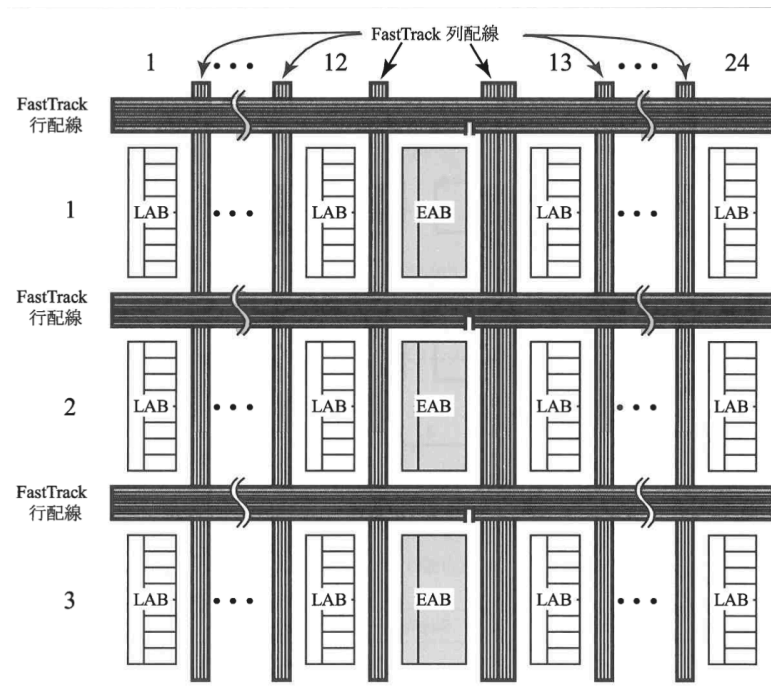


図 2.1: EP1K10 の構造

EP1K10 は 3 行 25 列構造で、各行に EAB が 1 個、LAB が 24 個配置されている。FastTrack 配線は高速伝送が可能で、行配線は 144 本、列配線は通常 24 本だが、EAB 横は 48 本となっている。EAB は 8 入力 16 出力の任意関数を実現可能で、RAM や FIFO などのメモリ機能やデータ処理にも利用できる。例えば、1 つの EAB で 256x16 ビットや 512x8 ビットの RAM を構成可能で、複数の EAB を組み合わせることでより大容量の RAM も実現できる。

## 2.2 実験で作成した stop watch のモジュール構成と動作原理

# 3 実験結果

## 3.1 実験手順

### 3.1.1 VHDL コードの作成

サブモジュールである周期 0.1 秒のパルス発生回路（PulseGen01）を VHDL で記述し、PulseGen01.vhd）として作成する。

### 3.1.2 コンパイルとデバッグ

Quartus Prime を用いて VHDL コードをコンパイルし、FPGA (MAX10 10M50) に実装可能な回路へ変換する。コンパイル時にエラーが発生した場合は、デバッグを行い記述を修正する。

### 3.1.3 シミュレーションによる動作確認

Quartus Prime のシミュレーション機能を利用して、設計した回路が仕様通り動作するか確認する。問題があれば再度デバッグを行う。

### 3.1.4 FPGA への書き込みとボード上での実機動作確認

動作が確認できた後、回路データを MAX10 10M50 ボードへダウンロードし、実際のハードウェア上で動作を確認する。

リスト 1: PulseGen01.vhd

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity PulseGen01 is
5     port (
6         CLKIN   : in std_logic;
7         RSTIN    : in std_logic;
8         PLSOUT   : out std_logic
9     );
10 end PulseGen01;
11
12 architecture Some_Description of PulseGen01 is
13     signal C_CNT : integer range 0 to 4999999;
14 begin
15
```

```

16  process(CLKIN, RSTIN)
17  begin
18      if (CLKIN'event and CLKIN = '1') then
19          if RSTIN = '0' then
20              C_CNT <= 0;
21              PLSOUT <= '0';
22          elsif C_CNT = 4999999 then
23              C_CNT <= 0;
24              PLSOUT <= '1';
25          else
26              C_CNT <= C_CNT + 1;
27              PLSOUT <= '0';
28          end if;
29      end if;
30  end process;
31
32  end Some_Description;

```

## 4 実験結果

### 4.1 使用機材

### 4.2 参考文献

1. FPGA とは, フィールドでプログラムできる論理回路

<https://jp.mathworks.com/discovery/fpga.html>

閲覧日：2025 年 5 月 21 日