

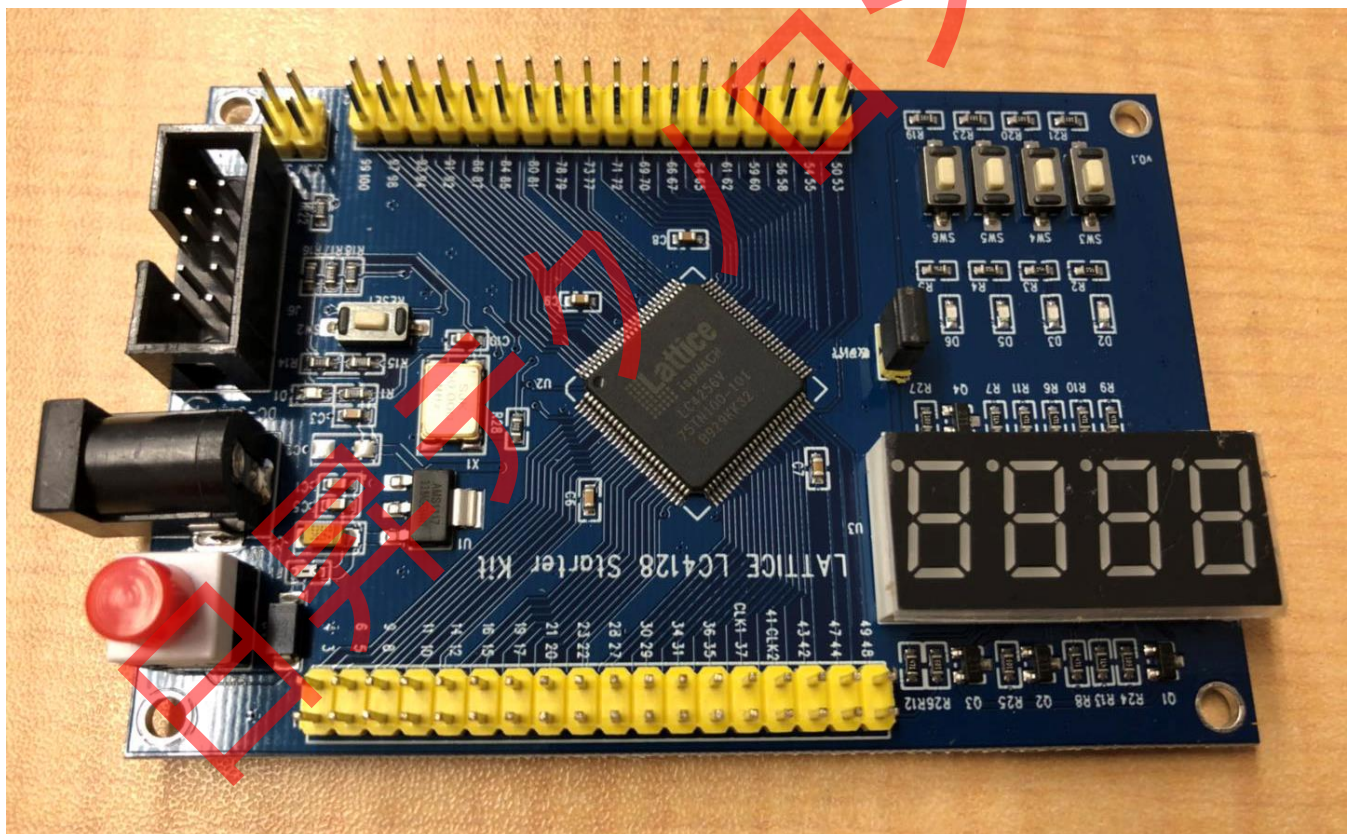
LATTICE CPLD LC4256 開発ボードのマニュアル

株式会社日昇テクノロジー

<https://www.csun.co.jp>

info@csun.co.jp

作成日 2020/10/02



copyright@2020

・修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2020/10/02

※ この文書の情報は、文書を改善するため、事前の通知なく変更されることがあります。最新版は弊社ホームページからご参照ください。「<https://www.csun.co.jp>」

※ (株)日昇テクノロジーの書面による許可のない複製は、いかなる形態においても厳重に禁じられています。

日昇テクノロジー

目次

1	注意事項	4
2	開発ボードの主な部品及びハードウェアリソース	5
2.1	主な部品	5
2.2	ハードウェアリソース	5
2.3	PCB とサイズ	6
3	開発ボードの実装と使用	7
3.1	開発ボードの実装	7
3.2	開発ボードの使用	7
3.2.1	ロジック開発プロセスの紹介	7
3.2.2	ダウンロード	11
4	開発ボード回路説明	15
4.1	電源回路	15
4.2	クロック回路	15
4.3	リセットキー回路	16
4.4	ユーザーキー回路	16
4.5	LED 回路	16
4.6	7SEG LED 回路	16
4.7	JTAG インタフェース回路	17
4.8	外部拡張 IO	17

1 注意事項

使用要求

- ケーブルを抜き差しする前に、開発ボードを使用中止し、電源を切る。
- 不明なもの又は液体が開発ボードと接触した場合に、すぐに開発ボードを使用中止し、電源を切って開発ボードに接続するケーブルを全部抜き出す。
- 異常状況が出る場合に、例えば、設備から煙が出る、異臭を放つ時に、すぐに開発ボードを使用中止し、電源を切る。
- 長時間に使用しない場合に、電源を切る。

静電気安全及び保護

静電気はしばしば開発ボードのチップを損傷する発生誘因となっている。そのため、開発ボードに触る前に十分な保護対策を行わなければならない。

ケーブルの挿抜安全

- ケーブルを挿抜する場合に、フラットケーブルの方向を間違い無い様に注意してください。間違った場合、ロジックチップ又はダウンロードケーブルを損傷する恐れがある。
- 電源を入れたまま JTAG ケーブルを挿抜操作禁止。通電状態操作すればロジックチップに内蔵された設定回路に致命的な損傷をもたなす。(コンピュータマザーボードのボードを挿抜と同じ)

外部電源の使用

開発ボードが DC 5V 電源入力をサポートする。DC 電源インターフェースを使う場合に、電圧が間違い又は極性が逆接するで開発ボードを損傷しないように、電源アダプタ出力が 5V、かつ内側が正極、外側が負極である事を確認する。

2 開発ボードの主な部品及びハードウェアリソース

2.1 主な部品

開発ボードのコアチップはLATTICE ispMACH 4000 シリーズの高速超低電力消費 CPLD LC4256V を採用する。ispMACH 4000 シリーズは 3.3V、2.5V、1.8V 三つ種類の動作電圧があり、それぞれが ispMACH 4000V、ispMACH 4000B、ispMACH 4000C シリーズである。本ボードはこの三つシリーズが使えて、出荷時に LC4256V を搭載している。チップ資料は以下の通り：

PARAMETERS	4032	4064	4128	4256	4384	4512
マクロセルセル数	32	64	128	256	384	512
tPD (ns)	2.5	2.5	2.7	3.0	3.5	3.5
Fmax (MHz)	400	400	333	322	322	322
パッケージ (I/O + 専用入力)						
44 TQFP	30 + 2	30 + 2				
48 TQFP	32 + 4	32 + 4				
100 TQFP		64 + 10	64 + 10	64 + 10		
128 TQFP			92 + 4			
176 TQFP				128 + 4	128 + 4	128 + 4
256 FBGA				128 + 4 / 160 + 4	192 + 4	208 + 4

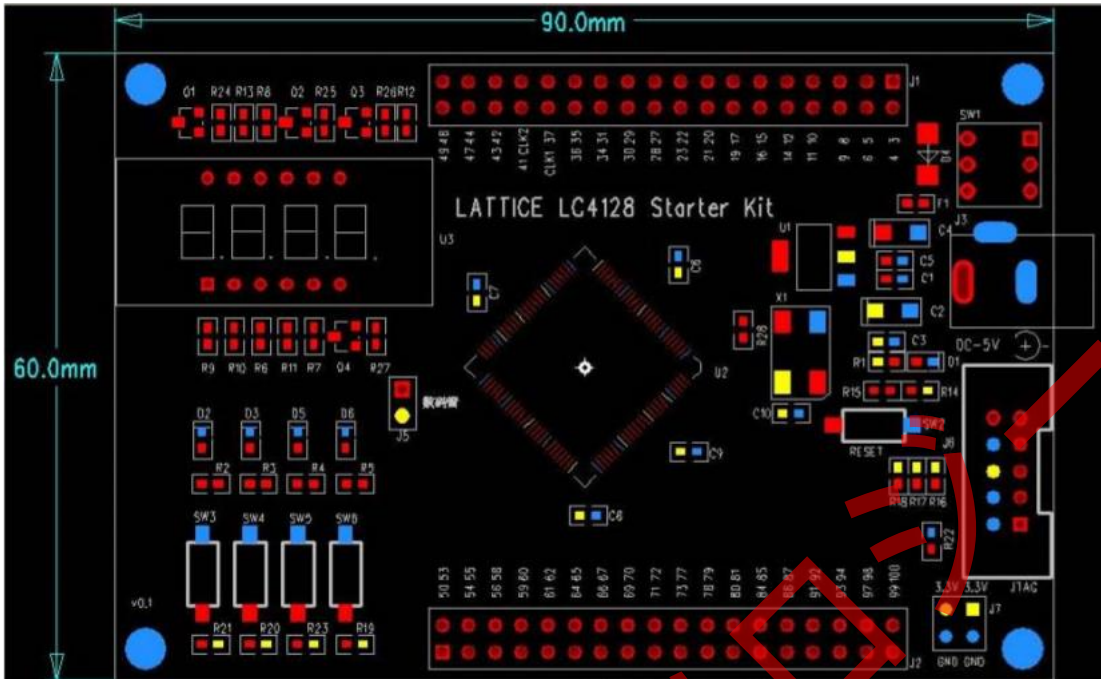
一般的には、マクロセル数と I/O 数は CPLD にとって重要な参考指標である。本ボード低密度のアプリケーション、例えばインタフェースブリッジ、I/O 拡張、デバイス設定などに適用する。

2.2 ハードウェアリソース

- メインチップは LATTICE 社の超高速低電力消費 CPLD、ispMACH4000 シリーズの LC4256V-75T100 を採用。
- 50MHz 水晶発振器により、システムが動作するメインクロックを提供する。
- AMS1117-3.3 低圧差動チップを採用し、最大 700mA の 3.3V 電圧出力までサポートする。
- IN5819 高速ショットキーダイオード、電源逆接の保護設計を持っている。
- Self-locking 電源スイッチ一つ。
- 5V 直流電源コネクタ、電源アダプタで給電できる。
- 赤 LED 一つ、電源表示灯とする。
- ブルー LED 四つ、LED 点滅試験に利用できる。
- 表示用 4 桁 7SEG LED 一つ。
- ユーザーキー四つ。
- リセットキー一つ、ユーザーキーとして利用可能。
- JTAG インタフェースをサポートする。
- 外部へ給電可能。
- デカップリング設計。数多くのデカップリングコンデンサーを搭載する。
- 拡張インタフェースソケット三つ、標準 2.54mm ピッチ。

写真は下図の通り：

2.3 PCB とサイズ



3 開発ボードの実装と使用

本開発ボードは以下の部品が含んでいる。

- 開発ボード 1個
- Mini USB線 1本
- スタッドとネジ 4本ずつ

3.1 開発ボードの実装

- スタッドを開発ボードの穴に取り付ける。
- ダウンロードケーブルを開発ボードに接続する。
- 電源アダプタを開発ボードに接続し、ボードに電源を提供する。
- 実装された開発ボードをデスクの上に平らに置く。
- パソコンに関連ソフトウェアをインストールする。例えば ispLEVER、テキスト編集ソフトウェア UltraEdit、コーディングチェックソフトウェア Nlint など。

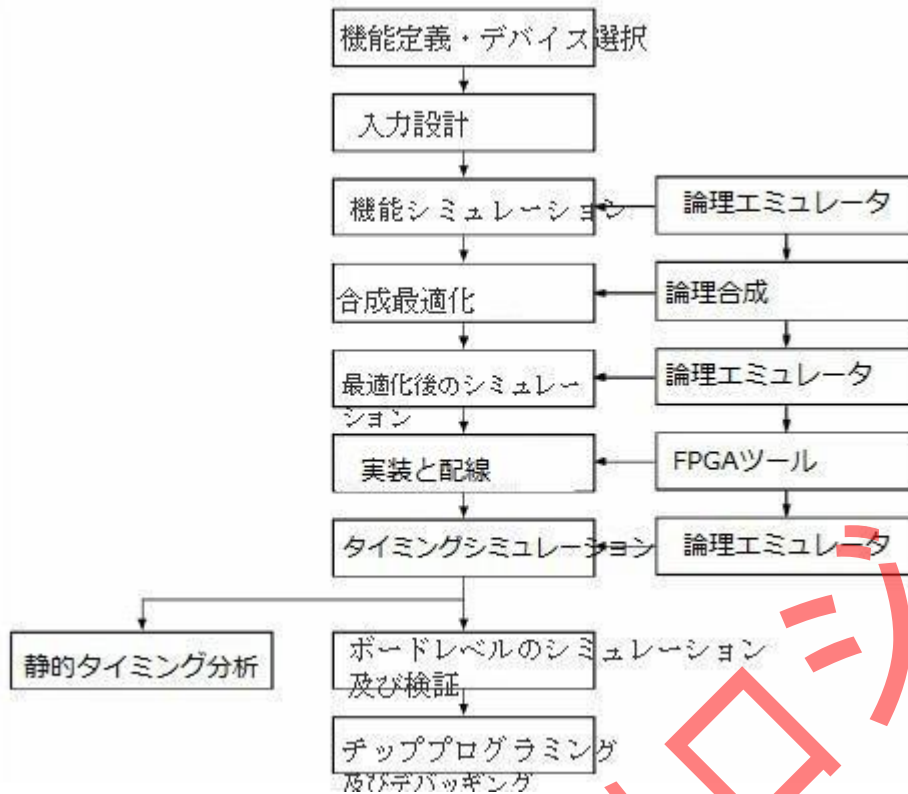
3.2 開発ボードの使用

開発ボードが正確に接続したあと、ボードに電源を入れる。

3.2.1 ロジック開発プロセスの紹介

ロジック開発は、ハードウェア設計とソフトウェア設計二つの部分を含んでいる。ハードウェアはロジックチップ回路、JTAG、クロック、リセット、メモリ、入出力インタフェース回路及び他のインタフェースが含んでいる。これは本ボードで提供している。ソフトウェアはHDLプロセスである。この部分はお客様の事情によって自分で設計する。ここでご参考までいくつかの例を提供する。

ロジック設計プロセスは、EDA 開発ソフトと編集ツールを利用して、ロジックチップに対する開発する。代表的なロジック開発プロセスは機能定義・デバイス選択、入力設計、機能シミュレーション、合成最適化、最適化後のシミュレーション、実現及び配線後のシミュレーション、ボードレベルのシミュレーション、チッププログラミング及びデバッグなどのステップを含んでいる。下図の通り：



機能定義：実現したい機能を定義する。例えば、開発ボード上の二つのLEDの点滅。

入力設計：ハードウェア記述言語（HDL）で実現したい機能を記述する（つまり、テキスト編集ソフトウェア UltraEdit に HDL 言語で記述）。回路図モードを推奨していない（非効率的な方法、維持するのが困難、モジュール構造と再利用も難しい。かつ、回路図のポータビリティが悪い）。例えば、下図は一つの LED 点滅の設計記述、つまり LED が 50000000 回の clk ずつ一回点滅する。


```

parameter TIME_COUNT = 32'd100000000;

/***** ポート声明 *****/
//input
input      clk;
input      reset_n;

//output
output [1:0] led_flash; //ledが1s 間隔で点滅、1-点灯、0-消灯

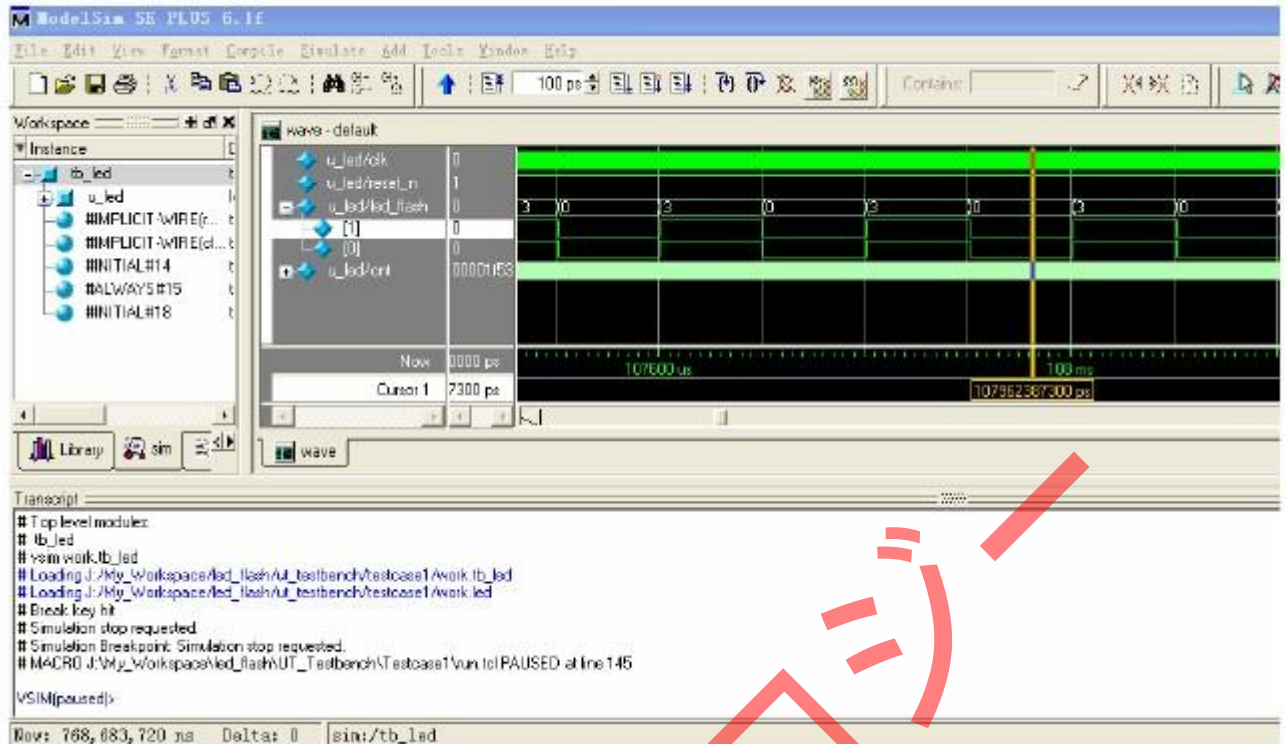
/***** ポート定義 *****/
reg [1:0] led_flash;

/***** 内部信号定義 *****/
reg [31:0] cnt;

/***** コード開始 *****/
always @(posedge clk or negedge reset_n)
begin
  if (reset_n == 1'b0)
    cnt <= 32'b0;
  else if (cnt >= TIME_COUNT - 1'b1)
    cnt <= #U_DLY 32'b0;
  else
    cnt <= #U_DLY cnt + 1'b1;
end

always @(posedge clk or negedge reset_n)
begin
  if (reset_n == 1'b0)
    led_flash <= 2'b00;
  else if (cnt == TIME_COUNT - 1'b1)
    led_flash <= #U_DLY 2'b11;
  else if (cnt == (TIME_COUNT >> 1) - 1'b1)
    led_flash <= #U_DLY 2'b00;
  else
    ;
end
  
```

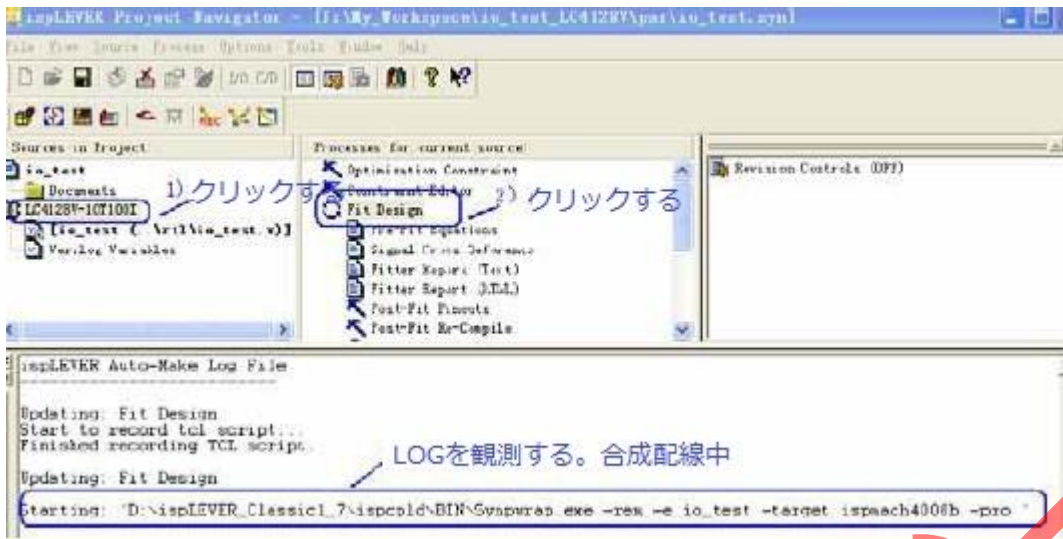
機能シミュレーション：前シミュレーションとも呼ばれる。コンパイルの前にユーザーにより設計された回路のロジック機能を検証する。一般的には Modesim 又は Questa でシミュレーションする。下図が LED 点滅のシミュレーション波形である。led_flash の値が一定な時間を間隔し 1 になり、LED を点灯する。



合成最適化：設計入力を、実際のゲート回路ではなく、AND ゲート、OR ゲート、NAND ゲート、RAM、トリガーなどの基本ロジックユニットから組み立てられた論理接続ネットリストにコンパイルされる。実際のゲート回路はメーターのレイアウトソフトウェアを利用して、合成に生成された標準ゲートレベル構造に応じて生成する。LATTICE 会社の部品は、設計を ispLEVER に導入し、ispLEVER ソフトウェアで合成最適化する (ispLEVER は Synplify 合成ツールを統合した)。

合成後のシミュレーション：合成後の結果と当初の設計が同じかどうかを確認する。シミュレーションの時、合成に生成された標準遅延ファイルを合成のシミュレーションモデルにマークし、ゲート遅延の影響を推定できる。

配線レイアウトと実装：ロジックメーカーのソフトウェアを利用して、ロジックを目標デバイス構造のソースにマッピングし、ロジックのベストレイアウトを決定する。ロジックと入出力機能の接続配線チャンネルを選択する上に、接続し、対応的なファイル (例えば設定ファイル、関連報告) を生成し、合成に生成されたロジックネットリストを具体的なロジックチップに配置する。LATTICE 会社の部品は、設計を ispLEVER に導入し、ispLEVER ソフトウェアで配線レイアウトを行って、設定ファイルを生成する。下図は配線レイアウトの例である。



タイミングシミュレーション：後シミュレーションとも呼ばれる。配線レイアウトの遅延情報を設計ネットリストにマークし、タイミングの違反（即、タイミング約束条件又はデバイス固有のタイミング規則に満足しない、例えば、セットアップ時間、ホールド時間など）を検出する。タイミングシミュレーションの遅延情報は最も正確、最も完全である。

ボードレベルのシミュレーション：主に高速回路設計に使用される。高速システムの信号完全性、電磁妨害などを分析し、一般的には第三者によりシミュレーションと検証が行われる。一般的にはしない。


チッププロフラミングとデバッグ：チッププロフラミングは配線レイアウトと実現後に生成された bit ファイルをロジックチップ又は設定チップにダウンロードし、定義された機能を実現させる。デバッグは FPGA チップのオンラインエンベデッドロジックアナライザで（例えば、XILINX ISE における chipScope, ALTERA Quartus における SignalTap）波形を採集し、かつ分析することである。

3.2.2 ダウンロード

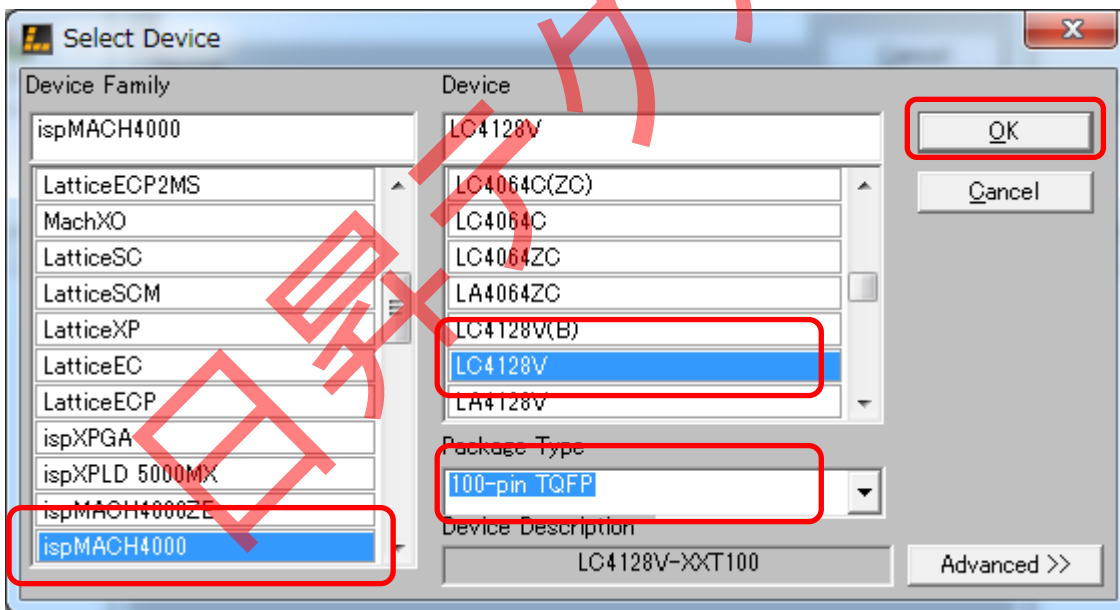
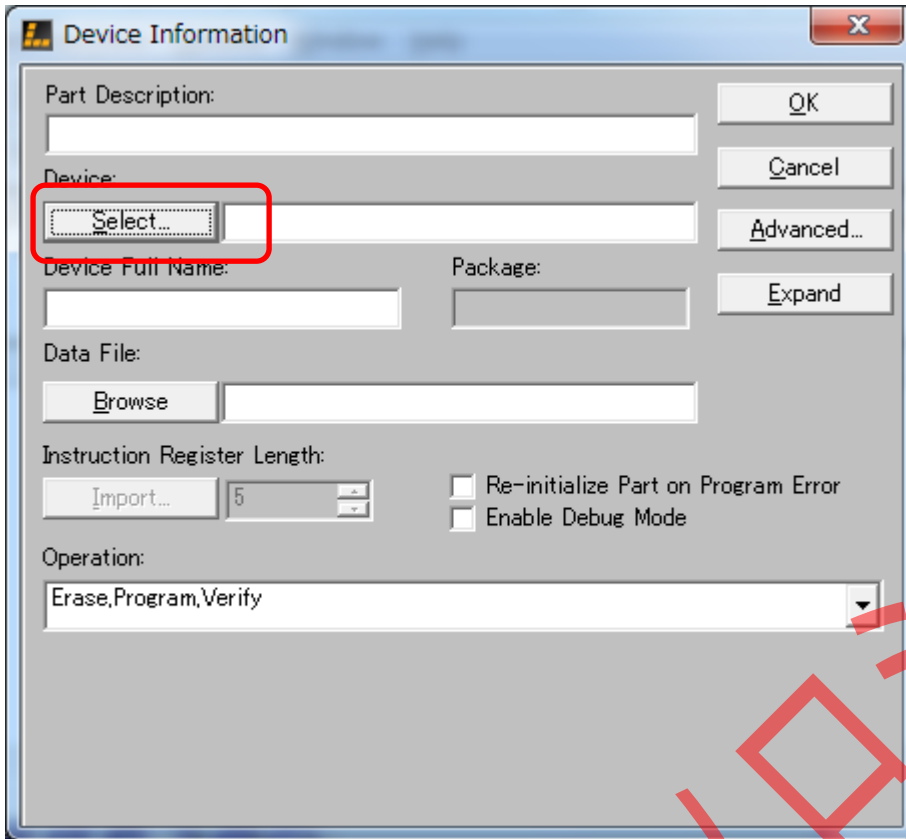
本開発ボードは JTAG モードをサポートする。次に、これについて紹介する。

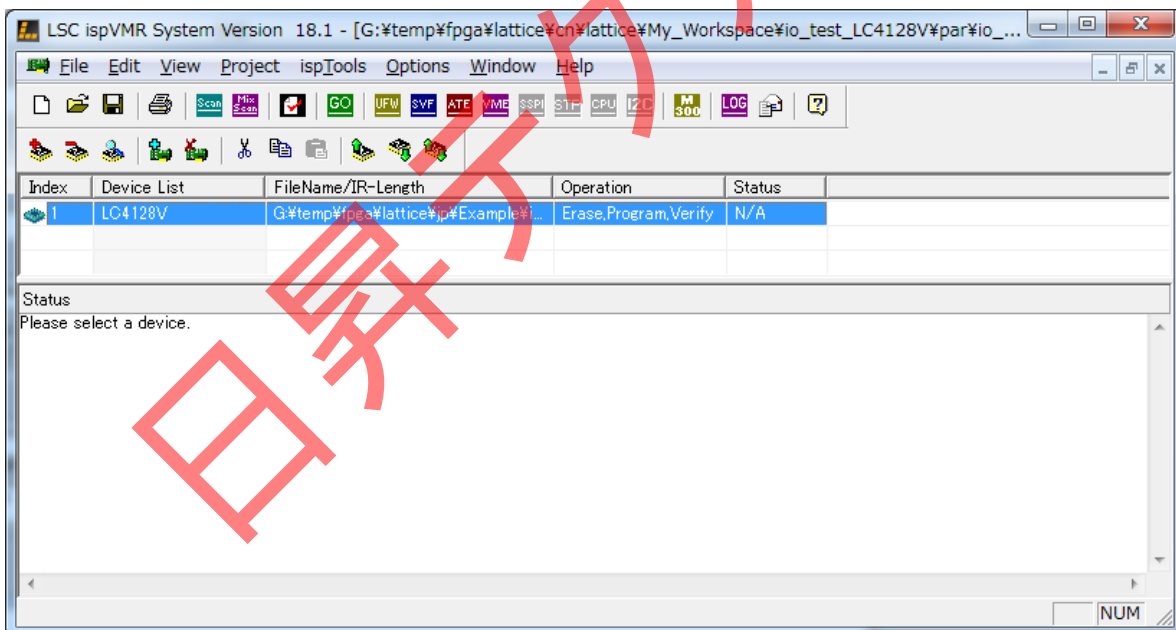
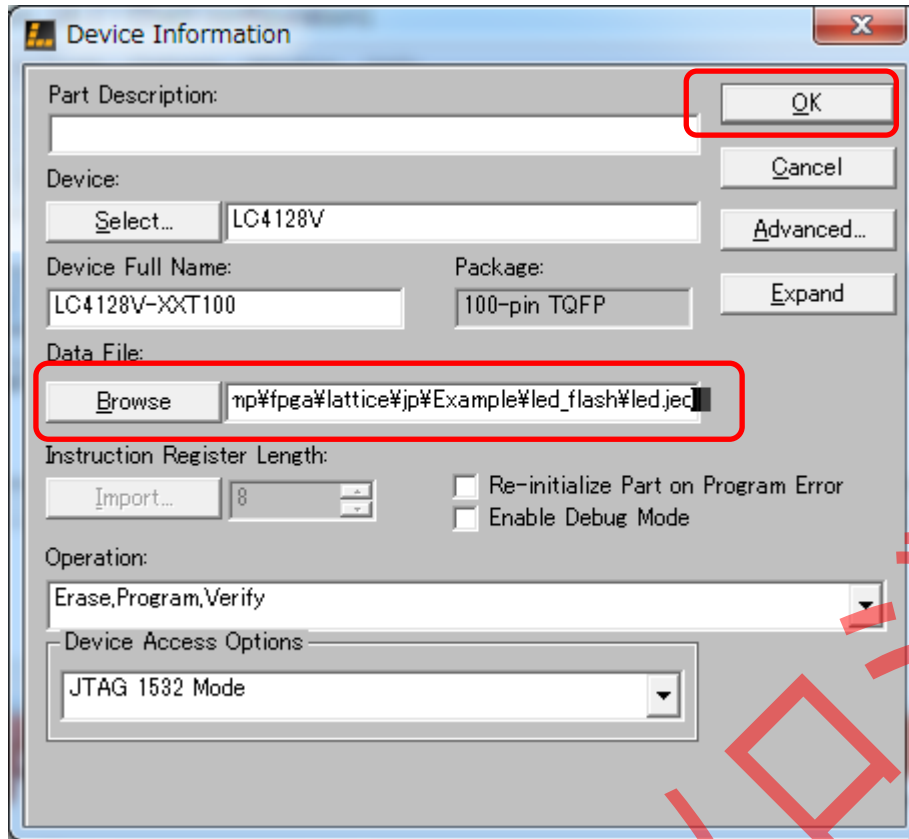
ダウンロードファイルは JED ファイル。JTAG インタフェースを通じて、ロジックを CPLD の内蔵 FLASH にダウンロードする。電源切れの場合に、内容は失わない。

- 1) ispLEVER ソフトウェアで設定ファイルを生成する（サフィックス JED）。
- 2) ダウンロードケーブルを開発ボードの JTAG コネクタに接続する。
- 3) 開発ボードに電源を入れる。

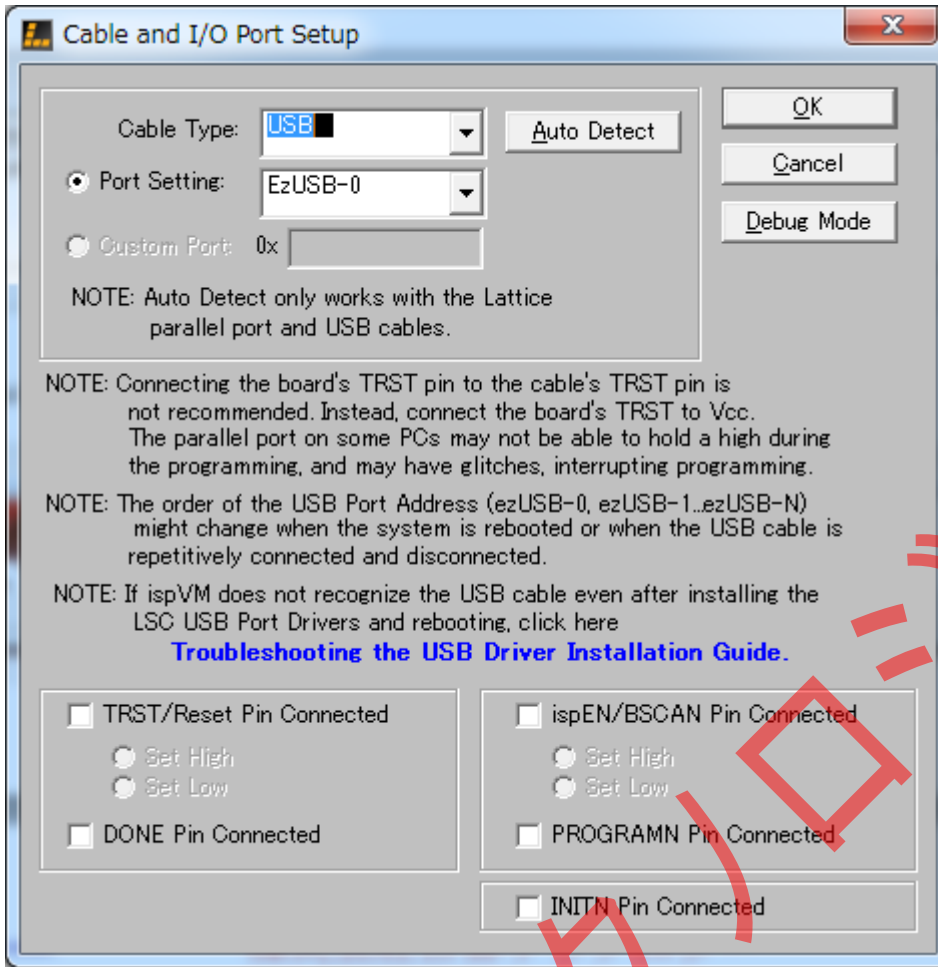
4) ispLEVER の「Tools」に「ispVM System」、又はツールバーに  をクリックし、プログラミングダウンロードのツールをオープンし、JED フォーマットのファイルをダウンロードする。

Add New Device をクリックする、-> Device のタイプを選ぶ（例えば LC4256V, T0FP100 パッケージ）、-> ダウンロードするロジックファイルを選ぶ、->ok をクリックする。

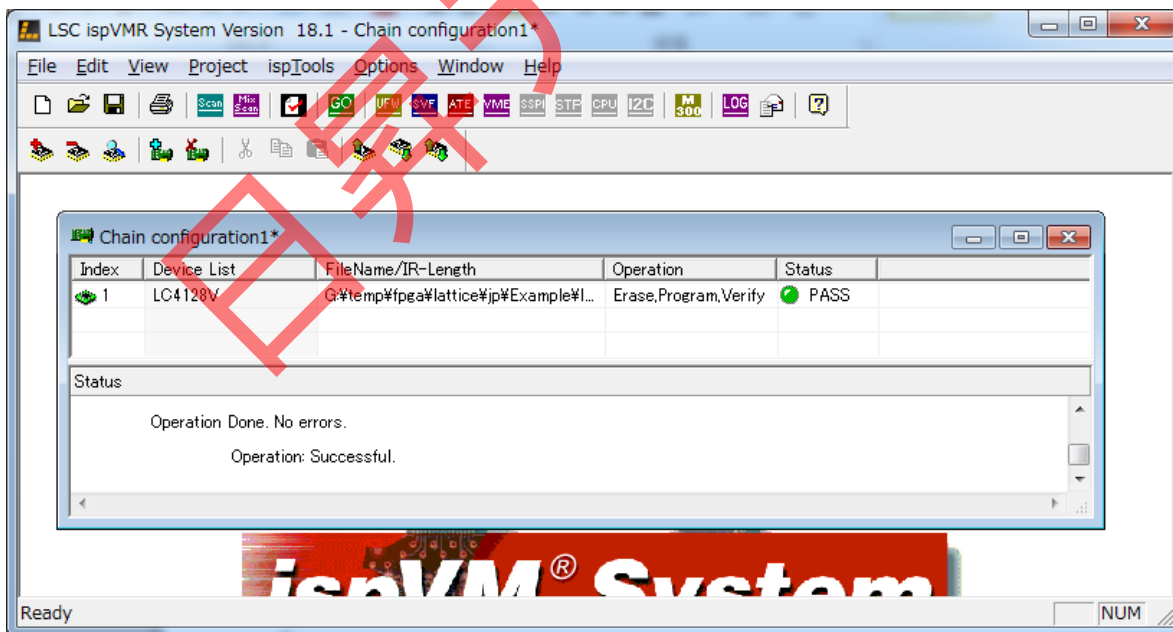




Options->Cable and I/O Port Setup をクリックして、Auto Detect ボタンをクリックする。



Project->Download をクリックする。



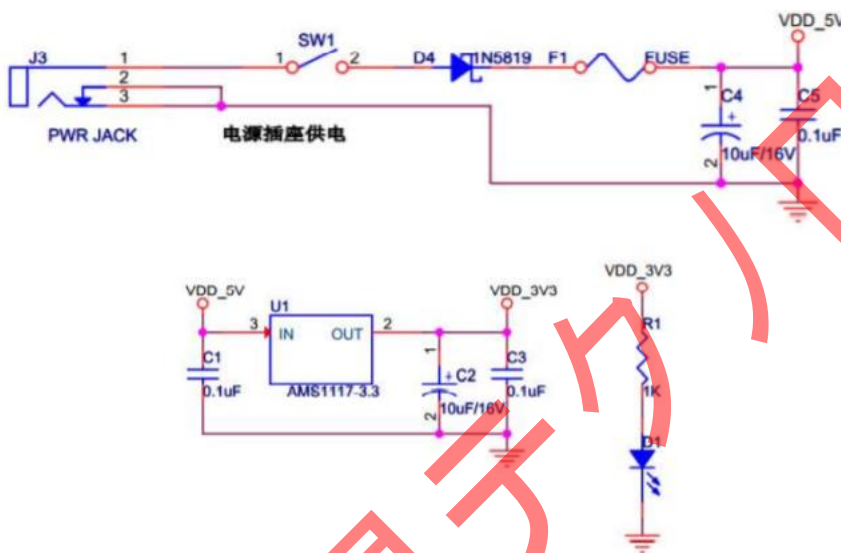
4 開発ボード回路説明

4.1 電源回路

本開発ボードの外部に5Vの電源を投入し、AMS1117-3.3Vレギュレーターを通じて3.3V電源を出力する。主にCPLD I/Oインタフェース、LED、リセット回路用の電源とする。

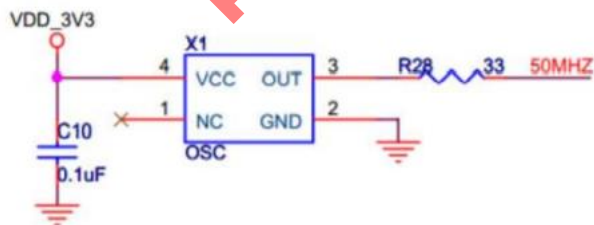
SW1はセルフロック電源スイッチ、D1は電源表示灯、D4は電源の逆接続を防ぐ表示灯である。

同時に、電力システムの安定性を向上させるために、電源入出力の部分に高品質の電解及びタンタルコンデンサを採用する。下図は回路図である。



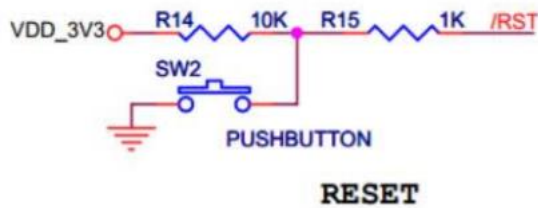
4.2 クロック回路

50Mアクティブ SMD 水晶発振器でシステムに実行クロックを提供する。CLK0 (PIN89) と繋ぐ。下図が回路図である。



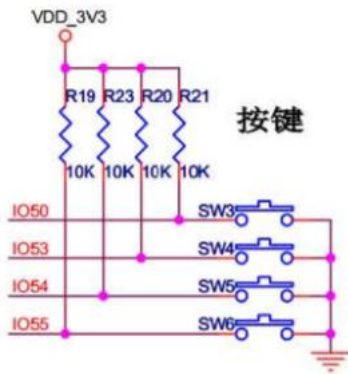
4.3 リセットキー回路

低レベルリセット、普通のボタンとしても使える。LC4256 の CLK3 (Pin88) と繋ぐ。下図が回路図である。



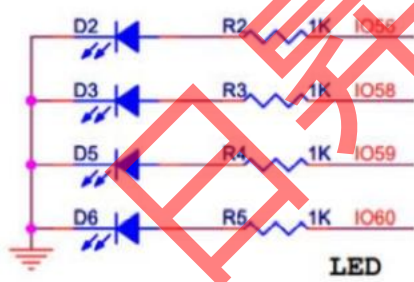
4.4 ユーザーキー回路

ユーザーキーを4つ搭載。回路図は下記：



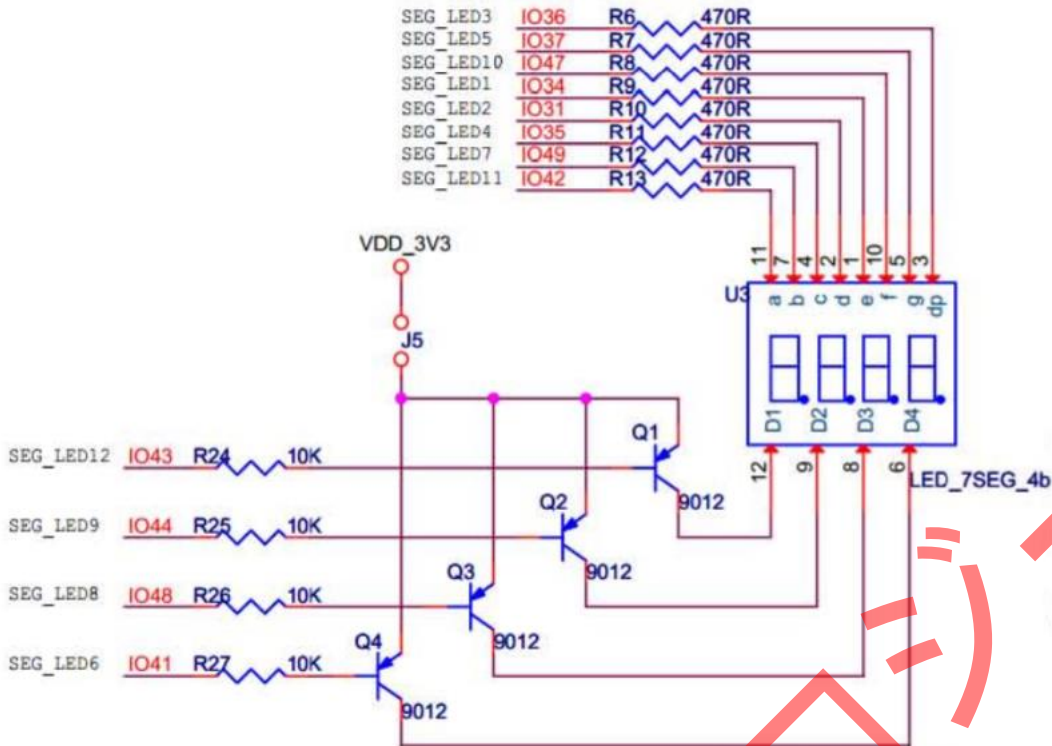
4.5 LED 回路

4つのLEDを搭載。ハイレベルで点灯。回路図は下記：



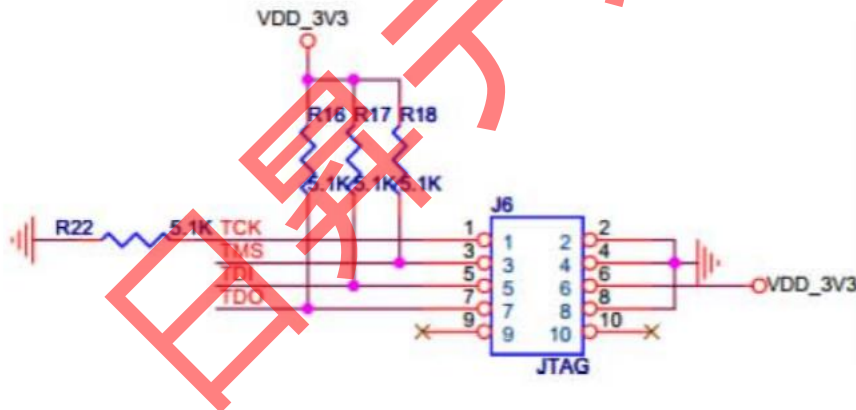
4.6 7SEG LED 回路

四桁7セグメント共通アノードLEDを搭載。使用する際、J5をショートする必要。



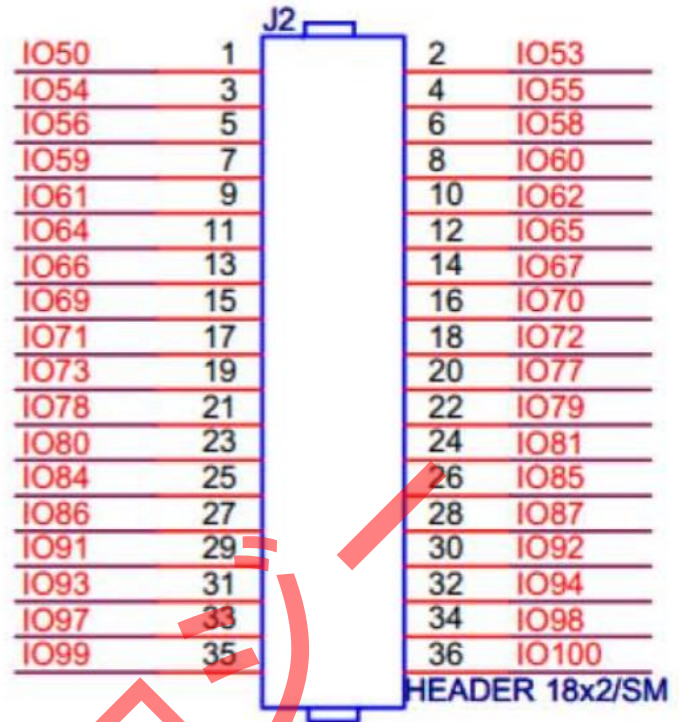
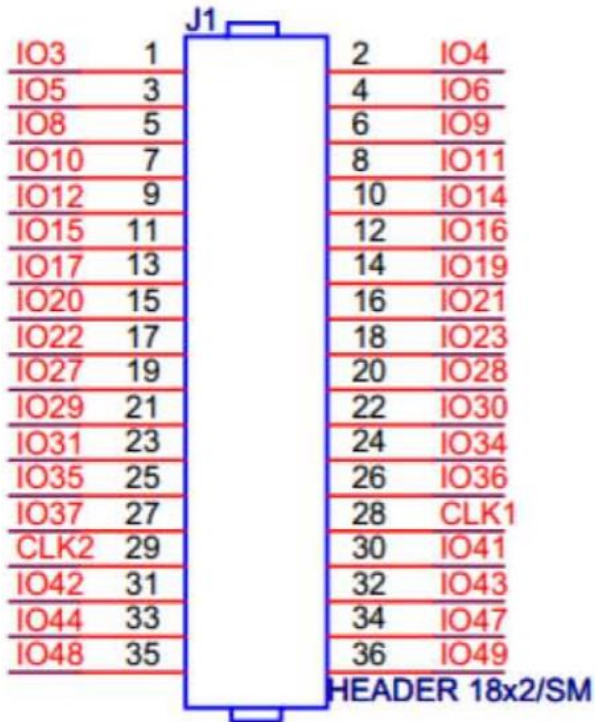
4.7 JTAG インタフェース回路

CPLD をダウンロードするために使用される。電源切れの場合に、CPLD のロジックが失わない。下図が回路図である。



4.8 外部拡張 IO

開発ボードは 2 つのピン配列で外部に引き出されている。ピン配列はダブル列、36pin、ピンの間隔は 2.54mm、ピンの定義は下図の通り：



以上。

日昇テクノロジー