



# Configuration Diagnostic Tool v1.0

June 19, 2017

Ng, Chin Chuan

PG Factory Applications Engineering

CEG Intel PSG

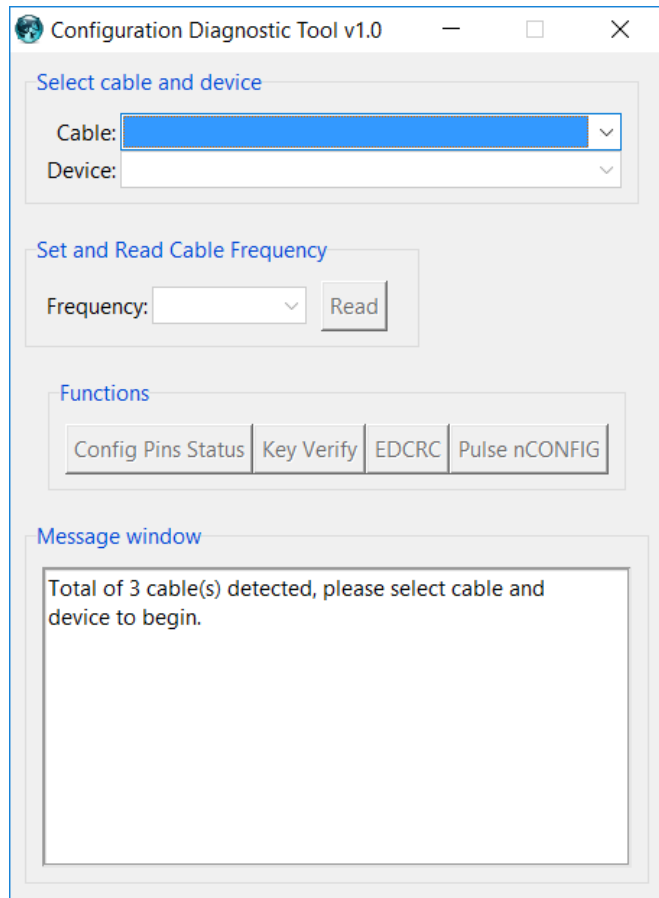
# Agenda

- Overview
- Supported FPGA Families
- Features
- Known Limitation
- Where to Download

# Overview (1/2)

Why we need this tool? Common questions:

- Configuration failure:
  - Have you set MSEL correctly?
  - Can you verify nCE connect to GDN?
  - What is nSTATUS and nCONFIG state?
  - Can you probe CONF\_DONE?
- Design security:
  - Can you run KEY\_VERIFY instruction to verify your key programming?
  - Do you program tamper protection bit?
- Error Detection CRC
  - Can you read EMR to understand the error type and location?
- How to force FPGA to reconfigure without pulse nCONFIG pin low?



# Overview (2/2)

The tool consists of:

- *config\_diag\_tool.tcl*
- *device\_data.tcl*

How to execute the tool:

- Your machine need to install Quartus Prime software in order to run this tool
  - Quartus Prime software come with Tcl interpreter
- If your Operation System has default Tcl interpreter to run .tcl file, then you may just double click *config\_diag\_tool.tcl*, or
- In Nios II Command Shell, execute the .tcl file with below command:
  - > *tclsh config\_diag\_tool.tcl*

# Supported FPGA Families

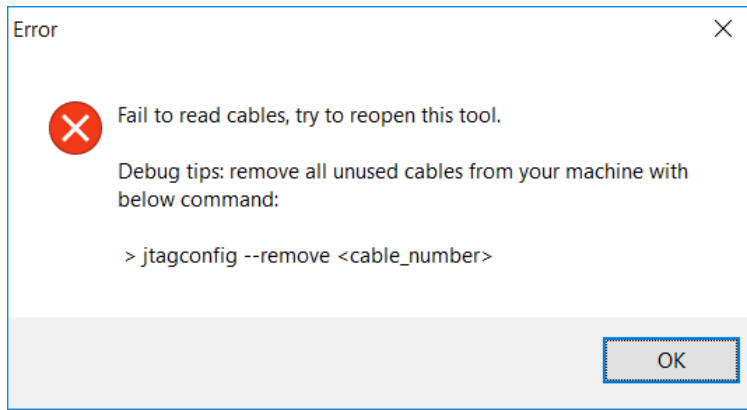
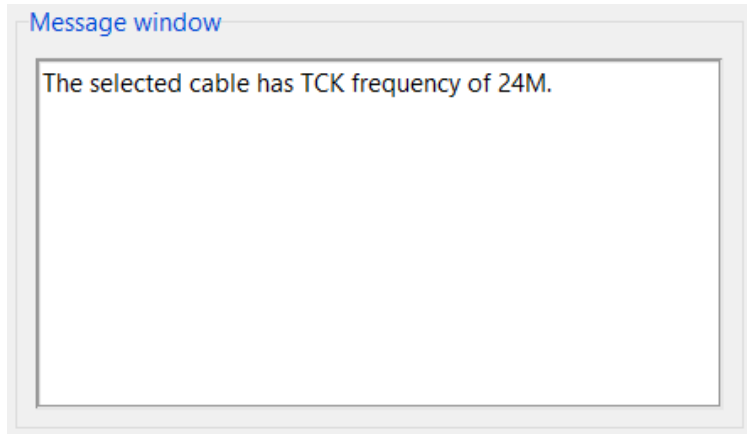
The tool can be made supportable for all Intel FPGA families, **except Stratix 10**, due to different architecture

- Currently supported
  - Stratix V, Arria V, Cyclone V including SoC devices
  - Arria 10 and SoC devices
- Support can be extended
  - All FPGA families prior to 28nm and 20nm devices

# Feature (1/6)

## Self-explanatory

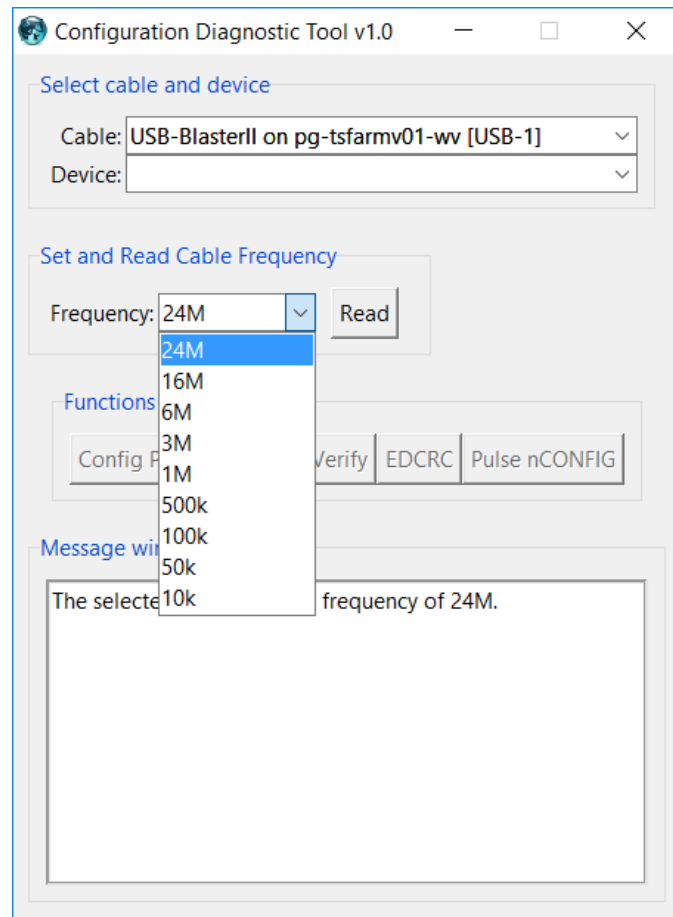
- Responses appeared in message window
- Debug tips when error occurred
  - More useful debug tips can be added from time to time



## Feature (2/6)

### USB Blaster II JTAG frequency configuration

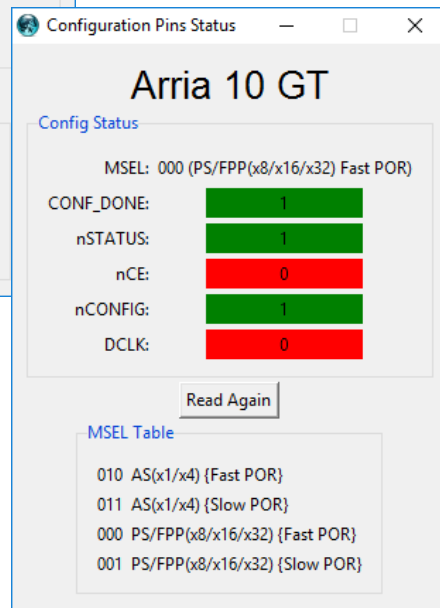
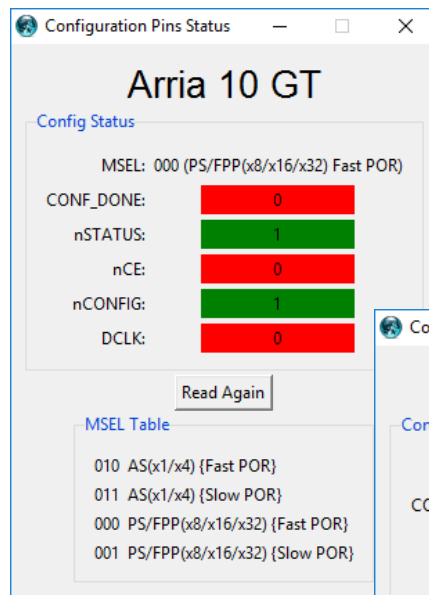
- Auto-detection upon cable selection
- Dropdown list for JTAG frequency setting



# Feature (3/6)

## Configuration Pins Status

- Read the FPGA configuration pins status
  - MSEL
  - CONF\_DONE
  - nSTATUS
  - nCE
  - nCONFIG
  - DCLK
- Enhancement can be made to allow custom pins to be read





# Features (4/6)

## Design Security

- Read the Key Verify Register and list down the bit information
  - Less dependent on User Guide

Key Verify Register for Arria 10 GT

### Arria 10 GT

Note: Bits not specified in the table are don't care.

Key Verify Register

Volatile Key [0]:	1	This bit is set when a volatile key has been successfully programmed into the device.
Attempt Non-volatile Key Programming [1]:	0	This bit is set to indicate that someone attempted to burn a non-volatile key in the OTP fused.
Disable Non-volatile Key [2]:	0	This bit is set to disable use of the non-volatile key.
Non-volatile Key [3]:	0	This bit is set to indicate that someone has successfully burned a non-volatile key into the OTP fuses.
Tamper Protection [4]:	0	This bit is set when FPGA is in Tamper Protection mode with either Non-volatile or Volatile key.
Volatile Key Lock [6]:	0	This bit is set to prevent the volatile key from being reprogrammed from external JTAG.
Force Configuration from HPS only [11]:	0	This bit is set when configuration is allowed from HPS only.
External JTAG Bypass [12]:	0	This bit is set to indicate that external JTAG is disabled.
HPS JTAG Bypass [13]:	0	This bit is set to indicate that HPS JTAG is disabled.
Disable Partial Reconfiguration and Scrubbing [14]:	0	This bit is set to indicate that external PR and external scrubbing (including HPS PR and HPS scrubbing) are disabled.
Disable Volatile Key [15]:	0	This bit is set to indicate that the volatile key is disabled.
Disable Key Related JTAG Instructions [17]:	0	This bit is set to indicate that external JTAG access to all key-related JTAG instructions is disabled.
JTAG Secure Mode [18]:	0	This bit is set to indicate that only mandatory JTAG instructions are allowed to be externally accessed.
Volatile Key Clear [20]:	0	This bit is set when the volatile key is successfully cleared from the device.

Read Again

# Feature (5/6)

## Error Detection CRC

- Read Error Message Register (EMR) to understand EDCRC failure location

The screenshot shows a software window titled "Error Message Register for Arria 10 GT". Inside, there's a section for "Arria 10 GT" with a note: "Note: The EMR value may not be correct when the FPGA is unconfigured or the EDCRC is not turned on." Below this is the "Error Message Register" section, which displays the following values:

- Full EMR[77:0] : 0x139054165133E572C904 (78 bits)
- Frame Address[77:62] : 0x4E41 (16 bits)
- Column-Based Double Word[61:60] : 0x1 (2 bits)
- Column-Based Bit[59:55] : 0x08 (5 bits)
- Column-Based Type[54:52] : 0x1 (3 bits)
- Frame-Based Syndrome[51:20] : 0x65133E57 (32 bits)
- Frame-Based Double Word[19:10] : 0x0B2 (10 bits)
- Frame-Based Bit[9:5] : 0x08 (5 bits)
- Frame-Based Type[4:2] : 0x1 (3 bits)
- Reserved[1:1] : 0x0 (1 bits)
- Column-Check-Bit Update[0:0] : 0x0 (1 bits)

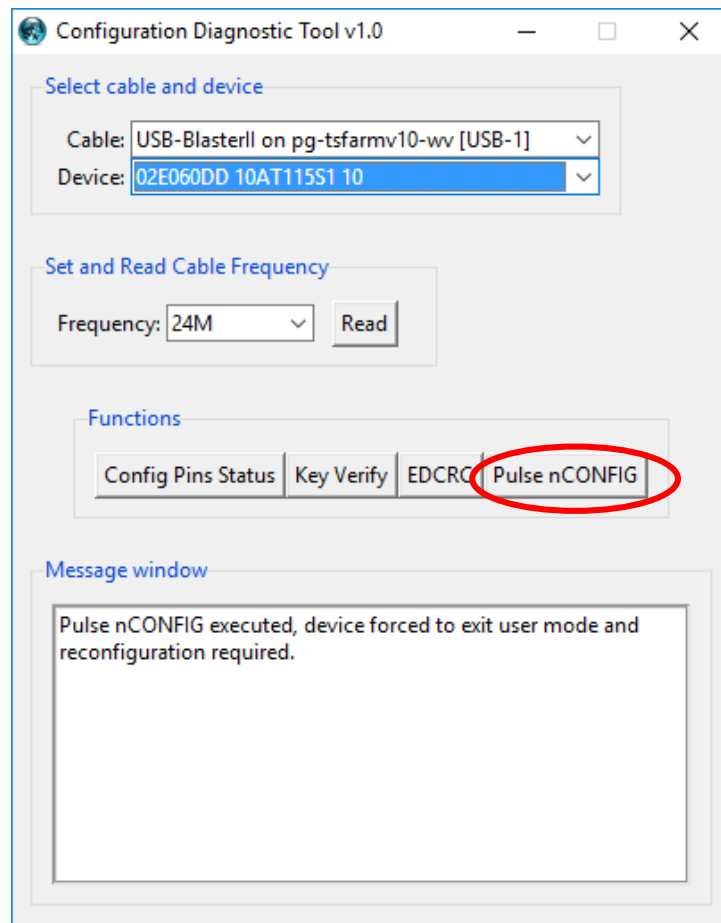
Below the register values is a "Read Again" button. Underneath that is the "Error Type Table" section, which contains the following table:

Frame-based [2:0]	b000	No error
Frame-based [2:0]	b001	Single-bit error
Frame-based [2:0]	b01X	Double-adjacent error
Frame-based [2:0]	b111	Uncorrectable error
Column-based [2:0]	b000	No error
Column-based [2:0]	b001	Single bit error
Column-based [2:0]	b01X	Double-adjacent error in a same frame
Column-based [2:0]	b10X	Double-adjacent error in a different frame
Column-based [2:0]	b110	Double-adjacent error in a different frame
Column-based [2:0]	b111	Uncorrectable error

# Features (6/6)

## Pulse nCONFIG instruction

- Force FPGA exit user mode remotely
  - This is not a documented feature
  - Useful for debug purpose



# Known Limitation

This tool should always work if Intel FPGA is the only device in JTAG chain.  
Some known limitation such as:

- If the JTAG chain consists of devices with unknown IR length (3<sup>rd</sup> party devices) to jtagconfig command
  - This tool uses “jtagconfig --debug” command to read cables and devices
- Unable to handle virtual JTAG
  - Buffer bit appeared in JTAG chain causes problem

```
Command Prompt

C:\Users\cng3>jtagconfig --debug
1) USB-BlasterII on pg-tsfarmv01-wv [USB-1]
  02A060DD  5AGTFD7(H3|K3)/5AGXBB7D(4|6)/.. (IR=10)
    Node 0C206E00  JTAG PHY #0
    Node 00486E00  Source/Probe #0
    Node 30006E00  SignalTap #0
    Design hash   AAB262BE138650F30731
  02A060DD  5AGTFD7(H3|K3)/5AGXBB7D(4|6)/.. (IR=10)
  020A40DD  5M(1270ZF324|2210Z)/EPM2210 (IR=10)
    Node 00406E01  (110:8) #1
    Node 00406E00  (110:8) #0
    Node 00406E7B  (110:8) #123

  Captured DR after reset = (02A060DD02A060DD020A40DD) [96]
  Captured IR after reset = (15555555) [30]
  Captured Bypass after reset = (0) [3]
  JTAG clock speed 24 MHz

2) USB-Blaster on pg-tsfarmv18-wv [USB-0]
  028010DD  EP4CGX15 (IR=10)
    Node 0C006E00  JTAG UART #0
    Node 19104600  Nios II #0
    Design hash   D7075B2FD58AF1FF68D9
  020A40DD  5M(1270ZF324|2210Z)/EPM2210 (IR=10)
    Node 00406E00  (110:8) #0

  Captured DR after reset = (028010DD020A40DD) [64]
  Captured IR after reset = (555555) [20]
  Captured Bypass after reset = (0) [2]

3) Remote server pg-tsfarmv10-wv: Authentication failure

C:\Users\cng3>
```

