SMH Technologies®

18/07/2024

Driver v. 5.03

Moreno Ortolan

Interfacing FlashRunner 2.0 with CPLD

SMH Technologies S.r.l.

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

CPLD Introduction

A **Complex Programmable Logic Device CPLD** is a programmable logic device with complexity between that of PALs and FPGAs, and architectural features of both.

The main building block of the CPLD is a macrocell, which contains logic implementing disjunctive normal form expressions and more specialized logic operations.

CPLD device must be programmed using a **SVF** or **VME** source file.

Starting for example from a **.pof** or **.jed** file, you can create a **.svf** to program the CPLD.
The conversion steps involve the use of silicon producer tools, to convert the **.pof** or **.jed** into an **.svf** file
Once the file to program is an **.svf** file it has to be converted into a **.frb** file.

The SVF to FRB conversion can be done with **Workbench v. 3.09.01** or higher and/or with command line **FRB converter v. 3.09.01** or higher.
You can also convert the SVF to VME with **svf2vme12.exe** contained in the FlashRunner Workbench installation folder.

If your SVF file has some comments inside that and you convert it into FRB or VME file using the setting “keep comments”, only comments starting with "**!->**" will be shown on the Workbench Real Time Log.

For example, if inside the SVF file you add the following comments:

```
!-> Check the IDCODE
!-> Program Bscan register
!-> Enable the programming mode
!-> Erase the device
!-> Read the status bit
!-> Program Fuse Map
!-> Program USERCODE
!-> Read the status bit
!-> Verify Fuse Map
!-> Verify USERCODE
!-> Program DONE bit
!-> Read the status bit
!-> Exit the programming mode
!-> Verify SRAM DONE Bit
```

During the program you can observe this into Real Time Log from CPLD driver version **5.02**:

```
---#TPCMD PROGRAM C
Processing Virtual Machine File (VME).
* VME Version: ____12.1.
* VME file type: compressed.
* VME Vendor: Lattice.
* SVF frequency is different from selected frequency.
  * Requested Clock is 1.00 MHz.
  * Generated Clock is 1.00 MHz.
  * Good samples: 16 [Range 0-15].
  * ID-Code Read: 0x01285043.
* [VME] - Check the IDCODE
* Elapsed time is 0.002 s.
* [VME] - Program Bscan register
* Elapsed time is 0.005 s.
* [VME] - Enable the programming mode
* Elapsed time is 0.002 s.
* [VME] - Erase the device
* Elapsed time is 1.875 s.
* [VME] - Read the status bit
* Elapsed time is 0.008 s.
* [VME] - Program Fuse Map
* Elapsed time is 0.106 s.
* [VME] - Program USERCODE
* Elapsed time is 0.959 s.
* [VME] - Read the status bit
* Elapsed time is 0.014 s.
* [VME] - Verify Fuse Map
* Elapsed time is 0.102 s.
* [VME] - Verify USERCODE
* Elapsed time is 0.536 s.
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

```
* [VME] - Program DONE bit
* Elapsed time is 0.018 s.
* [VME] - Read the status bit
* Elapsed time is 0.014 s.
* [VME] - Exit the programming mode
* Elapsed time is 0.005 s.
* [VME] - Verify SRAM DONE Bit
* Elapsed time is 0.002 s.
Closed ISPVM machine.
Time for Program C: 3.382 s.
>|
```

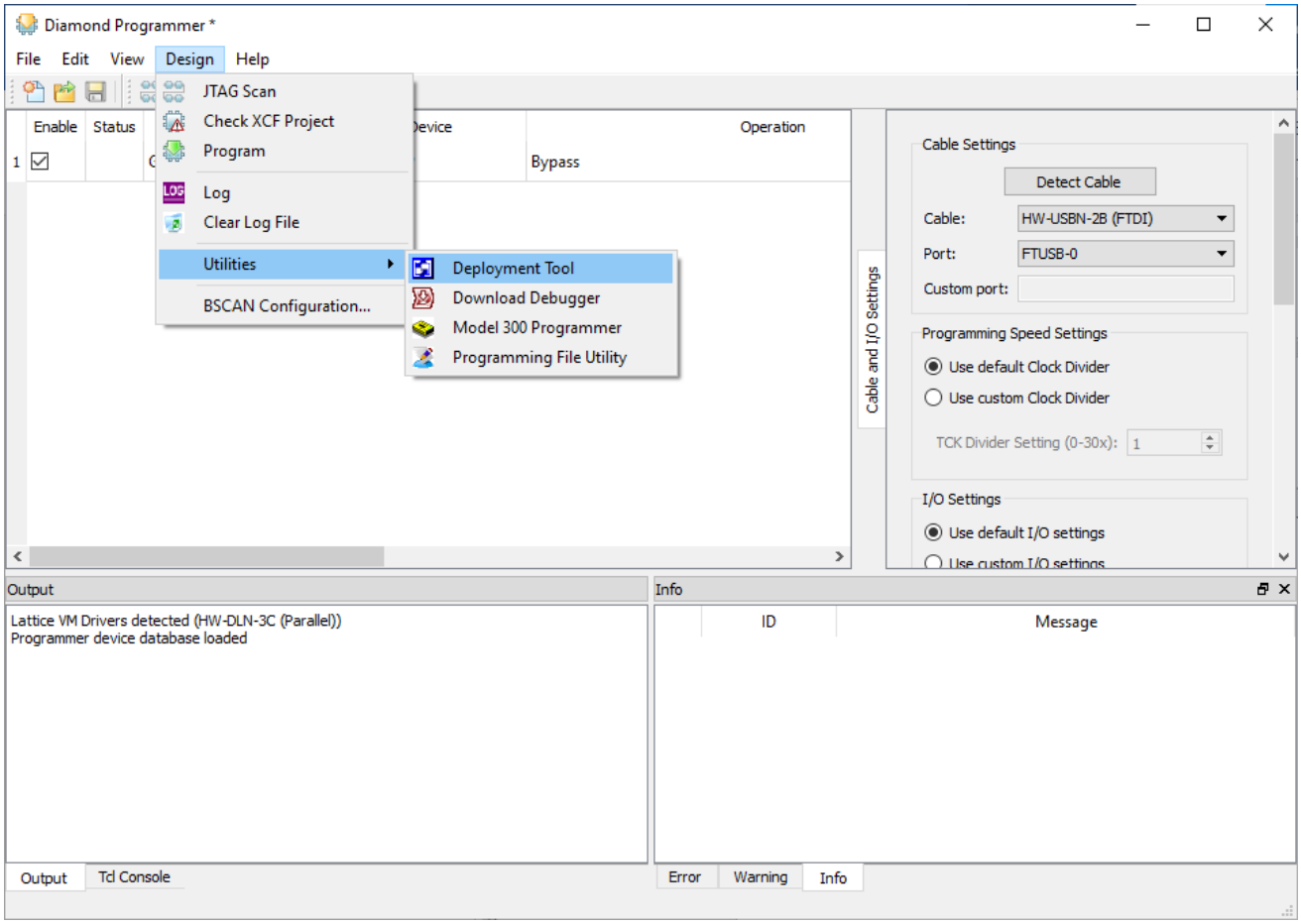
CPLD generation of SVF file

Typically, we need to execute some steps to obtain an FRB file starting from a POF/JED/Another format file for CPLDs.
Now we make some examples on of how to convert a JED file to an FRB file.
POF or JED files are typically based on CPLD semiconductor manufacturer, so we will use the following silicon producer tools:

- Lattice Diamond Programmer
- Xilinx Impact

Conversion from JED to SVF using Diamond Programmer (Lattice CPLDs)

For Lattice CPLDs you can use Diamond Programmer.
From Diamond Programmer open the Deployment Tool utility:



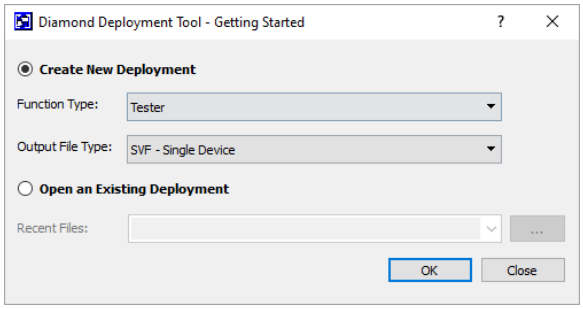
HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

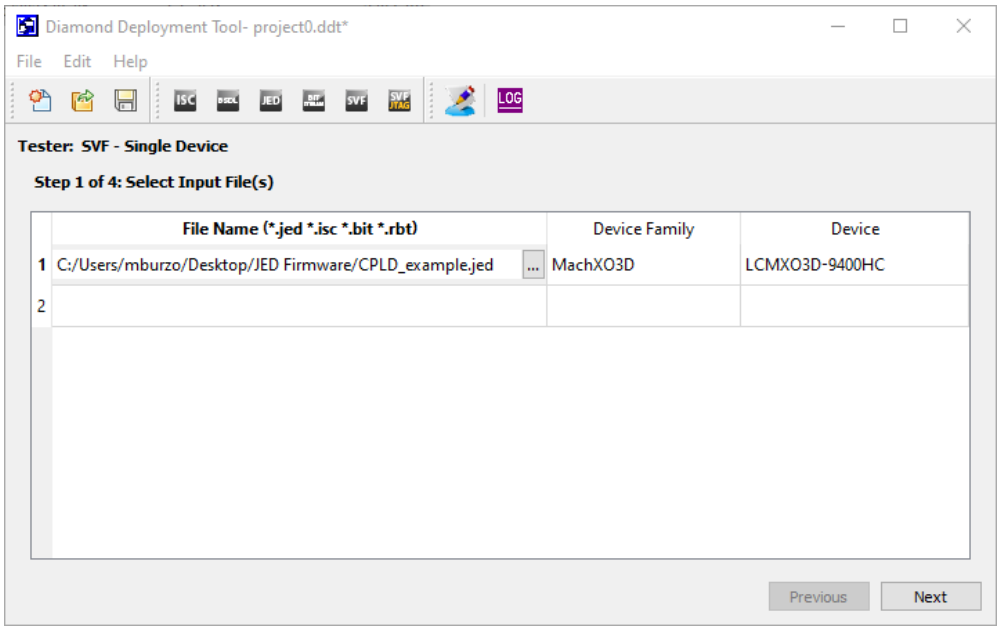
D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

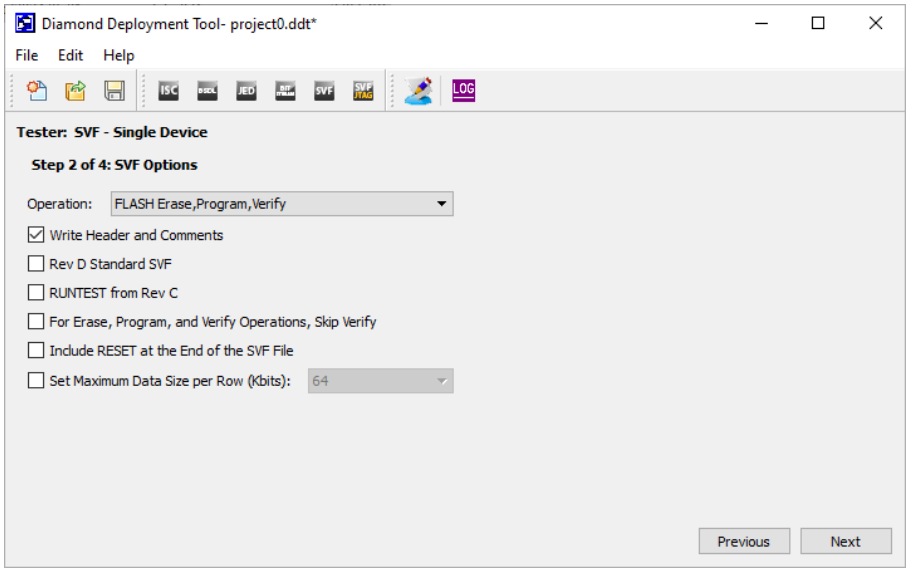
Select Tester – SVF Single Device:



Add JED file:



Select the operation:



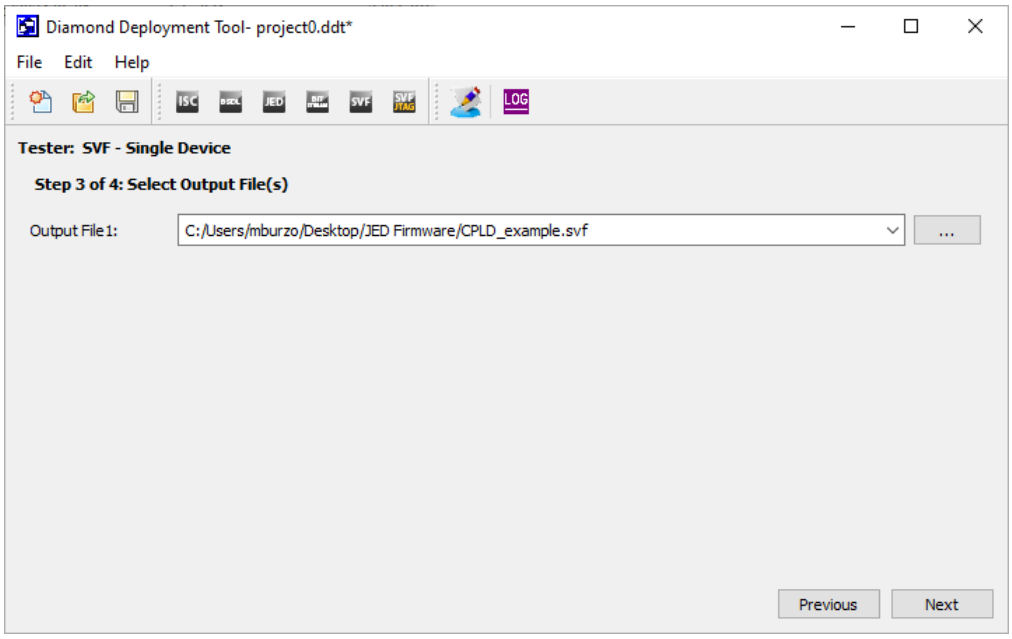
HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

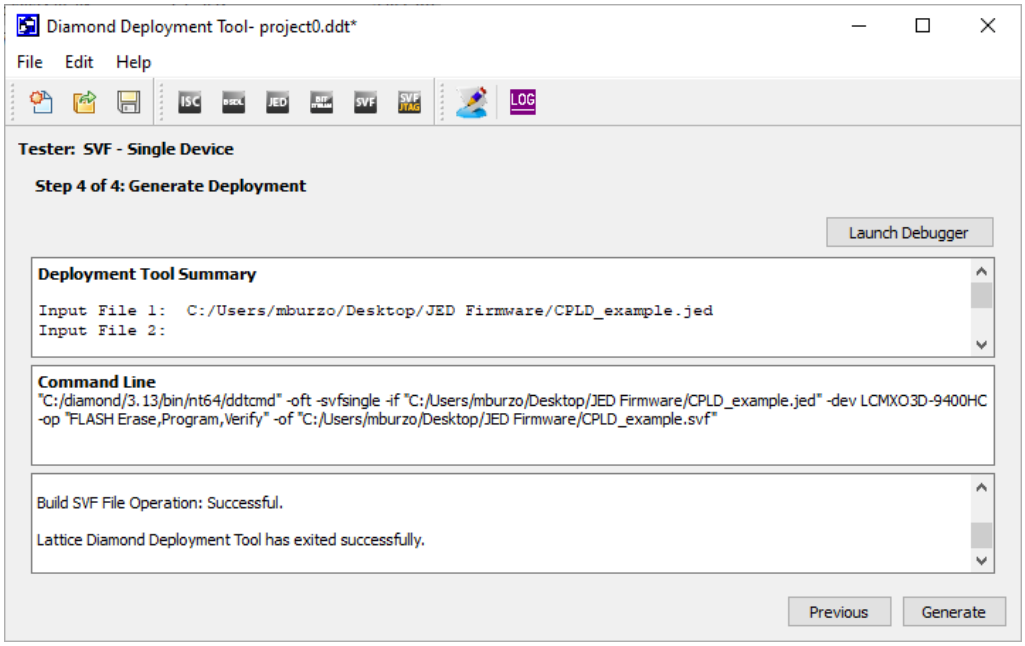
D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

Select the output file:



Generate the SVF:



HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

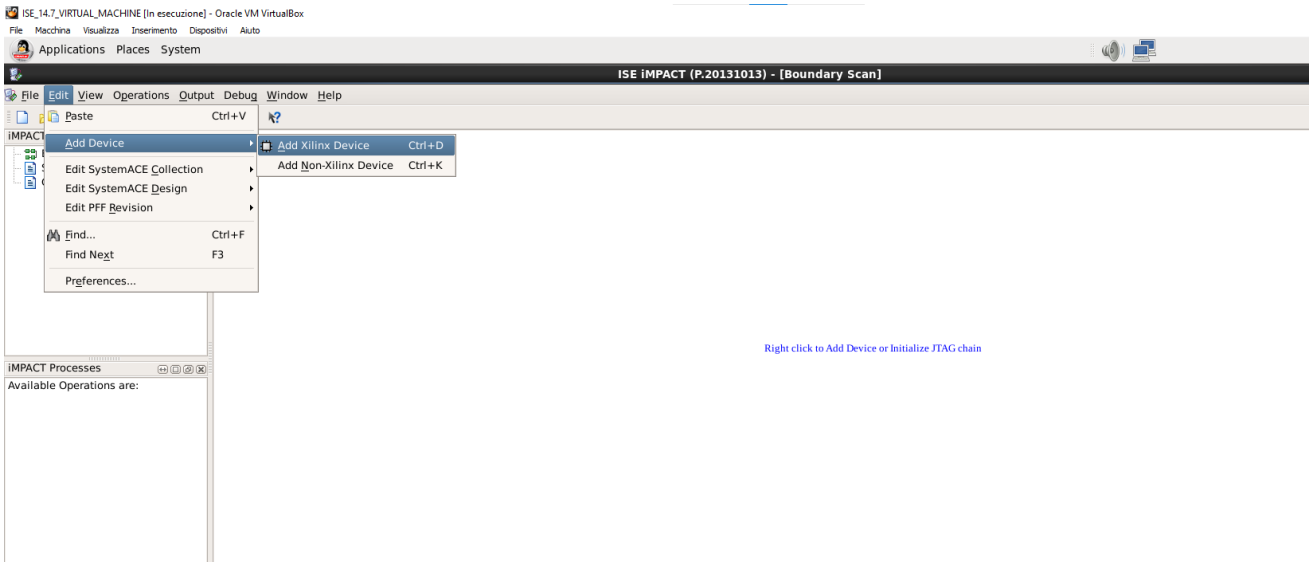
Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

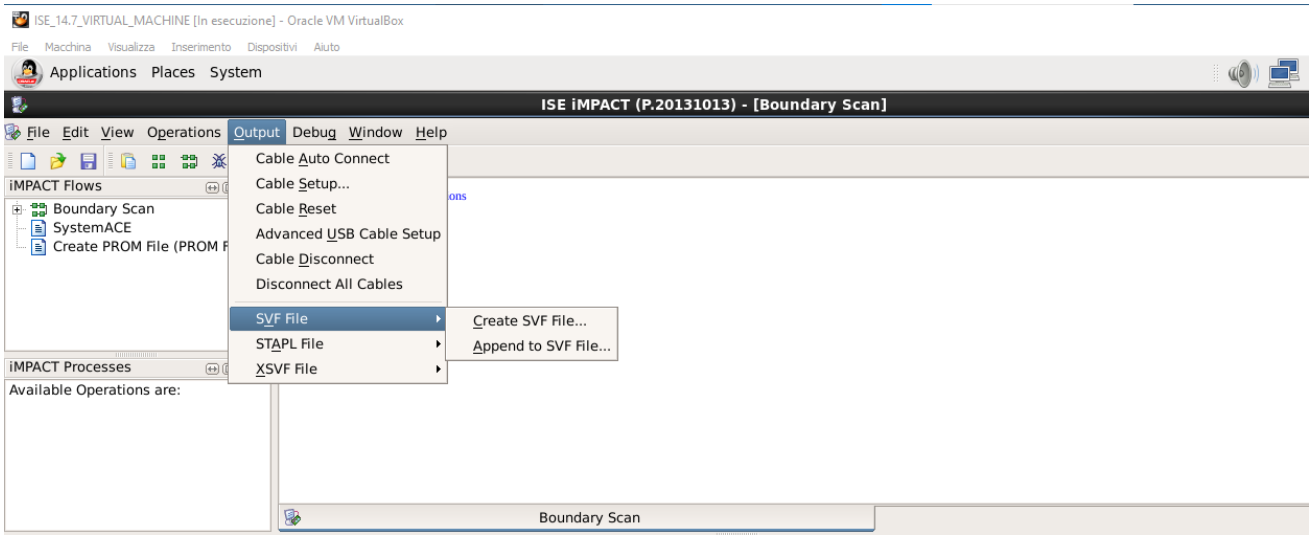
→smh-tech.com
info@smh-tech.com

Conversion from JED to SVF using Impact (Xilinx CPLDs)

For Xilinx CPLDs you can use Impact.
Add Xilinx Device by selecting the JED file:



Create SVF File:



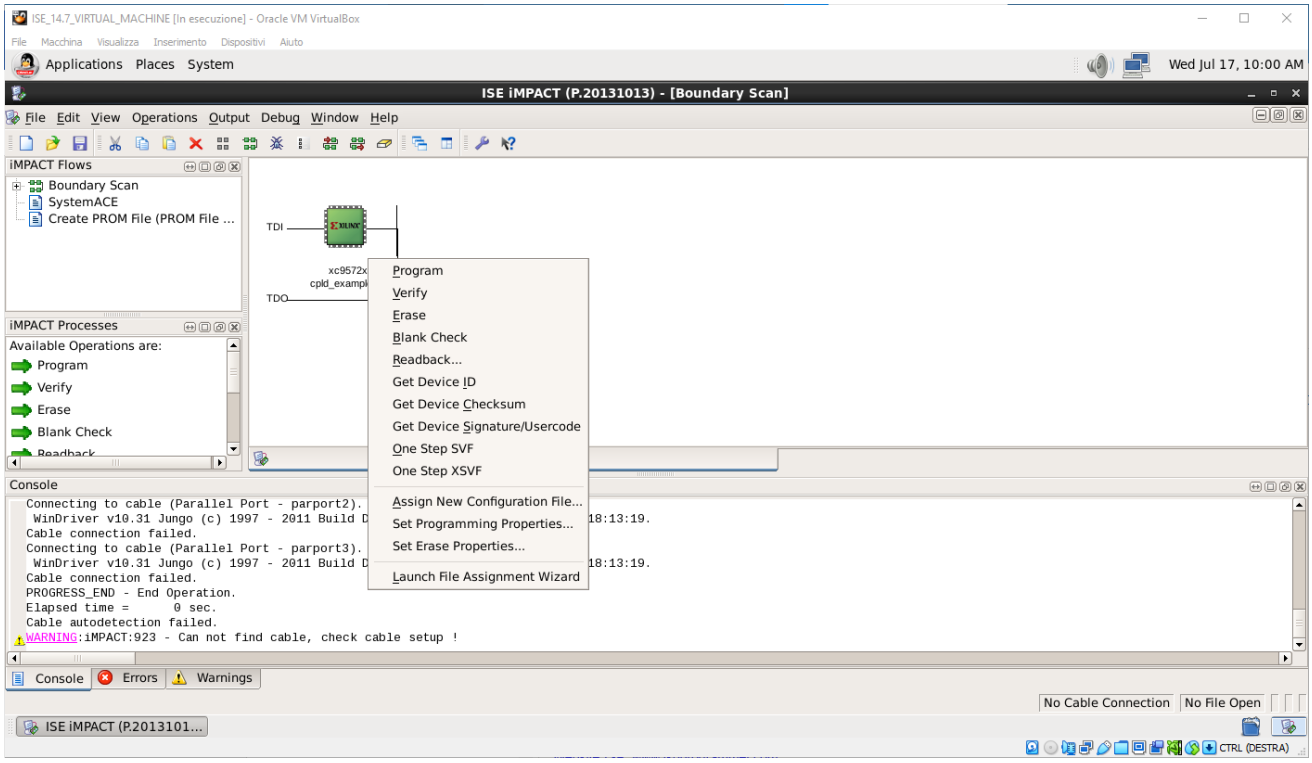
HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

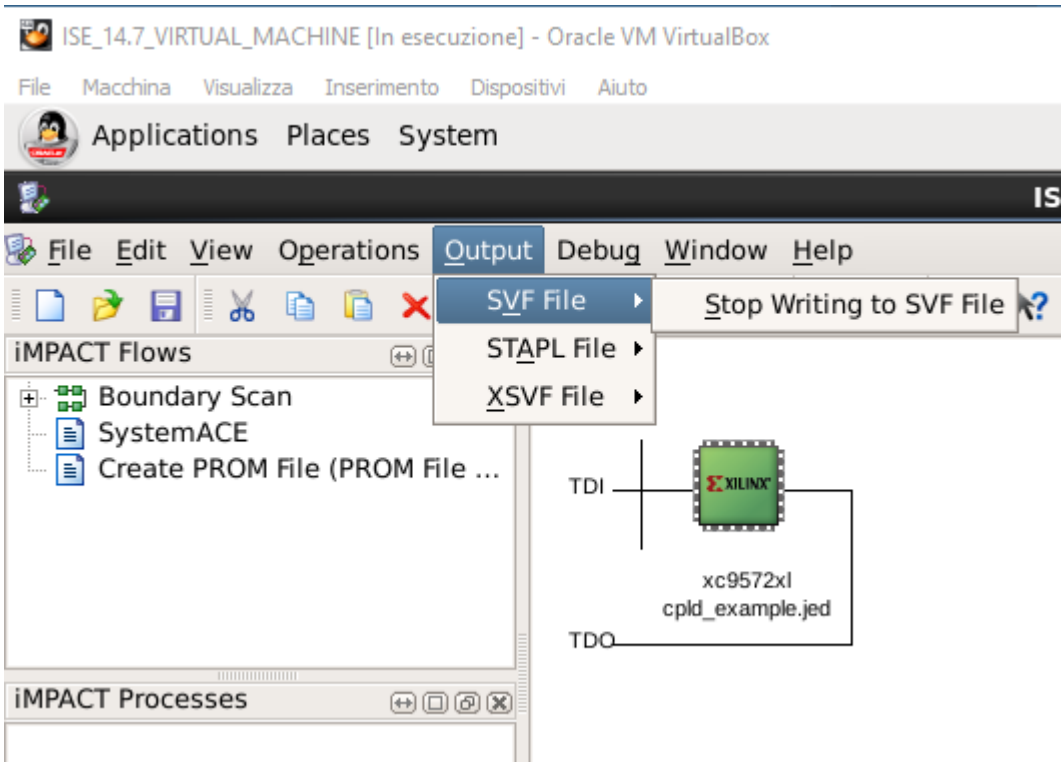
D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

Right click on device, then Program:



Stop writing into SVF file:



UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

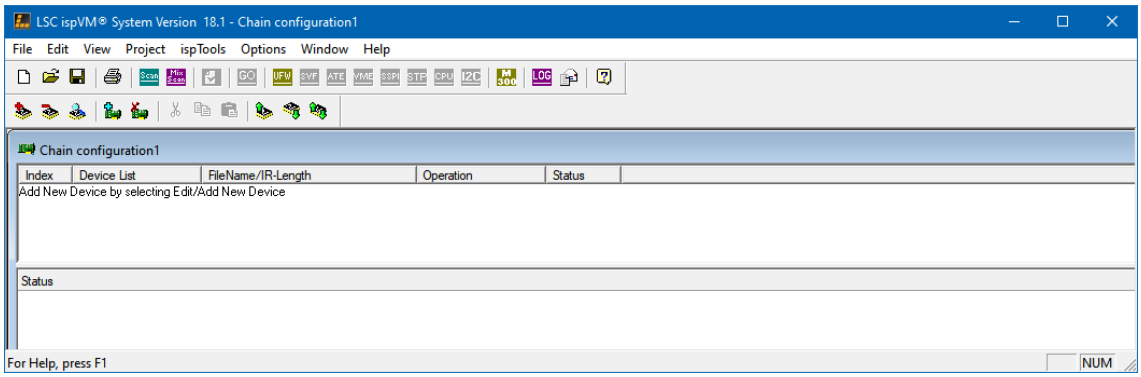
D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

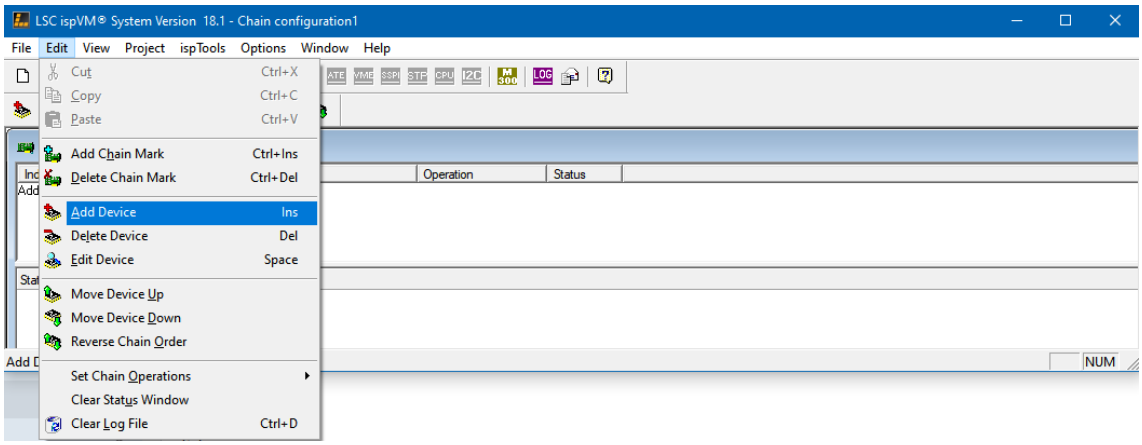
info@smh-tech.com

Conversion from JED to SVF using ispVM System (Lattice CPLDs – Legacy)

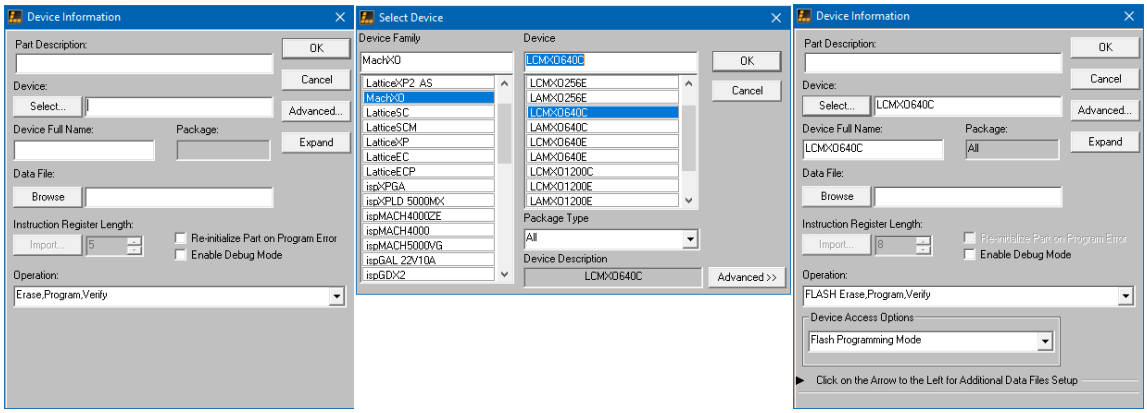
For Lattice CPLDs you can use also ispVM, but Diamond is preferable.
In this example we use an LCMXO640C CPLD of MachXO family.
When you open ispVM, you can find a window like this:



Now you need to add a new device, so click on Edit -> Add Device:



Now click on Select button into new window and add your device, in this case **LCMXO640C**:



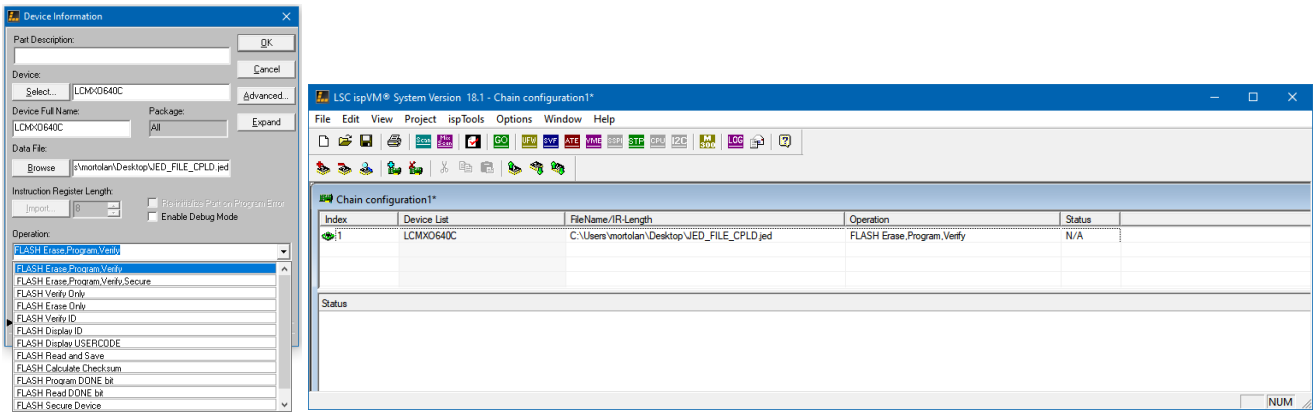
Now you can add the JED file into Data File section and you need to choose which operation you want to perform based on JED file.
In this case we only select Erase, Program and Verify.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

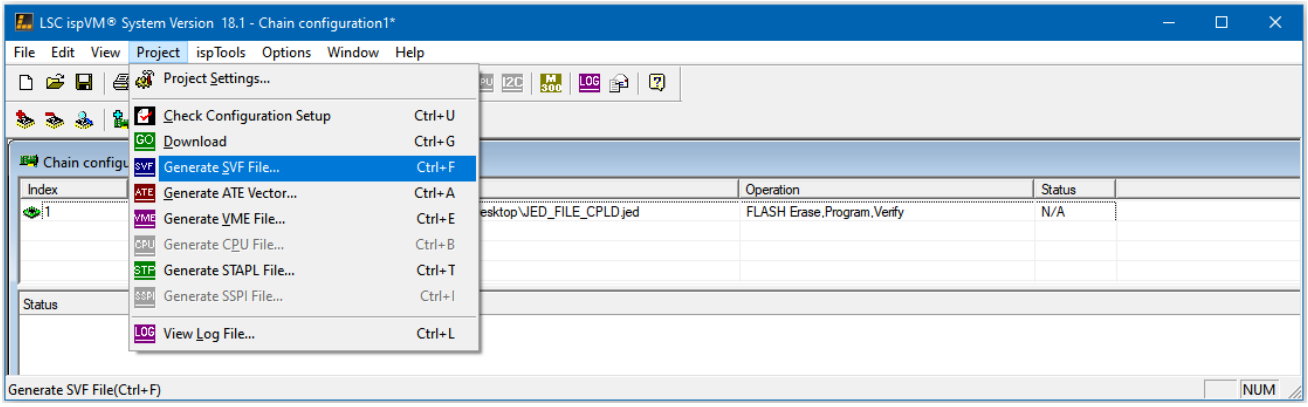
Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

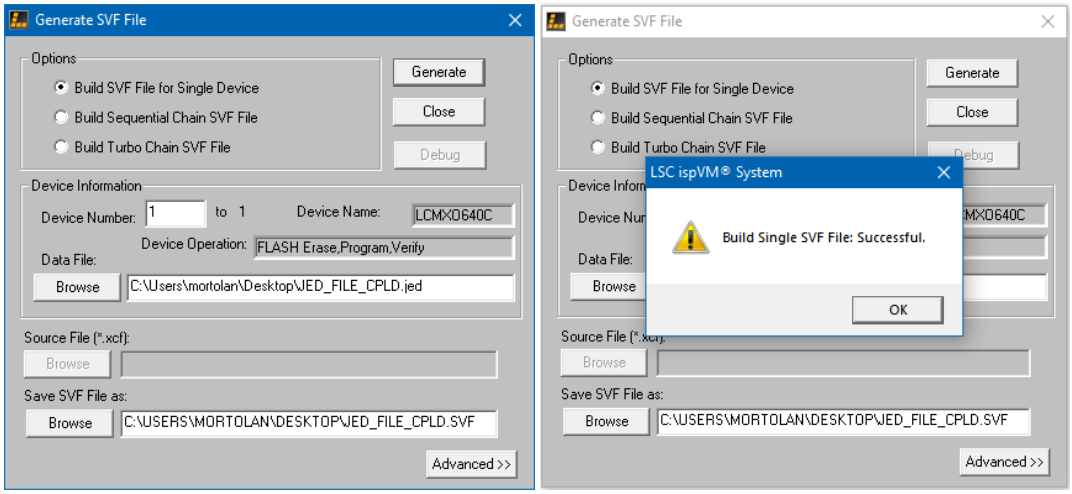
→smh-tech.com
info@smh-tech.com



Now on Project Menu you can select Project -> Generate SVF File:



In the grey window please select Build SVF File for Single Device and if not present, add the JED file into Data file section.
Select device number to 1.
Now add the SVF file name output into SVF file section.
At the end, click Generate to create the SVF file.



HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

SVF file analysis

By opening the generated SVF file with a simple editor, you can find a content like this:

```
! Initialize
! Row Width   :236
! Address Length :551
HDR 0;
HIR 0;
TDR 0;
TIR 0;
ENDDR   DRPAUSE;
ENDIR   IRPAUSE;
FREQUENCY   1.00e+006 HZ;
STATE   IDLE;
RUNTEST IDLE     15 TCK   1.00E-003 SEC;

!-> Check the IDCODE

! Shift in IDCODE(0x16) instruction
SIR 8    TDI  (16);
SDR 32   TDI  (FFFFFFFF)
        TDO  (01285043)
        MASK (FFFFFFFF);

! Program Bscan register

! Shift in Preload(0x1C) instruction
SIR 8    TDI  (1C);
SDR 320 TDI  (FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF);

! Enable the programming mode

! Shift in ISC ENABLE(0x15) instruction
SIR 8    TDI  (15);
RUNTEST IDLE     5 TCK   1.00E-003 SEC;
```

In this file you can find data to be programmed into CPLD but also all JTAG operations to be performed.
So, the size of SVF file typically is bigger that JED/POF files.

Other than that, all lines starting with "**!**" or "**//**" are considered comments and when you try to convert this SVF file to an VME file you have the possibility to remove all comments.

The SMH CPLD driver can print comments only if these comments starting with "**!->**".
We decide to do this because if the driver prints all the comments, it would create a very high overhead and therefore a loss of performance.

So, if you modify by your hand a comment in this way, SMH CPLD driver can print it:

```
* [VME] - Check the IDCODE
* Elapsed time is 0.002 s.
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

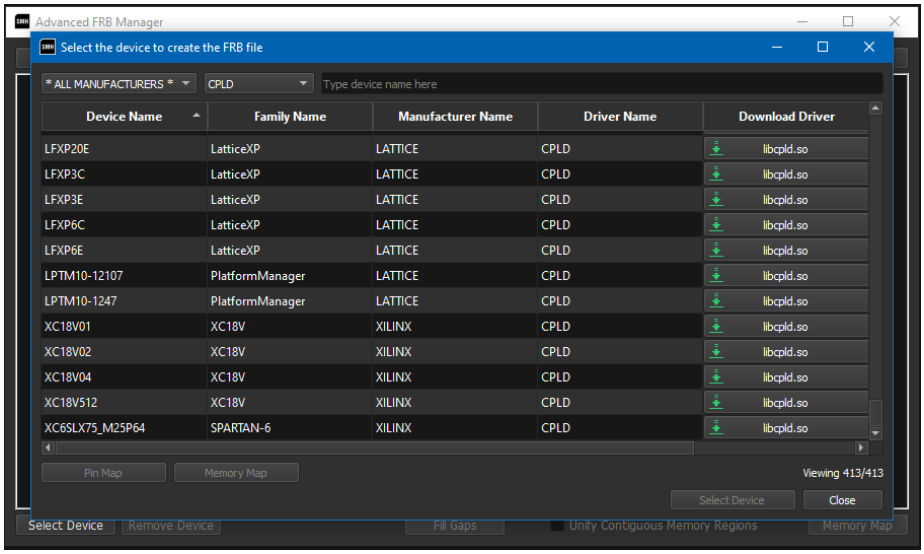
D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

CPLD SVF Conversion to FRB using Workbench 3.09.01

Now we can move to point two and we want to convert the **SVF** file to **FRB** file directly.
To do that, we need to use the Workbench **3.09.01** or higher version.

So, let's start with the **Workbench,** click on **Tools -> FRB Manager -> Advanced FRB Converter**



Now you can select the **CPLD** device and click on **Add** button.

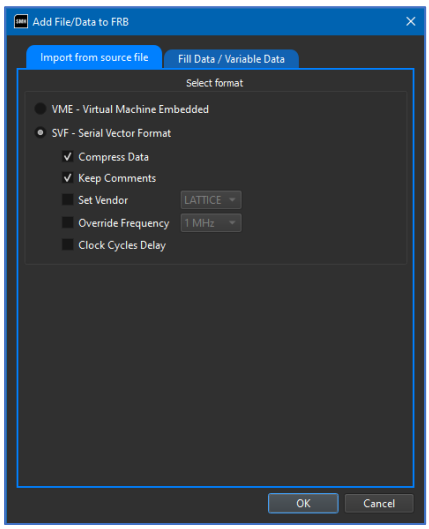
For CPLD devices you can find two possible options:

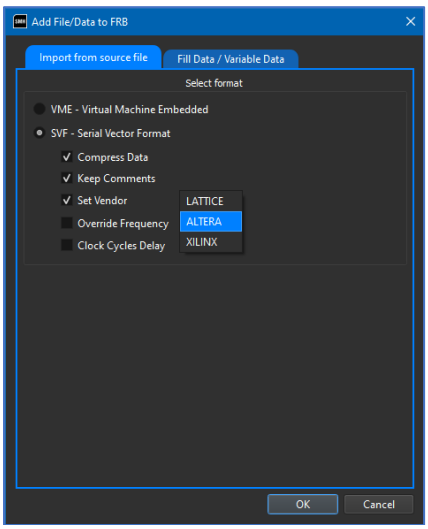
- **VME - Virtual Machine Embedded**
- **SVF - Serial Vector File**

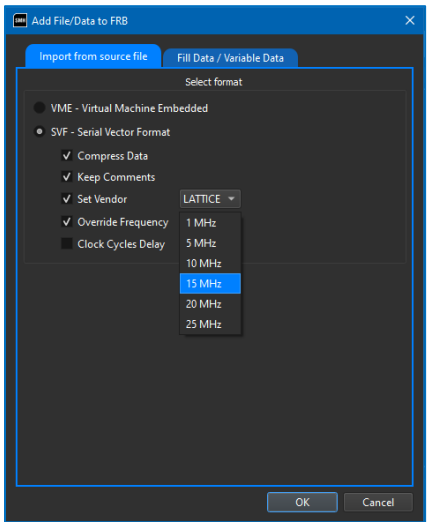
Here you have two options:

- If you select VME you can import a **Virtual Machine Embedded** file, created for example using svf2vme converter.
- If you select SVF you can import a **Serial Vector File** directly.

For **SVF** file you can choose multiple options:







HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

- **Compress Data**:
  With this option you can automatically compress SVF bytes when converting from SVF to FRB.
  Typically the size of an SVF file is larger than the result FRB size.
  This is quite obvious because for example if in the SVF file there is the RUNTEST command which occupies 7 bytes, in the FRB file RUNTEST will be replaced by a single byte which represents the RUNTEST command in the form of an opcode.
  However, if there are many consecutive equal bytes in the SVF file, such as 0x00, 0xFF or repeated sequences of bytes, it is possible to use the Compress Data option to further compress the generated FRB file.

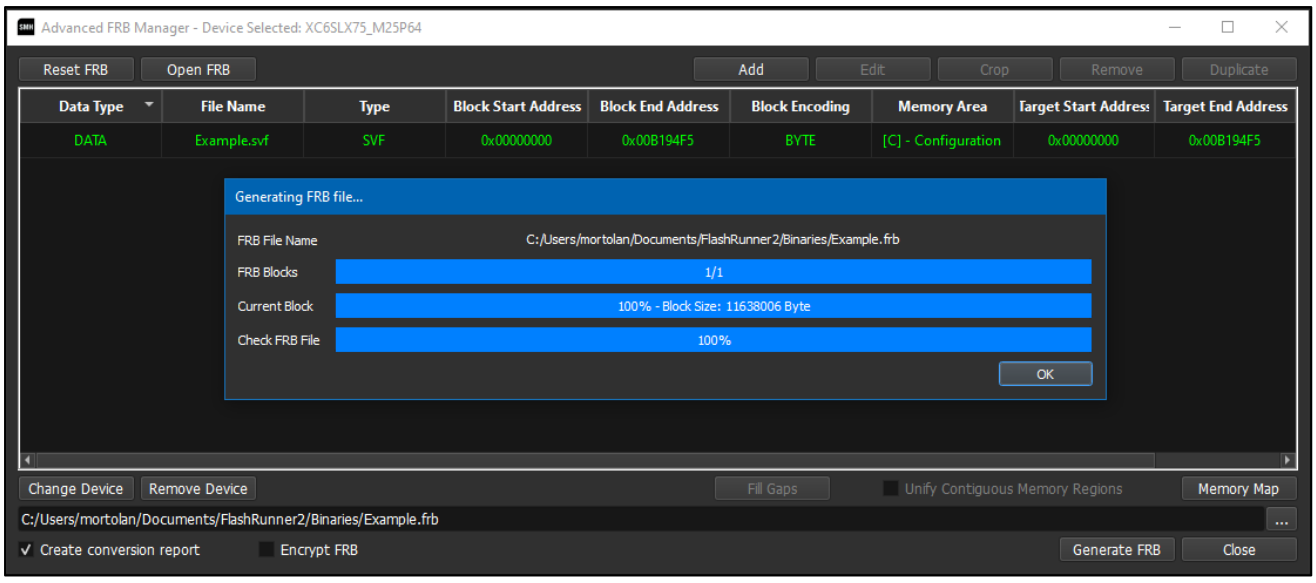
- **Keep Comments**:
  With the Keep Comments option you can keep or remove the comments present in the SVF file during the conversion to FRB file.

- **Set Vendor**:
  With Set Vendor option it's possible to set in the FRB file the vendor of the device being programmed.
  This value is not actually used in programming operations and is only used as a comment in the Real Time Log.
  Possible options for Set Vendor are Lattice, Altera, and Xilinx.
  By default, this parameter is not checked and therefore the default vendor set is Jtag Standard.

- **Override Frequency**:
  Typically in the SVF file there is a parameter called FREQUENCY set automatically by the tool used to create the SVF file.
  If it is not present, it is assumed that all delays present in the file are absolute and do not depend on frequency.
  If, however, the delays are set as clock strokes in the file, when the frequency changes, the necessary clock strokes must be recalculated so that the elapsed time is always the same.
  With override frequency you can modify the FREQUENCY value of the SVF file (The SVF file is not modified but the frequency value is changed during the conversion to FRB).

- **Clock Cycles Delay**:
  With this parameter you can decide if you want to add a delay proportional of clock cycles and frequency after all clock cycles commands.
  For example, if you set this parameter to YES and in the SVF file there's a RUNTEST command, you add the clock cycles and after that you add a delay proportional to clock cycles count.
  By default, this value is set to NO and typically you can leave this parameter set to NO.

Now import your **SVF** file and create the related **FRB**.



HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

CPLD SVF Conversion to VME using svf2vme

This method is obsolete because Workbench 3.09.01 is able to convert SVF file to VME file, but if you want you can continue to use it.

Now we can move to point two and we want to convert the **SVF** file to **VME** file.
To do that, we need to use the **svf2vme12.exe** provided by LATTICE. You can use this tool to convert SVF file from other CPLD silicon producer because this application is only a simple converter from SVF to VME.

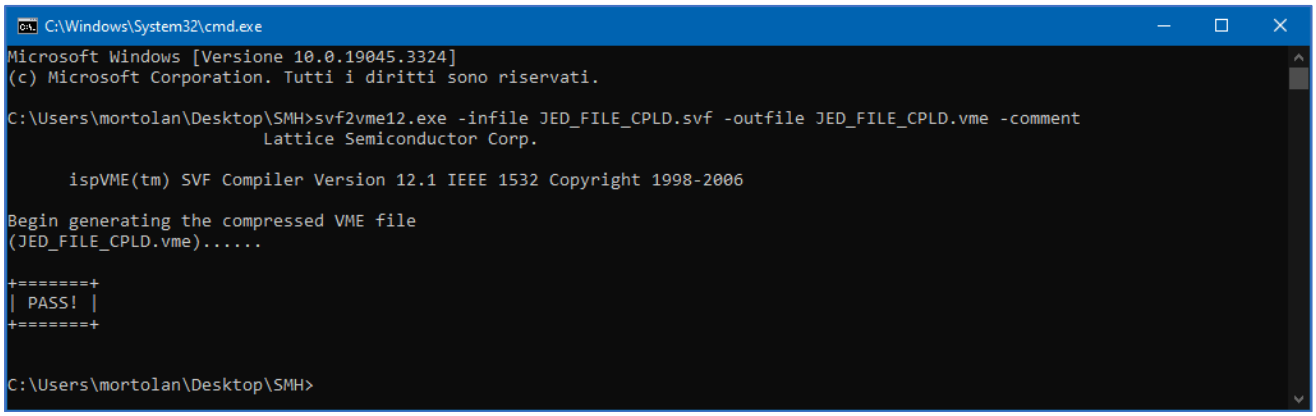
So, let's start with the svf2vme12.exe syntax:

svf2vme12.exe -infile <source file>.svf -outfile <output file>.vme -max_tck 1000000
svf2vme12.exe -infile <source file>.svf -outfile <output file>.vme -max_tck 1000000 -comment

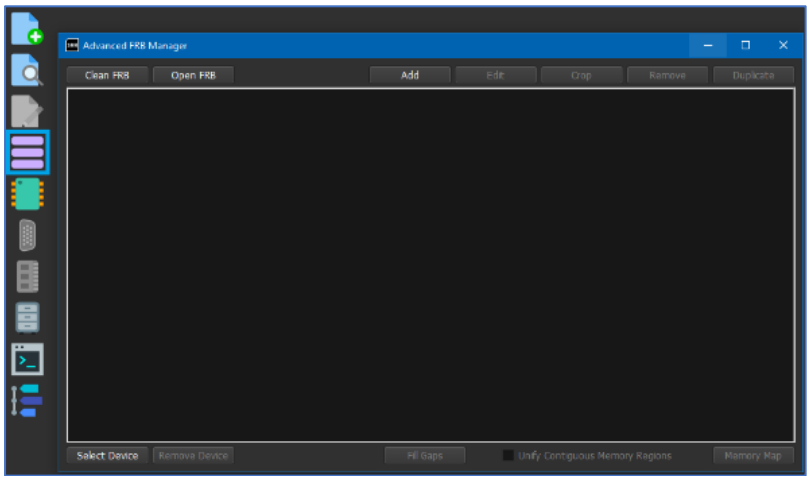
In the first line during the conversion the comments will not be added inside the VME file. This is useful if you want to maximize performance since you remove all comments, but then you cannot control the programming steps when programming the CPLD using the SMH driver. If you use the second line instead, all the comments are added but the SMH CPLD driver only prints the comments starting with "**!->**".

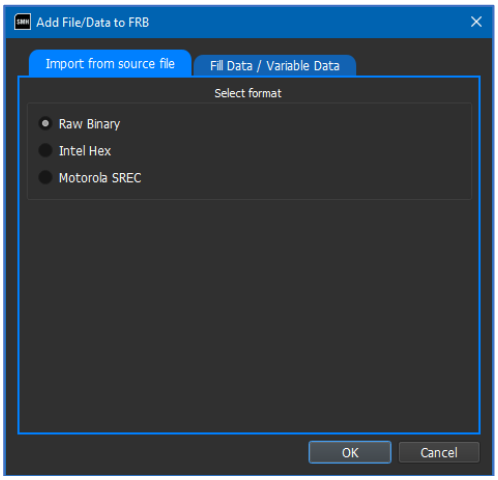
In this specific case we use the following syntax:

svf2vme12.exe -infile JED_FILE_CPLD.svf -outfile JED_FILE_CPLD.vme -comment



Now we can proceed to step three, the last step, so we can convert the **VME** file to **FRB** file.
To do that we need to open the **Workbench** and the **Advanced FRB Manager**. Then click to **Add** button and select **Raw Binary**.





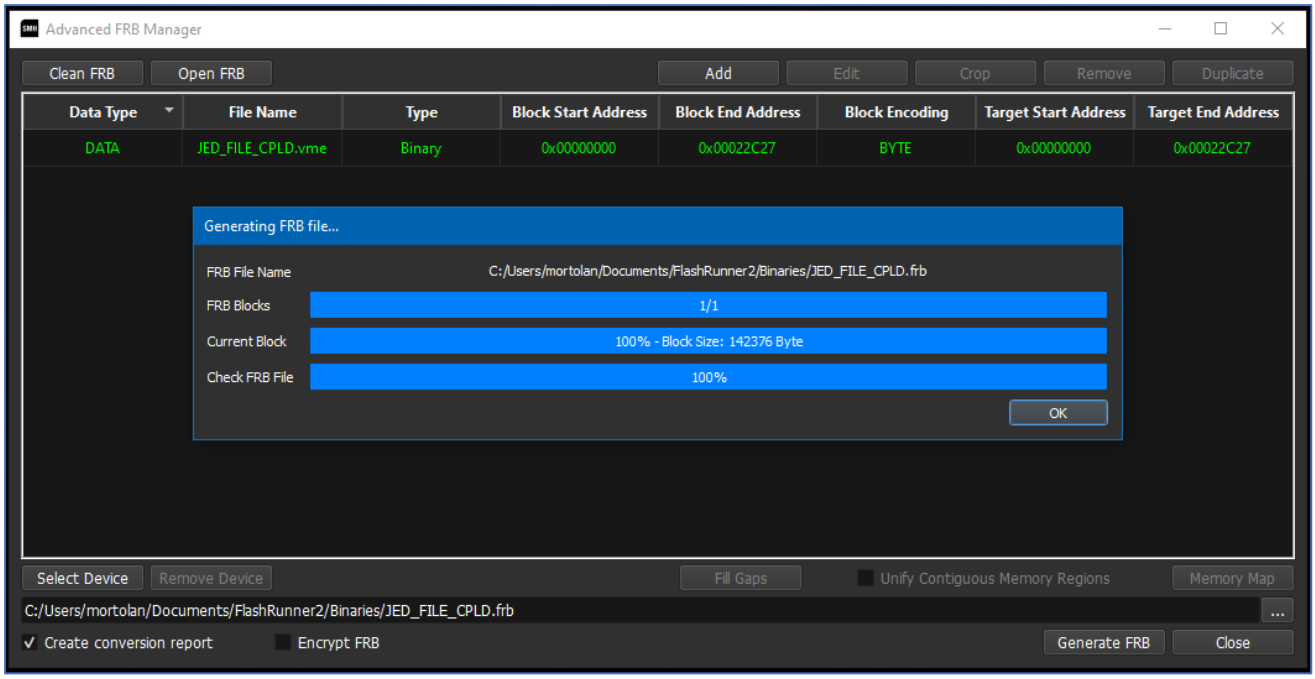
Now import your **VME** file and create the related **FRB**.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

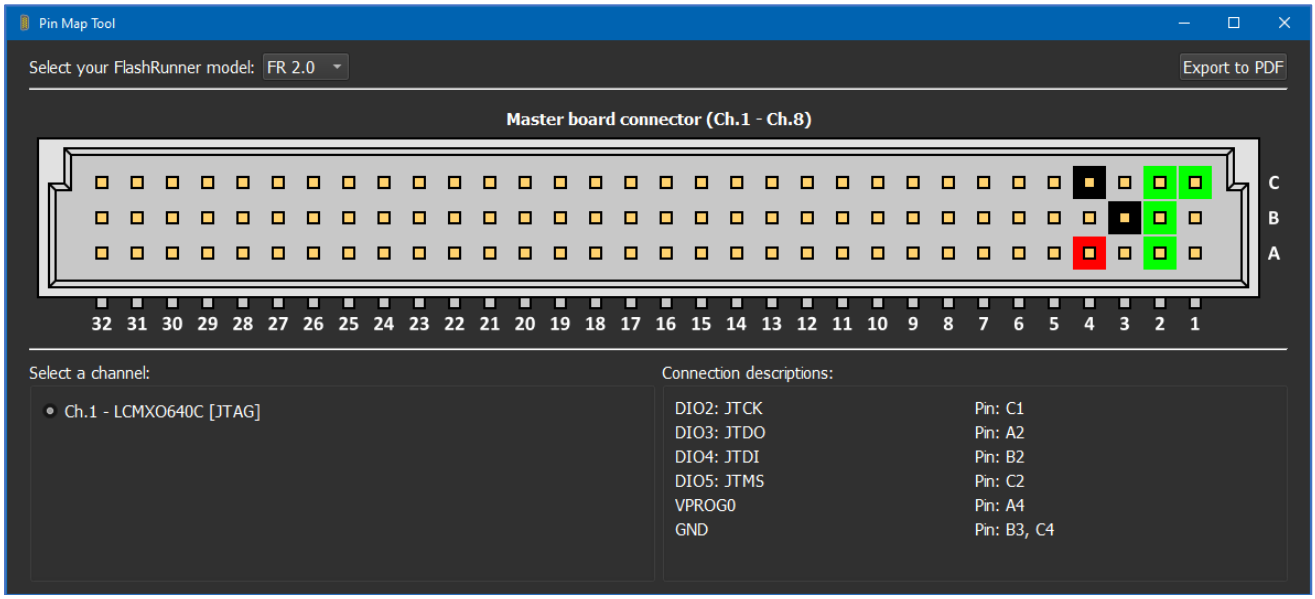


CPLD Protocol and PIN map

All CPLDs devices are programmed using JTAG protocol.

```
#TCSETPAR CMODE <JTAG>
```

CPLD PIN MAP



HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

CPLD Driver Parameters

The additional parameters are used to configure some specific options inside CPLD driver.

#TCSETPAR USE_SVF_FREQUENCY

Syntax: `#TCSETPAR USE_SVF_FREQUENCY <Use SVF Frequency>`

`<Use SVF Frequency>` Accepted parameters are Yes or No

Description: If this parameter is set to YES and if there's a protocol frequency inside the SVF file, this frequency will be used by the driver, overriding the one set by the user
If instead this parameter is set to NO or there isn't a protocol frequency in the SVF file, then the driver will use the frequency set in the project

Note: Default value is YES
Available from driver version **4.04**

#TCSETPAR SAMPLING_POINT

Syntax: `#TCSETPAR SAMPLING_POINT <Value>`

`<Value>` Accepted values are in the range 1-16

Description: Use this parameter to permanently set the sampling point of the FPGA
It is recommended to leave this parameter with the default value

Note: Default value 17

CPLD Driver Commands

#TPCMD CONNECT

`#TPCMD CONNECT`
Connect function, power on target device and entry.

#TPCMD PROGRAM

`#TPCMD PROGRAM <C>`
Processing of the entire SVF file.
From driver version **4.01** only **SVF** comments starting with "**!->**" are displayed in the Real Time Log.
From driver version **5.02** SVF comments trigger a timer so you can see into Real Time Log the elapsed time for all operations performed.

#TPCMD DISCONNECT

`#TPCMD DISCONNECT`
Disconnect function. Power off and exit.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

CPLD Driver Examples

Here you can see a complete example of CPLD projects.

1 – LCMXO640C example Commands

```
#TCSETPAR PROTCLK 10000000
#TCSETPAR PWDOWN 100
#TCSETPAR PWUP 100
#TCSETPAR USE_SVF_FREQUENCY YES
#TCSETPAR VPROG0 3300
#TCSETPAR CMODE JTAG
#TPSETSRC example.frb
#TPSTART
#TPCMD CONNECT
#TPCMD PROGRAM C
#TPCMD DISCONNECT
#TPEND
```

1 – LCMXO640C example Real Time Log

```
---#TPCMD CONNECT
Requested Clock is 10.00 MHz.
Generated Clock is 10.00 MHz.
Good samples: 11 [Range 2-12].
IDcode: 0x01285043.
Time for Connect: 0.104 s.
>|
---#TPCMD PROGRAM C
Processing Virtual Machine File (VME).
* VME Version: ____12.1.
* VME file type: compressed.
* VME Vendor: Lattice.
* SVF frequency is different from selected frequency.
  * Requested Clock is 1.00 MHz.
  * Generated Clock is 1.00 MHz.
  * Good samples: 16 [Range 0-15].
  * ID-Code Read: 0x01285043.
* [VME] - Check the IDCODE
* Elapsed time is 0.002 s.
* [VME] - Program Bscan register
* Elapsed time is 0.005 s.
* [VME] - Enable the programming mode
* Elapsed time is 0.002 s.
* [VME] - Erase the device
* Elapsed time is 1.875 s.
* [VME] - Read the status bit
* Elapsed time is 0.008 s.
* [VME] - Program Fuse Map
* Elapsed time is 0.106 s.
* [VME] - Program USERCODE
* Elapsed time is 0.959 s.
* [VME] - Read the status bit
* Elapsed time is 0.014 s.
* [VME] - Verify Fuse Map
* Elapsed time is 0.102 s.
* [VME] - Verify USERCODE
* Elapsed time is 0.536 s.
* [VME] - Program DONE bit
* Elapsed time is 0.018 s.
* [VME] - Read the status bit
* Elapsed time is 0.014 s.
* [VME] - Exit the programming mode
* Elapsed time is 0.005 s.
* [VME] - Verify SRAM DONE Bit
* Elapsed time is 0.002 s.
Closed ISPVM machine.
Time for Program C: 3.382 s.
>|
---#TPCMD DISCONNECT
>|
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

1 – LCMXO640C example Programming Times

Operation	Timings FlashRunner 2.0
Time for Connect	0.104 s
Execute SVF file	3.382 s
Cycle Time	**00:03:495 s**

2 – 5M80Z through EP3C5 example Commands

```
#TCSETPAR CUSTOMER_FIRMWARE 5M80Z.frb
#TCSETPAR FLASH_LOADER EP3C5.frb
#TCSETPAR PROTCLK 25000000
#TCSETPAR PWDOWN 100
#TCSETPAR PWUP 100
#TCSETPAR USE_SVF_FREQUENCY YES
#TCSETPAR VPROG0 3300
#TCSETPAR CMODE JTAG
#TPSTART
#TPCMD CONNECT
#TPCMD PROGRAM C
#TPCMD DISCONNECT
#TPEND
```

2 – 5M80Z through EP3C5 example Real Time Log

```
---#TPCMD CONNECT
Requested Clock is 25.00 MHz.
Generated Clock is 25.00 MHz.
Good samples: 6 [Range 4-9].
IDcode: 0x020A50DD.
Time for Connect: 0.104 s.
>|
---#TPCMD PROGRAM C
> Processing Flash Loader SVF file.
Processing Virtual Machine File (VME).
 * VME Version: ____12.1.
 * VME file type: compressed.
 * VME Vendor: Altera.
Closed ISPVM machine.
> Processing Customer Firmware SVF file.
FRB CRC32 = 0x9CE25FDB
FRB Headers collected.
Processing Virtual Machine File (VME).
* VME Version: ____12.1.
* VME file type: compressed.
* VME Vendor: Altera.
* SVF frequency is different from selected frequency.
  * Requested Clock is 20.00 MHz.
  * Generated Clock is 20.00 MHz.
  * Good samples: 8 [Range 2-9].
  * IDcode: 0x020A50DD.
Closed ISPVM machine.
Time for Program C: 17.633 s.
>|
---#TPCMD DISCONNECT
>|
```

2 – 5M80Z through EP3C5 example Programming Times

Operation	Timings FlashRunner 2.0
Time for Connect	0.104 s
Execute SVF file	17.633 s
Cycle Time	**00:17:845 s**

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

3 – M25P64 through XC6SLX75 example Commands

Program 8MiB **M25P64** serial memory through **XC6SLX75** FPGA.

```
#TCSETPAR PROTCLK 25000000
#TCSETPAR PWDOWN 100
#TCSETPAR PWUP 100
#TCSETPAR USE_SVF_FREQUENCY YES
#TCSETPAR VPROG0 3300
#TCSETPAR CMODE JTAG
#TPSTART
#TPCMD CONNECT
#TPCMD PROGRAM C
#TPCMD DISCONNECT
#TPEND
```

3 – M25P64 through XC6SLX75 example Real Time Log

```
---#TPCMD CONNECT
Requested Clock is 25.00 MHz.
Generated Clock is 25.00 MHz.
Good samples: 6 [Range 3-8].
IDcode: 0x0400E093.
Time for Connect: 0.104 s.
>|
-->TPCMD PROGRAM C
Processing Virtual Machine File (VME).
 * VME Version: VME Version: SMH 1.00.
 * VME File Type: compressed.
 * VME Vendor: Jtag Standard.
 * VME frequency is equal to inserted frequency.
 * Elapsed time is 0.001 s.
 * [VME] - Read FPGA IDcode
 * Elapsed time is 0.007 s.
 * [VME] - Download FPGA Flash Loader
 * Elapsed time is 18.298 s.
 * [VME] - M25P64 Sector erase
 * Elapsed time is 114.014 s.
 * [VME] - M25P64 Program
 * Elapsed time is 55.735 s.
 * [VME] - M25P64 Verify
 * Elapsed time is 24.070 s.
Closed ISPVM machine.
Time for Program C: 212.818 s.
>|
---#TPCMD DISCONNECT
>|
```

3 – M25P64 through XC6SLX75 example Programming Times

Operation	Timings FlashRunner 2.0
Time for Connect	0.104 s
Execute SVF file	212.818 s
Cycle Time	**03:32.980 s**

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

4 – EPM7128AE_XCV300_XC18V02 Chain example Commands

Program EPM7128AE CPLD, XCV300 FPGA and XC18V02 CPLD in chain.

```
#TCSETPAR PROTCLK 1000000
#TCSETPAR PWDOWN 100
#TCSETPAR PWUP 100
#TCSETPAR RSTDOWN 100
#TCSETPAR RSTDRV OPENDRAIN
#TCSETPAR RSTUP 100
#TCSETPAR USE_SVF_FREQUENCY YES
#TCSETPAR VPROG0 3300
#TCSETPAR CMODE JTAG
#TPSTART
#TPCMD CONNECT
#TPCMD PROGRAM C
#TPCMD DISCONNECT
#TPEND
```

4 – EPM7128AE_XCV300_XC18V02 Chain example Real Time Log

```
TPCMD CONNECT
Requested Clock is 1.00 MHz.
Generated Clock is 1.00 MHz.
Good samples: 16 [Range 0-15].
Device n.0 in chain is EPM7128AE with IDcode 0x171280DD.
Device n.1 in chain is XCV300 with IDcode 0xA0620093.
Device n.2 in chain is XC18V02 with IDcode 0xF5035093.
Time for Connect: 0.112 s.
>|
TPCMD PROGRAM C
FRB Headers collected.
Processing Virtual Machine File (VME).
 * VME Version: ____12.1.
 * VME File Type: compressed.
 * VME Vendor: Lattice.
 * VME frequency is equal to inserted frequency.
 * Elapsed time is 0.001 s.
 * [VME] - Read IDcode.
 * Elapsed time is 0.647 s.
 * [VME] - Boundary Scan Chain Contents: epm7128aet100 - xcv300_bg352 - xc18v02
 * Elapsed time is 0.001 s.
 * [VME] - Masserase
 * Elapsed time is 15.319 s.
 * [VME] - Blankcheck
 * Elapsed time is 3.215 s.
 * [VME] - Program
 * Elapsed time is 15.397 s.
 * [VME] - Verify
 * Elapsed time is 3.267 s.
Closed ISPVM machine.
Time for Program C: 38.222 s.
>|
---#TPCMD DISCONNECT
>|
```

4 – EPM7128AE_XCV300_XC18V02 Chain example Programming Times

Operation	Timings FlashRunner 2.0
Time for Connect	0.112 s
Execute SVF file	38.222 s
Cycle Time	**00:38.334 s**

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

CPLD Driver Changelog

Info about driver versions prior to 2.00
All driver versions prior to 2.00 are to be considered obsolete, please update your driver to the latest version.

Info about driver version 3.00 - 29/10/2020
Changed JTAG V FPGA to new JTAG FPGA.

Info about driver version 4.00 - 14/12/2020
Added new JTAG FPGA static 9 for FlashRunner HS.

Info about driver version 4.01 - 06/04/2021
Added new JTAG FPGA FlashRunner HS GP2 and GP4, improved SVF comment management.

Info about driver version 4.02 - 04/06/2021
Improved SVF handling in case of SVF frequency different from the selected frequency.
Added prints on the Real Time Log when the two frequencies are different.

Info about driver version 4.03 - 12/07/2021
Improved code to manage custom programming of multiple devices in the same chain.

Info about driver version 4.04 - 22/07/2021
Added TCSETPAR USE_SVF_FREQUENCY Yes/No.

Info about driver version 4.05 - 12/08/2021
Internal upgrade of the algorithm, no change to the operations it performs.

Info about driver version 4.06 - 04/11/2021
Internal update, upgraded management for JTAG FPGA reset.

Info about driver version 4.07 - 25/01/2022
Internal update and minor fixing.

Info about driver version 4.08 - 11/03/2022
JTAG FPGA management updated when FPGA is into busy state.

Info about driver version 4.09 - 16/06/2022
Print current FPGA version loaded into TPSTART command.
Upgraded internal code to align all drivers.

Info about driver version 5.00 - 31/07/2022
Added FPGA for new FlashRunner 2.0 models.

Info about driver version 5.01 - 01/09/2023
Internal driver update.

Info about driver version 5.02 - 25/01/2024
Added management of an FRB file created using an SVF file through the SMH Workbench (**SMH 1.00** version).
Added SVF frequency handling when the frequency set in the SVF file is less than 1 MHz.
Added new SVF comments management. Now comments trigger a timer so you can see into Real Time Log the elapsed time for all operations performed.
Added error handling in case of driver error. Now you can see the error stack of all functions called when the error occurs.
Updated Jtag transaction performance through better use of the Jtag FPGA.
Updated code to handle chained Jtag devices with pre or post bypass bits. Tested with three devices in the chain both in the case in which the target device is the first device in the chain and in the case in which the target device is the last in the chain.
Updated ID code print into Connect procedure.

Info about driver version 5.02 - 25/01/2024
Updated performances for CPLD driver when performing read operations.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com