

## F4FPGA

[F4FPGA documentation](#) を試してみる .

- F4FPGA 的 FPGA 開発フロー <https://f4pga.readthedocs.io/en/latest/flows/index.html#flows>
- How it works <https://f4pga.readthedocs.io/en/latest/how.html>

[Getting F4FPGA](#) を参考にセットアップしてみる

### 準備

```
$ sudo apt update
$ sudo apt install -y git wget xz-utils
$ mkdir $HOME/src/F4FPGA
$ git clone https://github.com/chipsalliance/f4pga-examples
$ cd f4pga-examples
```

### インストール

```
$ export F4PGA_INSTALL_DIR=$HOME/tools/f4pga
$ wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh -O conda_installer.sh
$ export FPGA_FAM=xc7
$ bash conda_installer.sh -u -b -p $F4PGA_INSTALL_DIR/$FPGA_FAM/conda
$ source "$F4PGA_INSTALL_DIR/$FPGA_FAM/conda/etc/profile.d/conda.sh"
$ conda env create -f $FPGA_FAM/environment.yml
```

実行しおわると ,

```
#
# To activate this environment, use
#
#   $ conda activate xc7
#
# To deactivate an active environment, use
#
#   $ conda deactivate
```

と表示されるので覚えておこう .

```
$ export F4PGA_PACKAGES='install-xc7 xc7a50t_test xc7a100t_test xc7a200t_test xc7z010_test'
```

以下の内容で , install.sh を作る .

```
mkdir -p $F4PGA_INSTALL_DIR/$FPGA_FAM
F4PGA_TIMESTAMP='20220920-124259'
F4PGA_HASH='007d1c1'
for PKG in $F4PGA_PACKAGES; do
    wget -qO-
https://storage.googleapis.com/symbiflow-arch-defs/artifacts/prod/foss-fpga-tools/symbiflow-arch-defs/continuous/insta
```

で

```
$ bash ./install.sh
```

しばらく待つとインストールがおわる。

使ってみる

インストール後にターミナルを開いたなら，

```
$ source $HOME/tools/f4pga/xc7/conda/etc/profile.d/conda.sh
```

と環境変数などを読み込んで，

```
$ conda activate xc7
```

と conda 環境に入る

セットアップで利用した f4pga-examples の下にある xc7 のサンプルディレクトリに移動。

```
$ cd f4pga-examples/xc7
```

で何かビルド。たとえば，

```
$ TARGET="arty_100" make -C counter_test
```

しばらく待つと counter\_test/build/arty\_100 の下にあれこれできる。top.bit もある。

FPGA にビットストリームを書き込むには，

```
$ TARGET="arty_100" make download -C counter_test
```

とする。パーミッションエラーになったら，セキュリティ的にアレだけど，

```
sudo chwon root $HOME/tools/f4pga/xc7/conda/envs/xc7/bin/openFPGAloader  
sudo chwon u+s $HOME/tools/f4pga/xc7/conda/envs/xc7/bin/openFPGAloader
```

とかして対応しておく。