

## Microwatt

[A tiny Open POWER ISA softcore written in VHDL 2008](#) だそうなので、とりあえず合成してみた。

```
git clone https://github.com/antonblanchard/microwatt.git
```

して fpga/topleve.vhd をトップに Vivado でプロジェクト作成。

Vivado で VHDL 2008 を解釈するにはファイルタイプを指定しないといけないので、  
tcl コンソールで

```
set_property FILE_TYPE {VHDL 2008} [get_files *.vhd]
set_property FILE_TYPE {VHDL 2008} [get_files *.vhdl]
```

とかする。

clock\_generator は IP Catalog から選んで生成。

よくわからないけど、とりあえず、合成通すためってことで、入力 125MHz 出力 100MHz でコア生成。

合成時に uPython のファームウェアを読み込むようなので、README に従って uPython をビルド。

```
git clone https://github.com/mikey/micropython
cd micropython
git checkout powerpc
cd ports/powerpc
make -j$(nproc)
```

PowerPC 用の gcc は、<https://toolchains.bootlin.com/> からダウンロード。

uPython は firmware.elf と firmware.bin ができるので、

とりあえず firmware.bin を hexdump で HEX 文字列に変換。

xdc は、exStickGE 用に、

```
set_property -dict {PACKAGE_PIN W11 IOSTANDARD LVCMOS33} [get_ports clk]
create_clock -period 8.000 -name sys_clk_pin -waveform {0.000 4.000} -add [get_ports clk]
set_property -dict {PACKAGE_PIN D19 IOSTANDARD LVCMOS33} [get_ports reset_n]
set_property -dict {PACKAGE_PIN C14 IOSTANDARD LVCMOS33} [get_ports uart0_txd]
set_property -dict {PACKAGE_PIN C15 IOSTANDARD LVCMOS33} [get_ports uart0_rxd]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCC0 [current_design]
```

と用意。

合成してみたら 100MHz だとタイミングメットせず。残念。

```
Worst Negative Slack: -2.049ns
```

らしい。リソースは、LUT 15692, LUTRAM 91, FF 4643, BRAM 256, DSP 16 とか。

## KWS 聴講

KWS@ 北千住で中原先生の招待講演きいてきた。

座長をおおせつかっていたので特別席でよかった。

Kは軽井沢じゃなくて、回路になってしまって久しいのだから、とか。

最近は姿勢推定のアプリが人気らしい

- Z. Cao, T.Simon, S.-E. Wei and Y.Sheikh, Realtime Multi-Person 3D Pose Estimation using Part Affinity Fields
- OpenPose, VGG19 + 6 Stages , [https://motchallenge.net/data/2D\\_MOT\\_2015](https://motchallenge.net/data/2D_MOT_2015)

クラウドと組み込みの違い

- FPGA Acceleration in the era of high level design, J.Freeman
  - cloud
    - many classes(1000s)
    - large workloads
    - high efficiency (perf./W)
    - server form factor
  - Embedded
    - Few classes(< 10)
    - Realtime frame rates
    - Low cost & low power
    - Custom form factor

リアルタイムもあれこれ

Ship classification from SAR image based on deep learning

[https://link.springer.com/chapter/10.1007/978-3-030-01054-6\\_2](https://link.springer.com/chapter/10.1007/978-3-030-01054-6_2)

プラットフォーム

- RPi - Jetson Nano - FPGA(Ultra96) - EdgeTPU
- 柔軟性 -> R&D コスト
- 柔軟な設計法

推論アクセラレータ

- CNN 90%
- 産総研 片岡さん マシンラーニング ノック
- Tutorial on Hardware Architectures for Deep Neural Networks
  - <http://eyeriss.mit.edu/tutorial.html>
  - <https://www.jics.utk.edu/files/images/csurre-reu/2015/Tutorial/DNN/DNN-Intro-2.pdf>

Light-weight approach

- fully pipelining vs. stage pipelining(= GPU)
- 混合精度
- スパース化
- Separable Convolution

- Point-wise + Depth-wise
- Shift-net, Shuffle-net
- 初段を Ternary でスピードも精度もあがる
  - 精度もあがる, のか .

probabilistics approach - 統計的手法

- PCA で可視化 ( 多次元を二次元にマッピング )
- ノイズくわえる
  - NCNN

Systematic Approach

- 入力画像と精度のトレードオフ
  - たとえば, OpenPose で 368x368 -> 128x128
- タイル分割
  - 前処理・後処理
- Syntegy: Algorithm-hardware co-design for convnet accelerators on embedded fpgas
  - <https://arxiv.org/abs/1811.08634>

次のトレンド

- 学習のアクセラレータ
  - なんか具体的にかわったの?
    - Lottery Ticket(?)
    - スパースありきで学習してもいい
- DA 的には設計デザインチョイス