

lowRISC/ibex

で遊んでみる

```
git clone https://github.com/lowRISC/ibex.git
sudo apt install srecord
sudo pip install fusesoc
sudo pip3 install -r python-requirements.txt
export PATH= /tools/riscv-gnu-toolchain/bin:$PATH
source /tools/Xilinx/Vivado/2020.2/settings64.sh
make build-arty-100 program-arty
```

で、ビルドして FPGA へのプログラムができる。

examples/fpga/artya7/rtl/top_artya7.sv の MEM_SIZE を

```
parameter int MEM_SIZE = 128 * 1024; // 256 kB
```

と変更すると

```
make build-arty-100 program-arty
```

と、Arty-35 向けにもビルドできる。

ビルドすると、

```
examples/sw/led
```

が BRAM に書かれた bit ファイルが生成されているよう。

ボード上の LD[3:0] が、0101(=0x0a) と 1010(~0x0a) の交互に点灯する。

合成オプションとしては、

```
-generic {SRAMInitFile ../../../../../../examples/sw/led/led.vmem}
```

というのが Vivado の Genral に Language Options で与えられている。

LED どうやって点灯してるのかと思えば、メモリへの書き込みデータを表示してるだけなのね。

```
// Connect the LED output to the lower four bits of the most significant
// byte
logic [3:0] leds;
always_ff @(posedge clk_sys or negedge rst_sys_n) begin
  if (!rst_sys_n) begin
    leds <= 4'b0;
  end else begin
    if (data_req && data_we) begin
      for (int i = 0; i < 4; i = i + 1) begin
        if (data_be[i] == 1'b1) begin
          leds <= data_wdata[i*8 +: 4];
        end
      end
    end
  end
end
assign LED = leds;
```

なるほど、面白い。