Deploy `mtkCPU` application on hardware in just 5 minutes

Install the dependencies

```
git clone https://github.com/bieganski/mtkcpu
cd mtkcpu
./install_toolchain.sh # installs riscv-none-embed-gcc
pip3 install . # installs mtkCPU and it's dependencies
```

Generate BSP files

```
./mtkcpu/cli/top.py gen_bsp
```

Expected output:

```
mateusz@mateusz:~/github/mtkcpu$ ./mtkcpu/cli/top.py gen_bsp
sw_bsp_path = ./mtkcpu/cli/../../sw/bsp
GPIO: adding output (sig led_r_0__o) to GPIO pin 0..
GPIO: adding output (sig led_g_0__o) to GPIO pin 1..
starting bsp code generation inside /home/mateusz/github/mtkcpu/sw/bsp
directory..
ok, /home/mateusz/github/mtkcpu/sw/bsp/periph_baseaddr.h file generated!
found 3 peripherials, of whom 2 is bsp-generatable..
generating /home/mateusz/github/mtkcpu/sw/bsp/uart.cc
generating /home/mateusz/github/mtkcpu/sw/bsp/uart.h
generating /home/mateusz/github/mtkcpu/sw/bsp/gpio.cc
generating /home/mateusz/github/mtkcpu/sw/bsp/gpio.h
ok, code generation done!
```

Generate linker script

First, you need linker script to be generated for your CPU configuration

```
./mtkcpu/cli/top.py gen_linker_script
```

Expected output:

```
INFO:root:writing linker script: using 0x80000000 address..
INFO:root:OK, linker script written to mtkcpu/sw/common/linker.ld file!
```

Compile software project

```
PROJ_NAME=blink_led
which riscv-none-embed-gcc # make sure it's already in your PATH (you
downloaded and extracted it in previous step).
cd sw/$PROJ_NAME
make # will generate .elf file
file build/$PROJ_NAME.elf # make sure it exists
```

Generate bitstream

Dependencies

For `Ubuntu 22.04` and newer:

```
sudo apt-get install yosys nextpnr-ice40 fpga-icestorm
```

For different distros follow the instructions in project references.

Bitstream generation

```
PROJ_NAME=blink_led
./mtkcpu/cli/top.py build -e sw/$PROJ_NAME/build/$PROJ_NAME.elf
```

The loadable content of provieded ELF file will be loaded into Block RAM memory.

After 1-2 minutes (for `iCE40` platform) you end up with `build/` directory with artifacts created and some build statistics printed:

```
INFO:./mtkcpu/cli/top.py:OK, Design was built successfully, printing out
some stats..
Info: Max frequency for clock 'cd_sync_clk12_0__i': 12.41 MHz (PASS at
12.00 MHz)

Info: Device utilisation:
Info:            ICESTORM_LC:  3279/ 5280    62%
Info:           ICESTORM_RAM:     6/   30    20%
Info:                  SB_IO:    13/   96    13%
Info:                  SB_GB:     5/    8    62%
Info:           ICESTORM_PLL:     0/    1     0%
Info:            SB_WARMBOOT:     0/    1     0%
Info:           ICESTORM_DSP:     0/    8     0%
Info:         ICESTORM_HFOSC:     0/    1     0%
Info:         ICESTORM_LFOSC:     0/    1     0%
```

```
Info:                 SB_I2C:     0/    2     0%
Info:                 SB_SPI:     0/    2     0%
Info:                 IO_I3C:     0/    2     0%
Info:            SB_LEDDA_IP:     0/    1     0%
Info:            SB_RGBA_DRV:     0/    1     0%
Info:         ICESTORM_SPRAM:     0/    4     0%
```

If you run the command above with additional `--program` param, it will program your board after build succeeded.

And this is it, your board is blinking happily!