Minimal OpenRISC System on Chip

Synthesis Examples

Note: different constraint files for different boards → inside of backend directory

1. **Spartan 3A DSP 1800**
   a) minsoc/rtl/verilog/minsoc_defines.v
      - no definitions change, ready to go

   b) minsoc/rtl/verilog/or1200/rtl/verilog/or1200_defines.v (optional, reduce logic use)
      - uncomment `define OR1200_XILINX_RAMB16
2. **Spartan 3E Starter Kit no Ethernet**
   a) minsoc/rtl/verilog/minsoc_defines.v
      - comment `define SPARTAN3A
      - uncomment `define SPARTAN3E
      - change CLOCK_DIVISOR from 5 to 2
      - comment `define ETHERNET

   b) minsoc/rtl/verilog/or1200/rtl/verilog/or1200_defines.v
      - uncomment `define OR1200_XILINX_RAMB16
      - uncomment `define OR1200_NO_DC
      - uncomment `define OR1200_NO_IC
      - uncomment `define OR1200_NO_DMMU
      - uncomment `define OR1200_NO_IMMU
3. **Spartan 3E Starter Kit with Ethernet**
   a) Synthesis properties:
      - Optimization Goal: Area
      - Optimization Effort: High

   b) minsoc/rtl/verilog/minsoc_defines.v
      - comment `define SPARTAN3A
      - uncomment `define SPARTAN3E
      - let CLOCK_DIVISOR at 5
      - change MEMORY_ADR_WIDTH from 13 to 12
      - uncomment `define ETHERNET

- comment `define UART
  - this is not necessary, though you will get 99% device usage if not commenting, 89% otherwise.

c) minsoc/rtl/verilog/or1200/rtl/verilog/or1200_defines.v

- uncomment `define OR1200_XILINX_RAMB16
- uncomment `define OR1200_NO_DC
- uncomment `define OR1200_NO_IC
- uncomment `define OR1200_NO_DMMU
- uncomment `define OR1200_NO_IMMU
- comment `define OR1200_MULT_IMPLEMENTED
- comment `define OR1200_MAC_IMPLEMENTED
- comment `define OR1200_PM_IMPLEMENTED
- comment `define OR1200_CFGR_IMPLEMENTED

d) minsoc/rtl/verilog/ethmac/rtl/verilog/eth_defines.v

- uncomment `define ETH_FIFO_XILINX
- uncomment `define ETH_XILINX_RAMB4

e) Collateral effects:

- from sw/support/Makefile.inc line 7:
  - GCC_OPT=-mhard-mul -g to GCC_OPT=-msoft-mul -g
- change sw/support/orp.ld:
  - ram: LENGTH = from 0x00006E00 to 0x00002E00
  - this is not much memory, I recommend the inclusion of the wb_ddr project to minsoc to use your DDR SRAM memory
- change sw/support/board.h
  - IN_CLK to 10000000 //(10MHz) this will make the simulation have problems with the uart output but will work on implementation
  - STACK_SIZE to 0x00180
  - UART_BAUD_RATE to 9600 //baudrate 115200 leads to a high baudrate skew due to a truncation. PC cannot recognize the output
- reduce sw/eth.c: **deprecated, valid until version 35, you can downgrade to proceed as this**
  - remove lines 230-231
  - remove lines 215-220
  - remove line 206
  - remove line 202

- change uart_print_long to uart_print_short, line 162
- change lines 98 and 99 to char tx_data[64] and char rx_data[64]
- remove lines 53-70 void uart_print_long(unsigned int ul) {}
- remove lines 31-42 void uart_interrupt(){}

f) Further area optimization possibilities: **(not necessary, DON'T DO)**

- ➢ Turn off: pic, tick timer or debug unit

4. Altera Devices

a) minsoc/rtl/verilog/minsoc_defines.v

- ➢ uncoment `define ALTERA_FPGA
- ➢ comment `define XILINX_FPGA
- ➢ comment `define SPARTAN3A
- ➢ select your memory amount "`define MEMORY_ADR_WIDTH   13"
- ➢ choose a clock division for your global clock related to your design max speed by changing the definition: "`define CLOCK_DIVISOR 5". Since you have an Altera device please use only even numbers for the division, odd numbers are going to be rounded down.
- ➢ Define your RESET polarity uncommenting "`define POSITIVE_RESET" or "`define NEGATIVE_RESET" and commenting the other.

b) minsoc/rtl/verilog/or1200/rtl/verilog/or1200_defines.v

- ➢ uncomment `define OR1200_ALTERA_LPM
- ➢ include `define OR1200_ALTERA_LPM_XXX