

Branch: master ▾

[cpu_basic](#) / README.md

[Find file](#)

[Copy path](#)



wolfram77 readme with images

55b37b4 9 minutes ago

1 contributor

118 lines (58 sloc) 2.87 KB

[Raw](#)

[Blame](#)

[History](#)



Design of a pentium-like 32-bit CPU.

Written while studying the course Advanced Computer architecture at IIIT Hyderabad, by professor R. Govindarajulu. This is a turing-complete 32-bit CPU with data movement, branch, arithmetic, and logical instructions. It follows the instruction format of Intel x86 processors, where each instruction takes 2 register operands and an optional immediate value. Like x86, this has 16 32-bit registers, a flag register, and an instruction pointer. The memory address is made to be 16-bit for simulation purposes.

Wrote this after following:

[gardintrapp/cpu_4004](#) by Oddbjorn Norstrand

Thank you.

synthesis

Quartus Prime Standard Edition - D:/Documents/cpu_basic/cpu_basic - cpu_basic

File Edit View Project Assignments Processing Tools Window Help

Search altera.com

Project Navigator Hierarchy

Entity:Instance
Cyclone IV E: EP4CE6E22C8
abc top

Compilation Report - cpu_basic

top.vhd

IP Catalog

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use work.txt_util.all;
use work.pkg_bits.all;
use work.pkg_mem.all;
use work.pkg_cpu.all;

entity top is
port(
    clk:  in  std_logic;
    rst:  in  std_logic;
    run:  in  std_logic;
    mout: in  mem_output;
    minp: out mem_input;
    cout: out cpu_output
);
end entity top;

architecture bh of top is
signal xi: cpu_internal;
begin
p_seq: process (clk, rst, run, xi)
variable i: cpu_internal;
variable m: mem_output;
variable vw: word1;
variable vdw: dword1;
variable vd, vs, imm: word;
variable zf, cf, sf: std_logic;
begin
-- reset? clean up
if rst = '1' then
report "RESET" severity warning;
i := xi;
-- clean up
i.state := st_halted;
i.regs := (others => (others => '0'));
i.flags := (others => '0');
i.ip := (others => '0');
i.op := (others => '0');
i.rd := (others => '0');
i.rs := (others => '0');
i.imm := (others => '0');
i.addr := (others => '0');
end if;
if run = '1' then
-- execute
begin
end if;
end if;
end process;
end architecture;

```

Tasks Compilation

- Compile Design
- Analysis & Synthesis
- Fitter (Place & Route)
- Assembler (Generate programming)
- Timing Analysis
- EDA Netlist Writer
- Edit Settings
- Program Device (Open Programmer)

Messages

100% 00:01:20

Quartus Prime Standard Edition - D:/Documents/cpu_basic/cpu_basic - cpu_basic

File Edit View Project Assignments Processing Tools Window Help

Search altera.com

Project Navigator Files Compilation Report - cpu_basic top.vhd

Flow Summary

Table of Contents

Flow Status Successful - Sat Apr 25 07:54:25 2020

Quartus Prime Version 19.1.0 Build 670 09/22/2019 SJ Standard Edition

Revision Name cpu_basic

Top-level Entity Name top

Family Cyclone IV E

Device EP4CE6E22C8

Timing Models Final

Total logic elements 3,558 / 6,272 (57 %)

Total registers 715

Total pins 80 / 92 (87 %)

Total virtual pins 0

Total memory bits 0 / 276,480 (0 %)

Embedded Multiplier 9-bit elements 8 / 30 (27 %)

Total PLLs 0 / 2 (0 %)

Tasks Compilation

Task

✓ ▶ Compile Design

✓ > ▶ Analysis & Synthesis

✓ > ▶ Fitter (Place & Route)

✓ > ▶ Assembler (Generate programming)

✓ > ▶ Timing Analysis

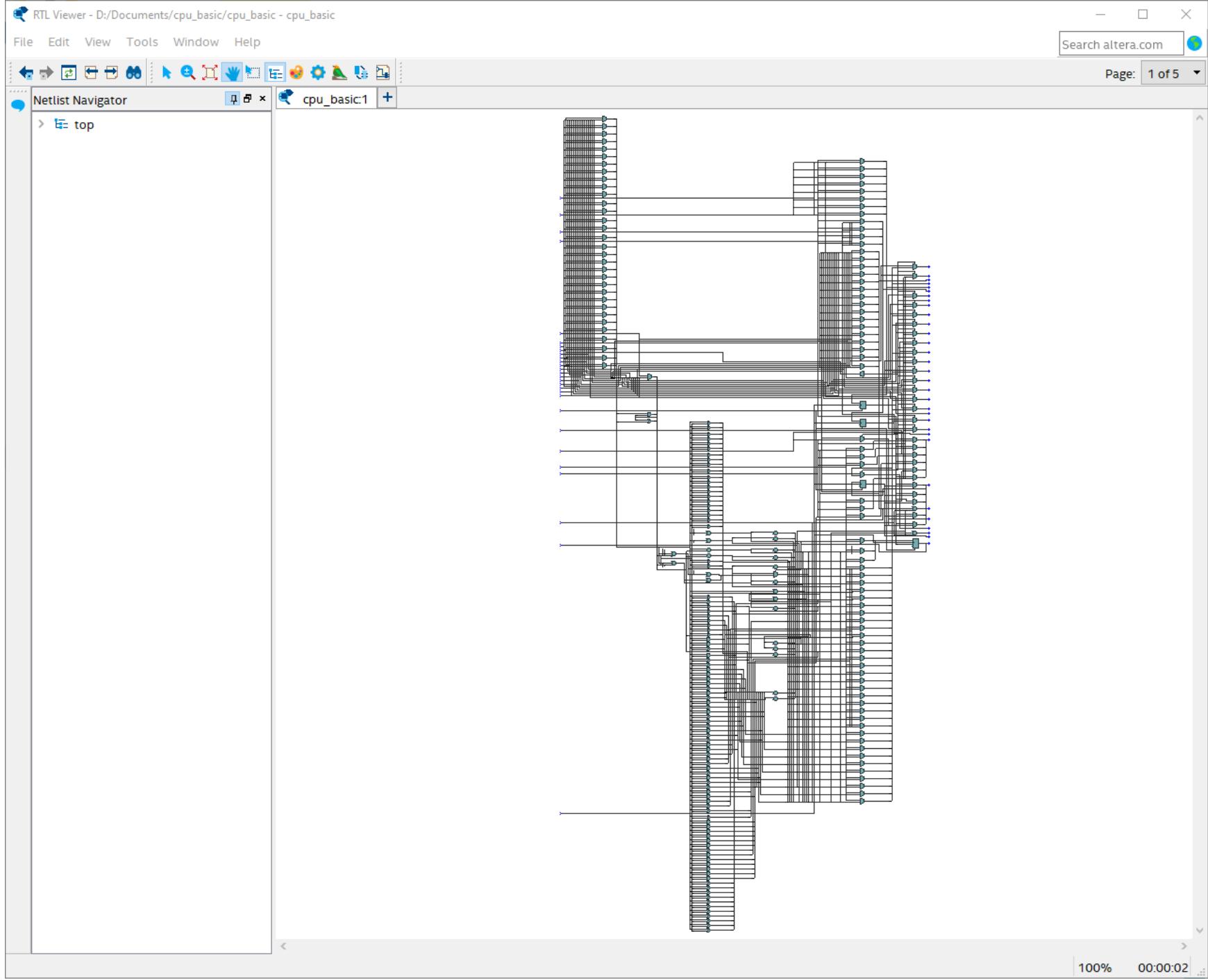
✓ > ▶ EDA Netlist Writer

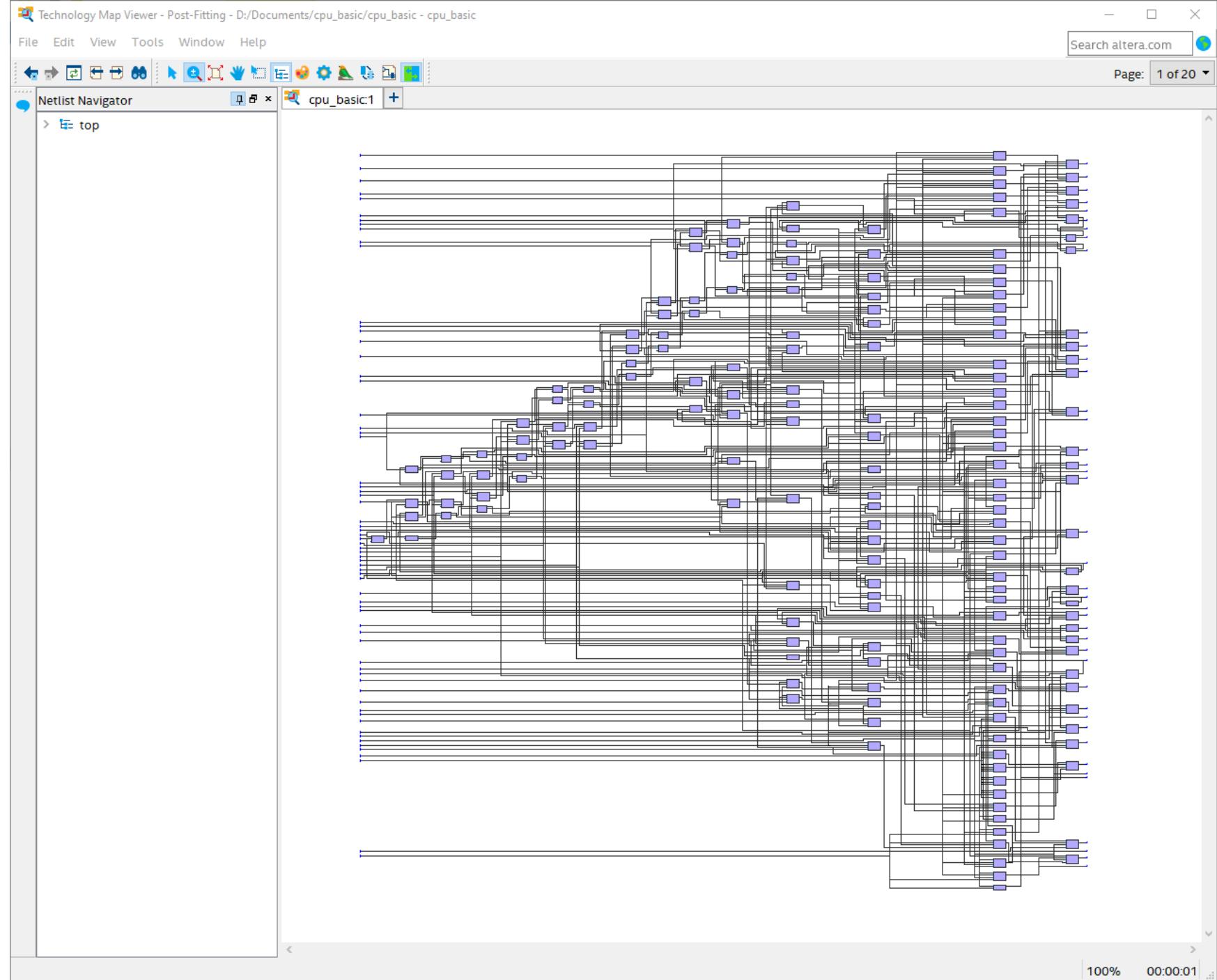
Edit Settings

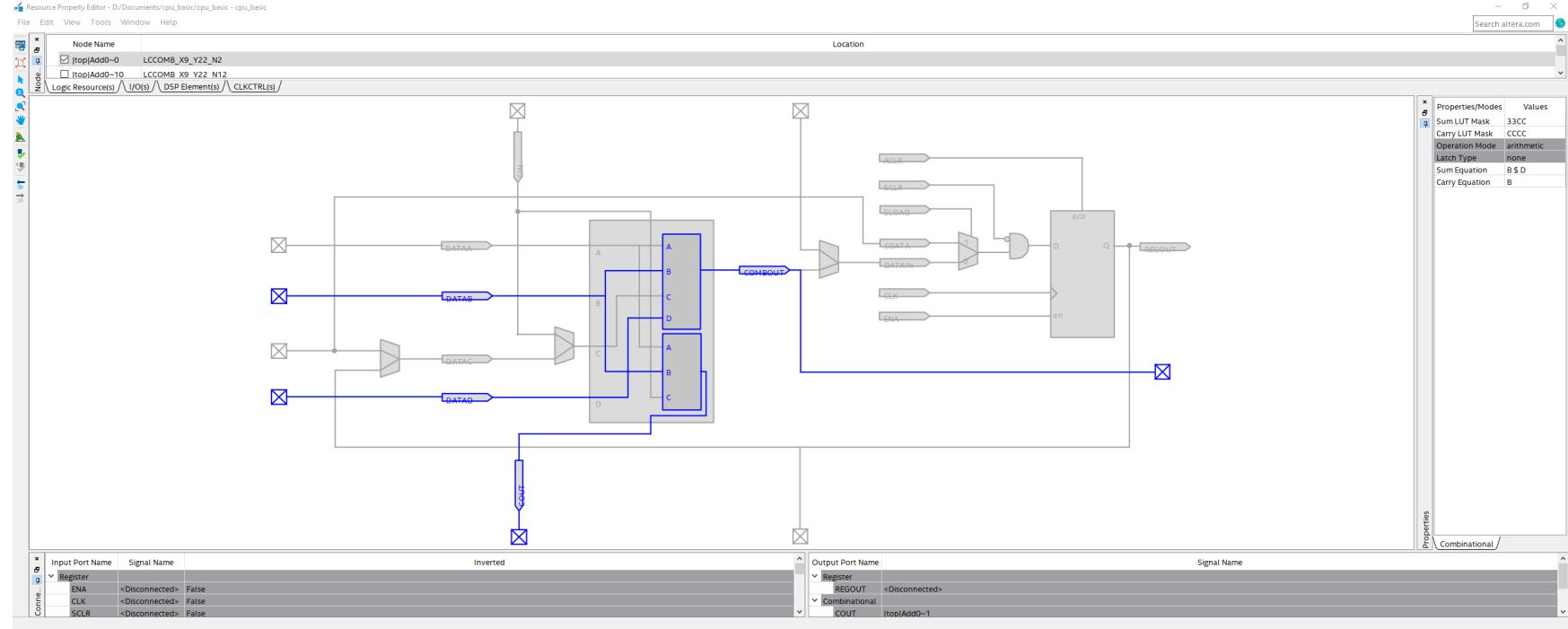
Program Device (Open Programmer)

Messages

100% 00:01:20







Pin Planner - D:/Documents/cpu_basic/cpu_basic - cpu_basic

File Edit View Processing Tools Window Help

Report

Report not available

80 / 92 pins used

(entity port)

Groups Report

Tasks

- Early Pin Planning
 - Early Pin Planning...
 - Run I/O Assignment Analy...
 - Export Pin Assignments...
 - Pin Finder...
- Highlight Pins
 - I/O Banks
 - VREF Groups
 - Edges

Named: * Edit:

All Pins

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Tristate Pres
in clk	Input			PIN_23	2.5 V (default)			8mA (default)	2 (default)		
out cout.ip[7]	Output			PIN_34	2.5 V (default)			8mA (default)	2 (default)		
out cout.ip[6]	Output			PIN_46	2.5 V (default)			8mA (default)	2 (default)		
out cout.ip[5]	Output			PIN_44	2.5 V (default)			8mA (default)	2 (default)		
out cout.ip[4]	Output			PIN_105	2.5 V (default)			8mA (default)	2 (default)		
out cout.ip[3]	Output			PIN_128	2.5 V (default)			8mA (default)	2 (default)		
out cout.ip[2]	Output			PIN_144	2.5 V (default)			8mA (default)	2 (default)		
out cout.ip[1]	Output			PIN_106	2.5 V (default)			8mA (default)	2 (default)		
out cout.ip[0]	Output			PIN_126	2.5 V (default)			8mA (default)	2 (default)		
out cout.op[7]	Output			PIN_125	2.5 V (default)			8mA (default)	2 (default)		
out cout.op[6]	Output			PIN_121	2.5 V (default)			8mA (default)	2 (default)		

0% 00:00:00

Top View

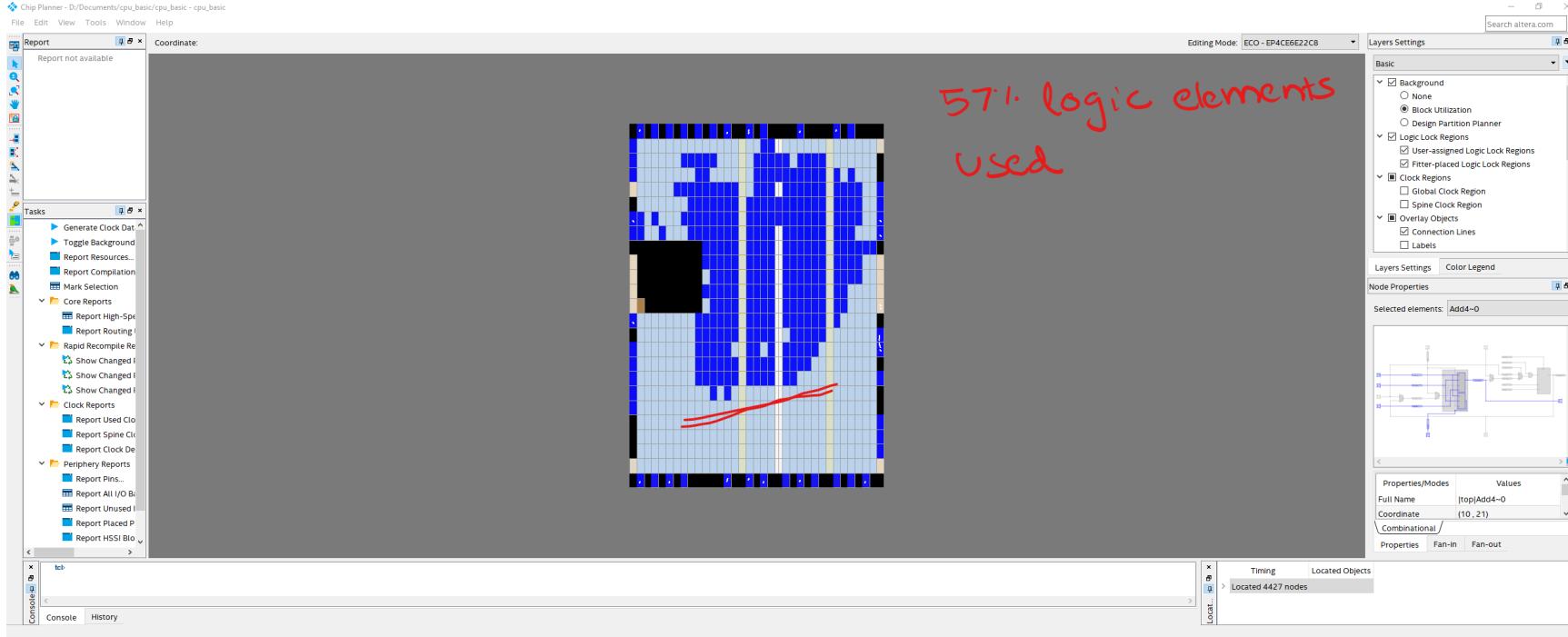
Wire Bond, with Exposed Pad

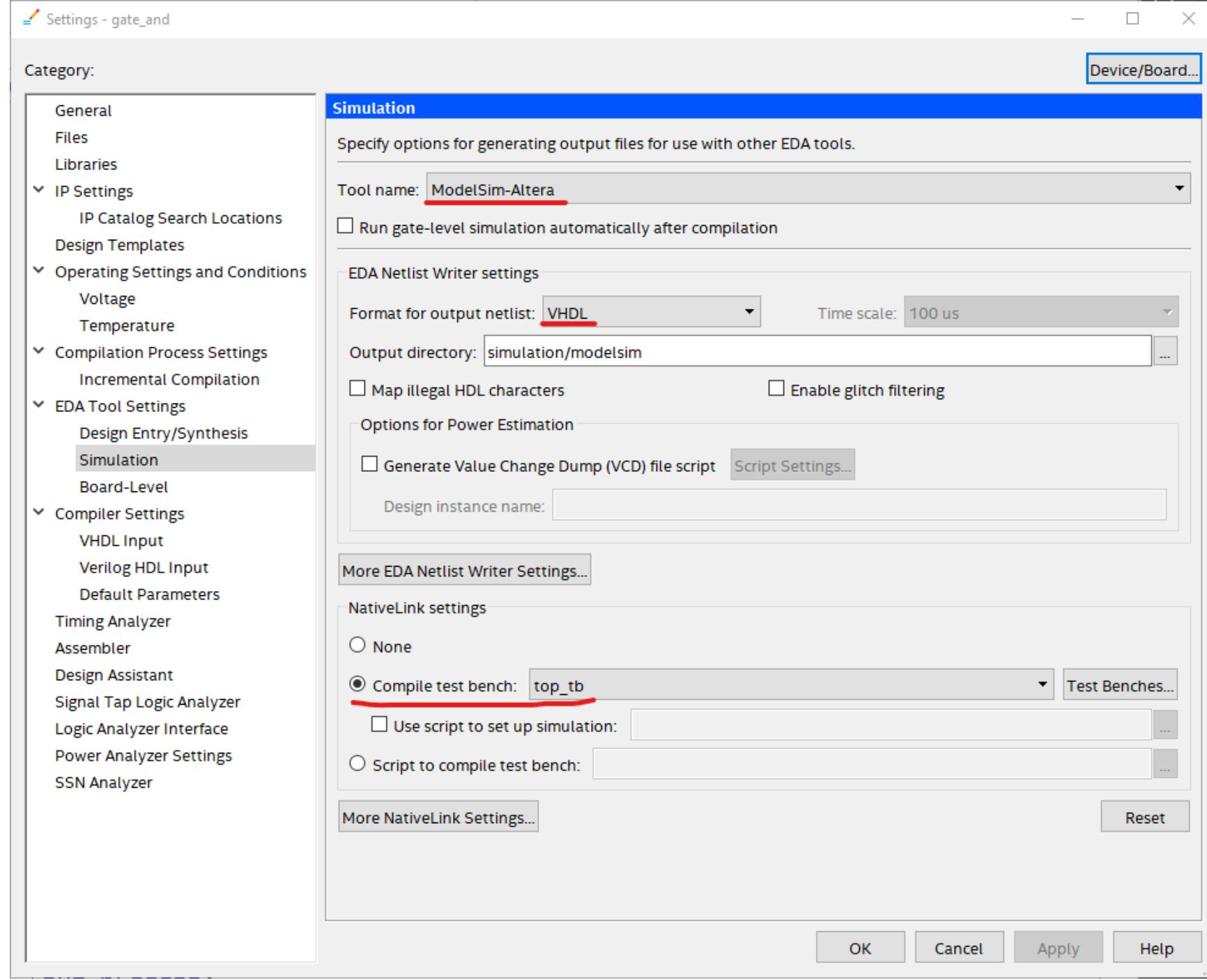
Cyclone IV E

EP4CE6E22C8

Pin Legend

Symbol	Pin Type
○	User I/O
●	User assigned...
●	Fitter assign...
●	Unbonded pad
●	Reserved pin
●	Other config...
E	DEV_OE
R	DEV_CLR
n	DIFF_n
p	DIFF_p
Q	DQ
S	DQS
L	CLK_n
J	CLK_p
L	Other PLL
D	Other dual pu...
O	MSEL0
1	MSEL1
2	MSEL2
D	CONF_DONE
E	nCE
F	nCONFIG
I	TDI
K	TCK
H	TMS





EDA Simulation Library Compiler - D:/Documents/gate_and/gate_and - gate_and

Search altera.com

Settings Messages

EDA simulation tool

Tool name: ModelSim

Executable location: D:\Program_Files\Intel\modelsim_ase\win32aloem

Current EDA simulation tool:
ModelSim ALTERA STARTER EDITION vsim 10.5b Simulator 2016.10 Oct 5 2016
Note: ModelSim-Altera software comes packaged with precompiled simulation libraries.

Compilation options

Library families

Available families: Cyclone IV GX

Selected families: Cyclone IV E

Library language

Verilog VHDL

Output

Output directory:

Show all messages Create log file

Apply settings to current project

Start Compilation Close Help

0% 00:00:00

design

File Edit Selection View Go Run Terminal Help

pkg_bits.vhd - cpu_basic - Visual Studio Code

EXPLORER

OPEN EDITORS

- top_tb.vhd M
- pkg_bits.vhd src

CPU_BASIC

- > db
- > docs
- > incremental_db
- > output_files
- > simulation
- src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - top_tb.vhd M
 - top.vhd M
 - txt_util.vhd
 - .gitignore
 - cpu_basic_nativelink_simulation.rpt
 - cpu_basic.qpf
 - cpu_basic.qsf M
 - LICENSE
 - README.md

OUTLINE

TIMELINE

pkg_bits.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

package pkg_bits is

    -- data sizes
    constant byte_values: integer := byte'high + 1;
    subtype bits4 is unsigned(3 downto 0);
    subtype bits8 is unsigned(7 downto 0);
    subtype bits16 is unsigned(15 downto 0);
    subtype bits32 is unsigned(31 downto 0);
    subtype bits33 is unsigned(32 downto 0);
    subtype bits64 is unsigned(63 downto 0);
    subtype bits65 is unsigned(64 downto 0);
    subtype word is bits32;
    subtype word1 is bits33;
    subtype dword is bits64;
    subtype dword1 is bits65;

end package pkg_bits;
```

Ln 24, Col 1 Spaces: 2 UTF-8 CRLF VHDL ⚙️ 🔍

File Edit Selection View Go Run Terminal Help

pkg_mem.vhd - cpu_basic - Visual Studio Code

EXPLORER

OPEN EDITORS

- top_tb.vhd M
- pkg_mem.vhd X

CPU_BASIC

- > db
- > docs
- > incremental_db
- > output_files
- > simulation
- src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - top_tb.vhd M
 - top.vhd M
 - txt_util.vhd
 - .gitignore
 - cpu_basic_nativelink_simulation.rpt
 - cpu_basic.qpf
 - cpu_basic.qsf M
 - LICENSE
 - README.md

OUTLINE

TIMELINE

top_tb.vhd

pkg_mem.vhd

```
src > pkg_mem.vhd
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4 use work.pkg_bits.all;
5
6
7 package pkg_mem is
8
9 -- input
10 type mem_input is record
11   addr: bits16;
12   data: bits16;
13   wr: std_logic;
14 end record mem_input;
15
16
17 -- output
18 type mem_output is record
19   data: bits16;
20 end record mem_output;
21
22
23 -- memory block
24 subtype mem_address is integer range 0 to 63;
25 type mem_block is array (mem_address) of bits8;
26
27 end package pkg_mem;
```

at addr

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF VHDL ⚙️ 🔍

File Edit Selection View Go Run Terminal Help

pkg_cpu.vhd - cpu_basic - Visual Studio Code

EXPLORER

OPEN EDITORS

- top_tb.vhd M
- pkg_cpu.vhd src
- CPU_BASIC
 - db
 - docs
 - incremental_db
 - output_files
 - simulation
- src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - top_tb.vhd M
 - top.vhd M
 - txt_util.vhd
 - .gitignore
 - cpu_basic_nativelink_simulation.rpt
 - cpu_basic.qpf
 - cpu_basic.qsf M
 - LICENSE
 - README.md

OUTLINE

TIMELINE

top_tb.vhd

pkg_cpu.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use work.pkg_bits.all;

package pkg_cpu is
    -- registers
    type cpu_registers is array (15 downto 0) of word;
    -- states
    type cpu_state is (
        st_halted,
        st_fetch0,
        st_fetch1,
        st_fetch2,
        st_fetch3,
        st_load0,
        st_load1,
        st_execute,
        st_store0,
        st_store1
    );
    -- opcodes
    constant OP_HALT: bits8 := x"00";
    constant OP_LOAD: bits8 := x"10";
    constant OP_STORE: bits8 := x"11";
    constant OP_MOVI: bits8 := x"12";
    constant OP_MOV: bits8 := x"13";
    constant OP_CMP: bits8 := x"20";
    constant OP_JMP: bits8 := x"21";
    constant OP_JZ: bits8 := x"22";
    constant OP_JNZ: bits8 := x"23";
    constant OP_JB: bits8 := x"24";
    constant OP_JBE: bits8 := x"25";
    constant OP_JG: bits8 := x"26";
    constant OP_JGE: bits8 := x"27";
    constant OP_ADD: bits8 := x"30";
    constant OP_ADC: bits8 := x"31";
    constant OP_SUB: bits8 := x"32";

```

Ln 81, Col 1 Spaces: 2 UTF-8 CRLF VHDL ⚙️ 🔍

decode while fetching

File Edit Selection View Go Run Terminal Help

pkg_cpu.vhd - cpu_basic - Visual Studio Code

EXPLORER

OPEN EDITORS

src > pkg_cpu.vhd

27
28
29
30 -- opcodes
31 constant OP_HALT: bits8 := x"00";
32
33 constant OP_LOAD: bits8 := x"10";
34 constant OP_STORE: bits8 := x"11";
35 constant OP_MOVI: bits8 := x"12";
36 constant OP_MOV: bits8 := x"13";
37
38 constant OP_CMP: bits8 := x"20";
39 constant OP_JMP: bits8 := x"21";
40 constant OP_JZ: bits8 := x"22";
41 constant OP_JNZ: bits8 := x"23";
42 constant OP_JB: bits8 := x"24";
43 constant OP_JBE: bits8 := x"25";
44 constant OP_JG: bits8 := x"26";
45 constant OP_JGE: bits8 := x"27";
46
47 constant OP_ADD: bits8 := x"30";
48 constant OP_ADC: bits8 := x"31";
49 constant OP_SUB: bits8 := x"32";
50 constant OP_SBB: bits8 := x"33";
51 constant OP_MUL: bits8 := x"34";
52 constant OP_IMUL: bits8 := x"35";
53 constant OP_DIV: bits8 := x"36";
54 constant OP_IDIV: bits8 := x"37";
55 constant OP_INC: bits8 := x"38";
56 constant OP_DEC: bits8 := x"39";
57
58 constant OP_AND: bits8 := x"40";
59 constant OP_OR: bits8 := x"41";
60 constant OP_NOT: bits8 := x"42";
61 constant OP_XOR: bits8 := x"43";
62 constant OP_SHL: bits8 := x"44";
63 constant OP SHR: bits8 := x"45";
64 constant OP_ROL: bits8 := x"46";
65 constant OP_ROR: bits8 := x"47";
66
67
68
69 -- flags
70 constant FL_CARRY: integer := 0;
71 constant FL_ZERO: integer := 1;
72 constant FL_SIGN: integer := 2;
73 constant FL_OVERFLOW: integer := 3;

move
branch
arithmetic
logical
flags

OUTLINE
TIMELINE

master* 0 △ 0 △ 0

Ln 81, Col 1 Spaces: 2 UTF-8 CRLF VHDL

File Edit Selection View Go Run Terminal Help

pkg_cpu.vhd - cpu_basic - Visual Studio Code

EXPLORER

OPEN EDITORS

- top_tb.vhd M
- pkg_cpu.vhd X
- CPU_BASIC
 - > db
 - > docs
 - > incremental_db
 - > output_files
 - > simulation
 - src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - top_tb.vhd M
 - top.vhd M
 - txt_util.vhd
 - .gitignore
 - cpu_basic_nativelink_simulation.rpt
 - cpu_basic.qpf
 - cpu_basic.qsf M
 - LICENSE
 - README.md

src > pkg_cpu.vhd

77 -- output
78 type cpu_output is record
79 -- status
80 state: bits4;
81 ip: bits8;
82 -- instruction
83 op: bits8;
84 rd: bits4;
85 rs: bits4;
86 end record cpu_output;

87
88
89 -- internal
90 type cpu_internal is record
91 -- status
92 state: cpu_state;
93 regs: cpu_registers;
94 flags: bits4;
95 ip: word;
96 -- instruction
97 op: bits8;
98 rd: bits4;
99 rs: bits4;
100 imm: word;
101 -- memory
102 addr: bits16;
103 data: bits16;
104 buff: word;
105 wr: std_logic;
106 end record cpu_internal;

107
108 function flags_word(v: word1) return bits4;
109 function flags_dword(v: dword1) return bits4;
110
111 end package pkg_cpu;

112
113
114
115
116 package body pkg_cpu is
117
118 function flags_word(v: word1) return bits4 is
119 variable f: bits4 := (others => '0');
120 begin
121 f(FL_CARRY) := v(32);
122 f(FL_SIGN) := v(31);
123 if v(31 downto 0) = 0 then f(FL_ZERO) := '1'; end if;
124 return f;
125 end function flags_word;

decoded

16 registers

memory buffer

OUTLINE

TIMELINE

master* 0 △ 0 △ 0

Ln 54, Col 14 Spaces: 2 UTF-8 CRLF VHDL ⚙ 🔍

File Edit Selection View Go Run Terminal Help top.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS

- top.vhd src M
- CPU_BASIC
 - db
 - docs
 - incremental_db
 - output_files
 - simulation
- src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - top_tb.vhd M
 - top.vhd M
 - txt_util.vhd
 - .gitignore
 - cpu_basic_nativelink_simulation.rpt
 - cpu_basic.qpf
 - cpu_basic.qsf M
 - LICENSE
 - README.md

top.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use work.txt_util.all;
use work.pkg_bits.all;
use work.pkg_mem.all;
use work.pkg_cpu.all;

entity top is
port (
    clk: in std_logic;
    rst: in std_logic;
    run: in std_logic;
    mout: in mem_output;
    minp: out mem_input;
    cout: out cpu_output
);
end entity top;

architecture bh of top is
signal xi: cpu_internal;
begin

p_seq: process (clk, rst, run, xi)
variable i: cpu_internal;
variable m: mem_output;
variable vw: word1;
variable vdw: dword1;
variable vd, vs, imm: word;
variable zf, cf, sf: std_logic;
begin

-- reset? clean up
if rst = '1' then
    report "RESET" severity warning;
    i := xi;

    -- clean up
    i.state := st_halted;
    i.regs := (others => (others => '0'));
    i.flags := (others => '0');
    i.ip := (others => '0');
    i.op := (others => '0');
    i.rd := (others => '0');
    i.rs := (others => '0');

    -- initial values
    i.pc := (others => '0');
    i.sp := (others => '0');
    i.cs := (others => '0');
    i.ss := (others => '0');
    i.a := (others => '0');
    i.d := (others => '0');
    i.w := (others => '0');
    i.m := (others => '0');
    i.f := (others => '0');
    i.zf := '0';
    i.cf := '0';
    i.sf := '0';

    -- initialize memory
    for i in 0 to 15 loop
        mem(i) := (others => '0');
    end loop;

```

← init execution

OUTLINE TIMELINE

master* 0 △ 0 △ 0 Ln 227, Col 12 Spaces: 2 UTF-8 CRLF VHDL

File Edit Selection View Go Run Terminal Help top.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS

src > top.vhd

```
cout.rs <= i.rs;
-- run? do something
elsif rising_edge(clk) then
    i := xi;
    m := mout;

case i.state is
    -- was halted? start fetch
when st_halted =>
    if run = '1' then i.state := st_fetch0; end if;

    -- request bytes 0-1
when st_fetch0 =>
    i.addr := i.ip(15 downto 0);
    i.state := st_fetch1;

    -- got bytes 0-1
    -- if short opcode, execute
    -- else request bytes 2-3
when st_fetch1 =>
    i.op := m.data(7 downto 0);
    i.rd := m.data(11 downto 8);
    i.rs := m.data(15 downto 12);
    i.addr := i.addr + 2;
    case i.op is
        when OP_LOAD |
            OP_STORE |
            OP_MOVI |
            OP JMP |
            OP_JZ |
            OP_JNZ |
            OP_JB |
            OP_JBE |
            OP_JG |
            OP_JGE =>
            i.ip := i.ip + 6;
            i.state := st_fetch2;
        when others =>
            i.ip := i.ip + 2;
            i.state := st_execute;
    end case;

    -- got bytes 2-3
    -- request bytes 4-5
when st_fetch2 =>
    i.imm(15 downto 0) := m.data;
    i.addr := i.addr + 2;
    i.state := st_fetch3;
```

run initiates the execution

instruction can be

- 6 bytes, or

- 2 bytes

File Edit Selection View Go Run Terminal Help top.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS

src > top.vhd

```
--when OP_ROL =>
-- vd := shift_left(vd, to_integer(vs));
-- ror rd, rs
--when OP_ROR =>
-- vd := shift_right(vd, to_integer(vs));
-- invalid
when others =>
    report "INVALID OPCODE" severity warning;
    i.state := st_halted;
end case;
-- i.flags(FL_CARRY) := cf;
-- i.flags(FL_SIGN) := sf;
-- i.flags(FL_ZERO) := zf;
i.regs(to_integer(i.rd)) := vd;
```

halt also invalid
opcode

update rd

```
-- invalid state
when others =>
    report "INVALID STATE" severity warning;
    i.state := st_halted;
end case;
```

new state

```
-- drive status
xi <= i;
```

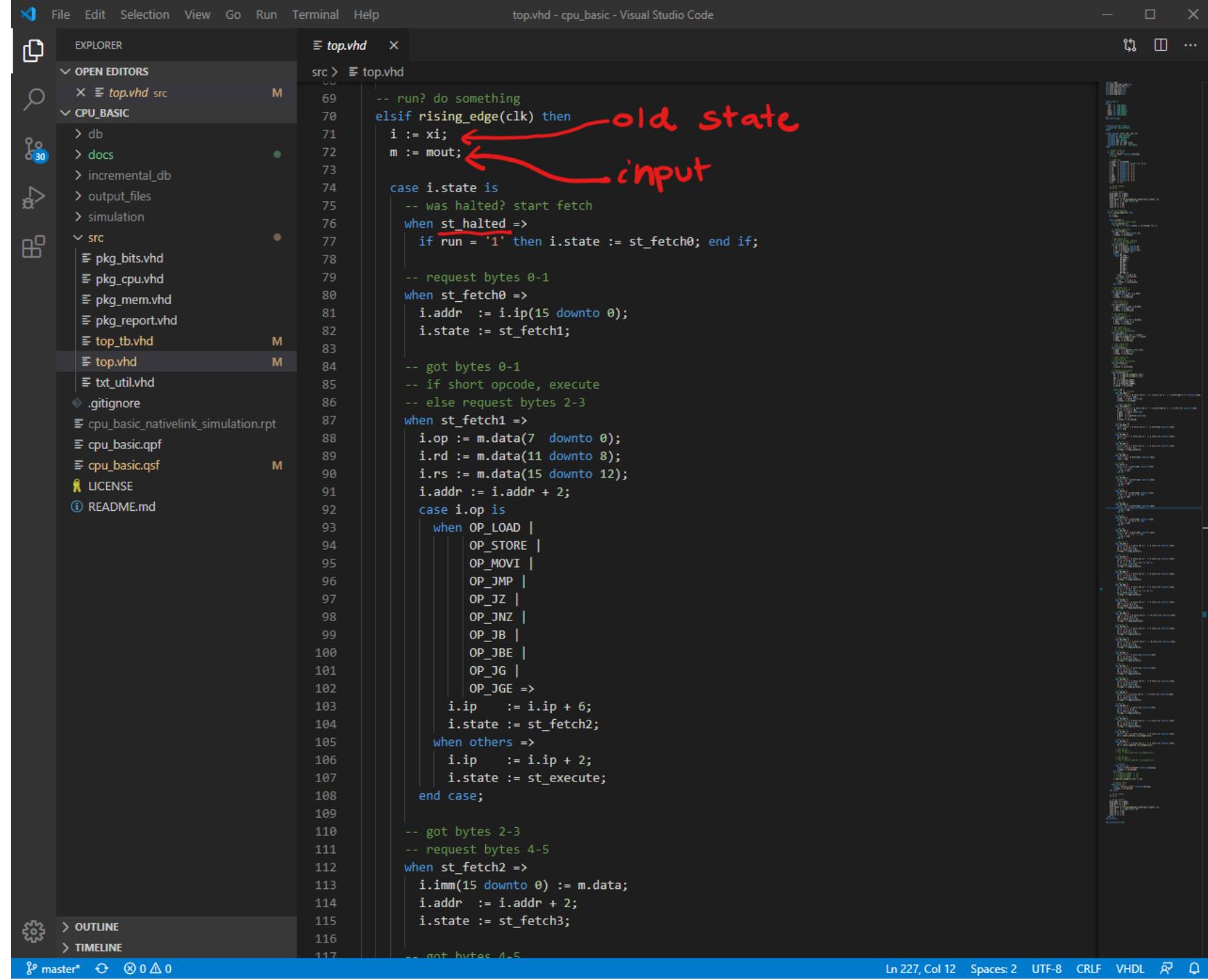
output

```
-- drive outputs
minp.addr <= i.addr;
minp.data <= i.data;
minp.wr <= i.wr;
cout.state <= to_unsigned(cpu_state'pos(i.state), 4);
cout.ip <= i.ip(7 downto 0);
cout.op <= i.op;
cout.rd <= i.rd;
cout.rs <= i.rs;
end if;
end process;
```

```
end architecture bh;
```

OUTLINE TIMELINE

master* 0 △ 0 △ 0 Ln 227, Col 12 Spaces: 2 UTF-8 CRLF VHDL



File Edit Selection View Go Run Terminal Help top.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS top.vhd

src > top.vhd

```
-- request bytes 0-1
when st_fetch0 =>
    i.addr := i.ip(15 downto 0);
    i.state := st_fetch1;

-- got bytes 0-1
-- if short opcode, execute
-- else request bytes 2-3
when st_fetch1 =>
    i.op := m.data(7 downto 0);
    i.rd := m.data(11 downto 8);
    i.rs := m.data(15 downto 12);
    i.addr := i.addr + 2;
    case i.op is
        when OP_LOAD |
            OP_STORE |
            OP_MOVI |
            OP JMP |
            OP_JZ |
            OP_JNZ |
            OP_JB |
            OP_JBE |
            OP_JG |
            OP_JGE =>
            i.ip := i.ip + 6;
            i.state := st_fetch2;
        when others =>
            i.ip := i.ip + 2;
            i.state := st_execute;
    end case;

-- got bytes 2-3
-- request bytes 4-5
when st_fetch2 =>
    i.imm(15 downto 0) := m.data;
    i.addr := i.addr + 2;
    i.state := st_fetch3;

-- got bytes 4-5
-- start execution
when st_fetch3 =>
    i.imm(31 downto 16) := m.data;
    i.addr := i.addr + 2;
    i.state := st_execute;

-- got bytes 0-1
-- request bytes 2-3
when st_load0 =>
    i.buf(15 downto 0) := m.data;
```

long instructions require 4 clock cycles

short instructions require 2 clock cycles

execute after fetching



> OUTLINE
> TIMELINE



master* ↻ ⌂ 0 △ 0

Ln 227, Col 12 Spaces: 2 UTF-8 CRLF VHDL ⌂ ⌂

File Edit Selection View Go Run Terminal Help top.vhd - CPU_BASIC - Visual Studio Code

EXPLORER OPEN EDITORS top.vhd

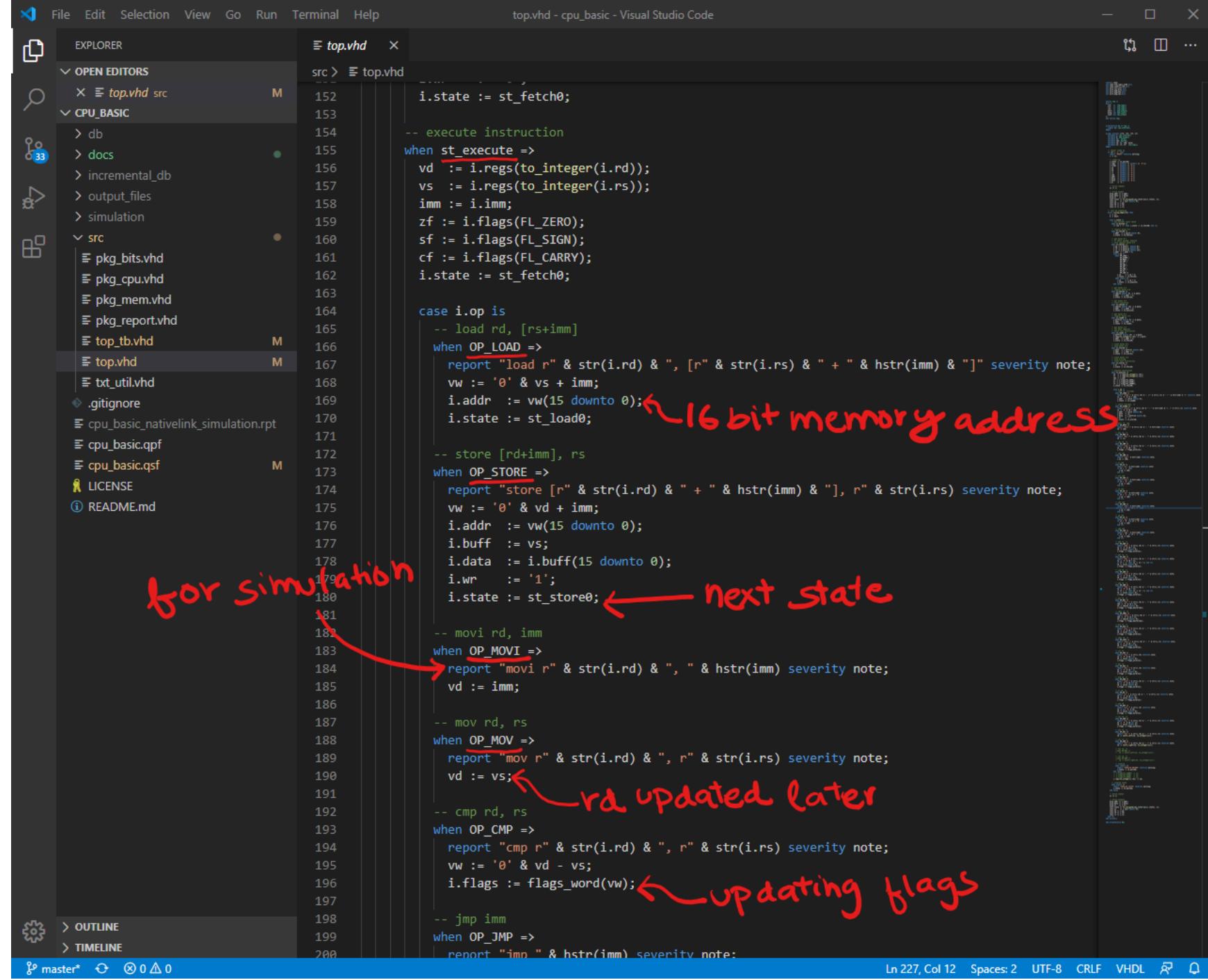
src > top.vhd

111 -- request bytes 4-5
112 when st_fetch2 =>
113 i.imm(15 downto 0) := m.data;
114 i.addr := i.addr + 2;
115 i.state := st_fetch3;
116
117 -- got bytes 4-5
118 -- start execution
119 when st_fetch3 =>
120 i.imm(31 downto 16) := m.data;
121 i.addr := i.addr + 2;
122 i.state := st_execute;
123
124 -- got bytes 0-1
125 -- request bytes 2-3
126 when st_load0 =>
127 i.buff(15 downto 0) := m.data;
128 i.addr := i.addr + 2;
129 i.state := st_load1;
130
131 -- got bytes 2-3
132 -- set dest register
133 -- fetch next instruction
134 when st_load1 =>
135 i.buff(31 downto 16) := m.data;
136 i.regs(to_integer(i.rd)) := i.buff;
137 i.addr := i.addr + 2;
138 i.state := st_fetch0;
139
140 -- wrote bytes 0-1
141 -- send bytes 2-3
142 when st_store0 =>
143 i.data := i.buff(31 downto 16);
144 i.addr := i.addr + 2;
145 i.state := st_store1;
146
147 -- wrote bytes 2-3
148 -- stop writing
149 -- fetch next instruction
150 when st_store1 =>
151 i.wr := '0';
152 i.state := st_fetch0;
153
154 -- execute instruction
155 when st_execute =>
156 vd := i.regs(to_integer(i.rd));
157 vs := i.regs(to_integer(i.rs));
158 imm := i.imm;
159 zf := i.flags(FL_ZERO);

update rd →

load and store instructions take a total of 3 clock cycles

fetch next



File Edit Selection View Go Run Terminal Help top.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS top.vhd

src > top.vhd

191
192
193
when OP_CMP =>
report "cmp r" & str(i.rd) & ", r" & str(i.rs) severity note;
vw := '0' & vd - vs;
i.flags := flags_word(vw);

194
195
196
197
198
199
when OP_JMP =>
report "jmp " & hstr(imm) severity note;
i.ip := imm;

200
201
202
203
204
205
when OP_JZ =>
report "jz " & hstr(imm) severity note;
if zf = '1' then
| i.ip := imm;
end if;

206
207
208
209
210
211
212
when OP_JNZ =>
report "jnz " & hstr(imm) severity note;
if zf = '0' then
| i.ip := imm;
end if;

213
214
215
216
217
218
when OP_JB =>
report "jb " & hstr(imm) severity note;
if sf = '1' and zf = '0' then
| i.ip := imm;
end if;

219
220
221
222
223
224
225
when OP_JBE =>
report "jbe " & hstr(imm) severity note;
if sf = '1' or zf = '1' then
| i.ip := imm;
end if;

226
227
228
229
230
231
232
when OP_JG =>
report "jg " & hstr(imm) severity note;
if sf = '0' and zf = '0' then
| i.ip := imm;
end if;

233
234
235
236
237
238
239
when OP_JGE =>
report "jge " & hstr(imm) severity note;

jump if equal
if \= if < if < if >

File Edit Selection View Go Run Terminal Help top.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS top.vhd

src > top.vhd

```
245 -- add rd, rs
246 when OP_ADD =>
247     report "add r" & str(i.rd) & ", r" & str(i.rs) severity note;
248     vw := '0' & vd + vs;
249     vd := vw(31 downto 0);
250     i.flags := flags_word(vw);
```

extra bit for carry flag

```
251
252
253
254 -- adc rd, rs
255 when OP_ADC =>
256     report "adc r" & str(i.rd) & ", r" & str(i.rs) severity note;
257     vw := '0' & vd + vs;
258     if cf = '1' then vw := vw + 1; end if;
259     vd := vw(31 downto 0);
260     i.flags := flags_word(vw);
```

use carry flag in add

```
261
262
263
264 -- sub rd, rs
265 when OP_SUB =>
266     report "sub r" & str(i.rd) & ", r" & str(i.rs) severity note;
267     vw := '0' & vd - vs;
268     vd := vw(31 downto 0);
269     i.flags := flags_word(vw);
```

```
270
271
272
273 -- sbb rd, rs
274 when OP_SBB =>
275     report "sbb r" & str(i.rd) & ", r" & str(i.rs) severity note;
276     vw := '0' & vd - vs;
277     if cf = '1' then vw := vw - 1; end if;
278     vd := vw(31 downto 0);
279     i.flags := flags_word(vw);
```

use carry flag for borrow

```
280
281
282
283 -- mul rd, rs
284 when OP_MUL =>
285     report "mul r" & str(i.rd) & ", r" & str(i.rs) severity note;
286     vdw := '0' & vd * vs;
287     vd := vdw(31 downto 0);
288     i.flags := flags_dword(vdw);
```

truncate 32bit result

```
289
290
291
292 -- imul rd, rs
293 when OP_IMUL =>
294     report "imul r" & str(i.rd) & ", r" & str(i.rs) severity note;
295     vdw := '0' & vd * vs;
296     vd := vdw(31 downto 0);
297     i.flags := flags_dword(vdw);
```

```
298
299
300
301 -- div rd, rs
302 when OP_DIV =>
303     report "div r" & str(i.rd) & ", r" & str(i.rs) severity note;
304     vw := '0' & vd / vs;
305     vd := vw(31 downto 0);
```

LN 227, Col 12 Spaces: 2 UTF-8 CRLF VHDL

File Edit Selection View Go Run Terminal Help top.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS

src > top.vhd

```
316      -- and rd, rs
317      when OP_AND =>
318          report "and r" & str(i.rd) & ", r" & str(i.rs) severity note;
319          vw := '0' & vd and vs;
320          vd := vw(31 downto 0);
321          i.flags := flags_word(vw);
322
323
324      -- or rd, rs
325      when OP_OR =>
326          report "or r" & str(i.rd) & ", r" & str(i.rs) severity note;
327          vw := '0' & vd or vs;
328          vd := vw(31 downto 0);
329          i.flags := flags_word(vw);
330
331      -- not rd
332      when OP_NOT =>
333          report "not r" & str(i.rd) severity note;
334          vw := not '0' & vd;
335          vd := vw(31 downto 0);
336          i.flags := flags_word(vw);
337
338      -- xor rd, rs
339      when OP_XOR =>
340          report "xor r" & str(i.rd) & ", r" & str(i.rs) severity note;
341          vw := '0' & vd xor vs;
342          vd := vw(31 downto 0);
343          i.flags := flags_word(vw);
344
345      -- shl rd, rs
346      when OP_SHL =>
347          report "shl r" & str(i.rd) & ", r" & str(i.rs) severity note;
348          vd := shift_left(vd, to_integer(vs));
349
350
351      -- shr rd, rs
352      when OP_SHR =>
353          report "shr r" & str(i.rd) & ", r" & str(i.rs) severity note;
354          vd := shift_right(vd, to_integer(vs));
355
356      -- rol rd, rs
357      --when OP_ROL =>
358          -- vd := shift_left(vd, to_integer(vs));
359
360      -- ror rd, rs
361      --when OP_ROR =>
362          -- vd := shift_right(vd, to_integer(vs));
363
364      -- invalid
      when others =>
```

OUTLINE TIMELINE

master* 0 △ 0 △ 0 Ln 227, Col 12 Spaces: 2 UTF-8 CRLF VHDL

testbench

File Edit Selection View Go Run Terminal Help top_tb.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS

- top_tb.vhd src M
- CPU_BASIC
 - db
 - docs
 - incremental_db
 - output_files
 - simulation
- src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - top_tb.vhd M
 - top.vhd M
 - txt_util.vhd
 - .gitignore
 - cpu_basic_nativelink_simulation.rpt
 - cpu_basic.qpf
 - cpu_basic.qsf M
 - LICENSE
 - README.md

top_tb.vhd X

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use work.txt_util.all;
use work.pkg_bits.all;
use work.pkg_mem.all;
use work.pkg_cpu.all;

entity top_tb is
end entity top_tb;

architecture bh of top_tb is
constant ID_CPU: string := "cpu";
constant ID_MEM: string := "mem";
constant tclk: time := 1 ns;

signal clk: std_logic;
signal rst: std_logic;
signal run: std_logic;
signal mout: mem_output;
signal minp: mem_input;
signal cout: cpu_output;
signal mem: mem_block;
signal mcopy: mem_block;
signal mwr: std_logic;

begin
DUT: entity work.top port map (clk, rst, run, mout, minp, cout);

-- run clock
p_clk: process
begin
    clk <= '0';
    wait for tclk / 2;

    clk <= '1';
    wait for tclk / 2;
end process;

-- 16-bit memory
p_mem: process (clk)
begin
    variable i: mem_input;
    variable o: mem_output;
    variable a: integer range 0 to 65535;
    i := minp;
```

64 byte memory block
for setting up

OUTLINE TIMELINE

master* 0 △ 0 △ 0 Ln 21, Col 20 Spaces: 2 UTF-8 CRLF VHDL

File Edit Selection View Go Run Terminal Help top_tb.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS

- top_tb.vhd src
- CPU_BASIC
 - db
 - docs
 - incremental_db
 - output_files
 - simulation
- src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - top_tb.vhd M
 - top.vhd M
 - txt_util.vhd
 - .gitignore
 - cpu_basic_nativelink_simulation.rpt
 - cpu_basic.qpf
 - cpu_basic.qsf M
 - LICENSE
 - README.md

generate clock with period 1ns

```
30
31
32 -- run clock
33 p_clk: process
34 begin
35   clk <= '0';
36   wait for tclk / 2;
37
38   clk <= '1';
39   wait for tclk / 2;
40 end process;

-- 16-bit memory
44 p_mem: process (clk)
45   variable i: mem_input;
46   variable o: mem_output;
47   variable a: integer range 0 to 65535;
48 begin
49   i := minp;
50   o := mout;
51   a := to_integer(i.addr);
52
53   if falling_edge(clk) then
54     -- output
55     o.data(7 downto 0) := mem(a);
56     o.data(15 downto 8) := mem(a + 1);
57     if mwr = '1' then
58       mem <= mcopy;
59     end if;
60     if i.wr = '1' then
61       report "write @" & str(a) & ":" & hstr(i.data) severity note;
62       mem(a)    <= i.data(7 downto 0);
63       mem(a + 1) <= i.data(15 downto 8);
64     end if;
65     mout <= o;
66   end if;
67 end process;

-- run test
71 p_init: process
72 begin
73   -- initialize
74   rst <= '1';
75   run <= '0';
76   mcopy <= (others => x"00");
77   mwr <= '1';
78   wait for tclk;
```

} read memory at addr

} write data at addr

8bit memory with 16 bit interface

Ln 21, Col 20 Spaces: 2 UTF-8 CRLF VHDL ⚙ 🔍

File Edit Selection View Go Run Terminal Help top_tb.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS

- top_tb.vhd src
- CPU_BASIC
 - db
 - docs
 - incremental_db
 - output_files
 - simulation
- src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - top_tb.vhd M
 - top.vhd M
 - txt_util.vhd
 - .gitignore
 - cpu_basic_nativelink_simulation.rpt
 - cpu_basic.qpf
 - cpu_basic.qsf M
 - LICENSE
 - README.md

top_tb.vhd X

```
src > top_tb.vhd
09
70  -- run test
71  p_init: process
72  begin
73    -- initialize
74    rst <= '1';
75    run <= '0';
76    mcopy <= (others => x"00");
77    mwr <= '1';
78    wait for tclk;
79    rst <= '0';
80    mwr <= '0';

81
82  -- test square
83  if 1 = 1 then
84    -- write program
85    mcopy <=
86      -- movi r0, 0000000A (10)
87      0 => OP_MOVI, 1 => x"00",
88      2 => x"0A", 3 => x"00", 4 => x"00", 5 => x"00",
89      -- mul r0, r0
90      6 => OP_MUL, 7 => x"00",
91      -- store [r1+00000000], r0
92      8 => OP_STORE, 9 => x"01",
93      10 => x"00", 11 => x"00", 12 => x"00", 13 => x"00",
94      -- halt
95      others => OP_HALT
96    );
97    mwr <= '1';
98    rst <= '1';
99    run <= '0';
100   wait for tclk;
101   mwr <= '0';
102   rst <= '0';

103
104  -- run
105  run <= '1';
106  wait for tclk;
107  run <= '0';
108  wait for 20 * tclk; } Start execution } write program to memory & reset CPU } result stored at mem(0) ← wait for result → 102 = 100

109
110  -- check result
111  assert (to_integer(mem(0)) = 100) report "SQUARE failed" severity failure;
112  report "SQUARE passed" severity note;
113 end if;

114
115  -- test factorial
116  if 1 = 1 then
117    -- write program
```

OUTLINE
TIMELINE

master* 0 △ 0 △ 0 Ln 21, Col 20 Spaces: 2 UTF-8 CRLF VHDL

File Edit Selection View Go Run Terminal Help top_tb.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS

src > top_tb.vhd

```
114  
115 -- test factorial  
116 if 1 = 1 then  
117     -- write program  
118     mcopy <= (  
119         -- movi r0, 00000005 ←  
120         0 => OP_MOVI, 1 => x"00",  
121         2 => x"05", 3 => x"00", 4 => x"00", 5 => x"00",  
122         -- inc r1  
123         6 => OP_INC, 7 => x"01",  
124         -- mul r1, r0  
125         8 => OP_MUL, 9 => x"01",  
126         -- dec r0  
127         10 => OP_DEC, 11 => x"00",  
128         -- cmp r0, r2  
129         12 => OP_CMP, 13 => x"20",  
130         -- jnz 00000008  
131         14 => OP_JNZ, 15 => x"00",  
132         16 => x"08", 17 => x"00", 18 => x"00", 19 => x"00",  
133         -- store [r2+00000000], r1  
134         20 => OP_STORE, 21 => x"12",  
135         22 => x"00", 23 => x"00", 24 => x"00", 25 => x"00",  
136         -- halt  
137         others => OP_HALT  
138     );  
139     mwr <= '1';  
140     rst <= '1';  
141     run <= '0';  
142     wait for tclk;  
143     mwr <= '0';  
144     rst <= '0';  
145  
146     -- run  
147     run <= '1';  
148     wait for tclk;  
149     run <= '0';  
150     wait for 100 * tclk; } start execution ← wait for complete  
151  
152     -- check result  
153     assert (to_integer(mem(0)) = 120) report "FACTORIAL failed" severity failure;  
154     report "FACTORIAL passed" severity note;  
155 end if;  
156  
157     -- test prime  
158 if 1 = 1 then  
159     -- write program  
160     mcopy <= (  
161         -- movi r0, 0000000D (13)  
162 )
```

INPUT = 5 } result stored at mem(0)

} Write program to memory & reset CPU

} Start execution

5! = 120

File Edit Selection View Go Run Terminal Help top_tb.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS

src > top_tb.vhd

156
157
158 -- test prime
159 if i = 1 then
160 -- write program
161 mcopy <= (input=13
162 -- movi r0, 000000D (13)
163 0 => OP_MOVI, 1 => x"00",
164 2 => x"00", 3 => x"00", 4 => x"00", 5 => x"00",
165 -- inc r1
166 6 => OP_INC, 7 => x"01",
167 -- inc r1
168 8 => OP_INC, 9 => x"01",
169 -- mov r2, r0
170 10 => OP_MOV, 11 => x"02",
171 -- mov r3, r0 }
172 12 => OP_MOV, 13 => x"03",
173 -- div r3, r1
174 14 => OP_DIV, 15 => x"13",
175 -- mul r3, r1
176 16 => OP_MUL, 17 => x"13",
177 -- mov r4, r0
178 18 => OP_MOV, 19 => x"04",
179 -- sub r4, r3
180 20 => OP_SUB, 21 => x"34",
181 -- jz 0000028 (40) }
182 22 => OP_JZ, 23 => x"00",
183 24 => x"28", 25 => x"00", 26 => x"00", 27 => x"00",
184 -- inc r1
185 28 => OP_INC, 29 => x"01",
186 -- cmp r2, r1
187 30 => OP_CMP, 31 => x"12",
188 -- jnz 000000C (12)
189 32 => OP_JNZ, 33 => x"00",
190 34 => x"0C", 35 => x"00", 36 => x"00", 37 => x"00",
191 -- inc r5
192 38 => OP_INC, 39 => x"05",
193 -- store [r6+00000000], r5 }
194 40 => OP_STORE, 41 => x"56",
195 42 => x"00", 43 => x"00", 44 => x"00", 45 => x"00",
196 -- halt
197 others => OP_HALT
198);
199 mwr <= '1';
200 rst <= '1';
201 run <= '0';
202 wait for tclk;
203 mwr <= '0';
204 rst <= '0';

input=13

Store result at mem(0)

OUTLINE TIMELINE

master* 0 △ 0 △ 0

Ln 21, Col 20 Spaces: 2 UTF-8 CRLF VHDL

File Edit Selection View Go Run Terminal Help top_tb.vhd - cpu_basic - Visual Studio Code

EXPLORER OPEN EDITORS

- top_tb.vhd src M
- CPU_BASIC
 - > db
 - > docs
 - > incremental_db
 - > output_files
 - > simulation
- src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - top_tb.vhd M
 - top.vhd M
 - txt_util.vhd
 - .gitignore
 - cpu_basic_nativelink_simulation.rpt
 - cpu_basic.qpf
 - cpu_basic.qsf M
 - LICENSE
 - README.md

top_tb.vhd

```
src > top_tb.vhd
175      -- mul r3, r1
176      16 => OP_MUL, 17 => x"13",
177      -- mov r4, r0
178      18 => OP_MOV, 19 => x"04",
179      -- sub r4, r3
180      20 => OP_SUB, 21 => x"34",
181      -- jz 0000028 (40)
182      22 => OP_JZ, 23 => x"00",
183      24 => x"28", 25 => x"00", 26 => x"00", 27 => x"00",
184      -- inc r1
185      28 => OP_INC, 29 => x"01",
186      -- cmp r2, r1
187      30 => OP_CMP, 31 => x"12",
188      -- jnz 000000C (12)
189      32 => OP_JNZ, 33 => x"00",
190      34 => x"0C", 35 => x"00", 36 => x"00", 37 => x"00",
191      -- inc r5
192      38 => OP_INC, 39 => x"05",
193      -- store [r6+00000000], r5
194      40 => OP_STORE, 41 => x"56",
195      42 => x"00", 43 => x"00", 44 => x"00", 45 => x"00",
196      -- halt
197      others => OP_HALT
198 );
199 mwr <= '1';
200 rst <= '1';
201 run <= '0';
202 wait for tclk;
203 mwr <= '0';
204 rst <= '0';
205
206      -- run
207      run <= '1';
208      wait for tclk;
209      run <= '0';
210      wait for 500 * tclk; } start execution
211
212      -- check result
213      assert (to_integer(mem(0)) = 1) report "PRIME failed" severity failure;
214      report "PRIME passed" severity note;
215 end if;
216 wait;
217 end process;
218 end architecture bh;
```

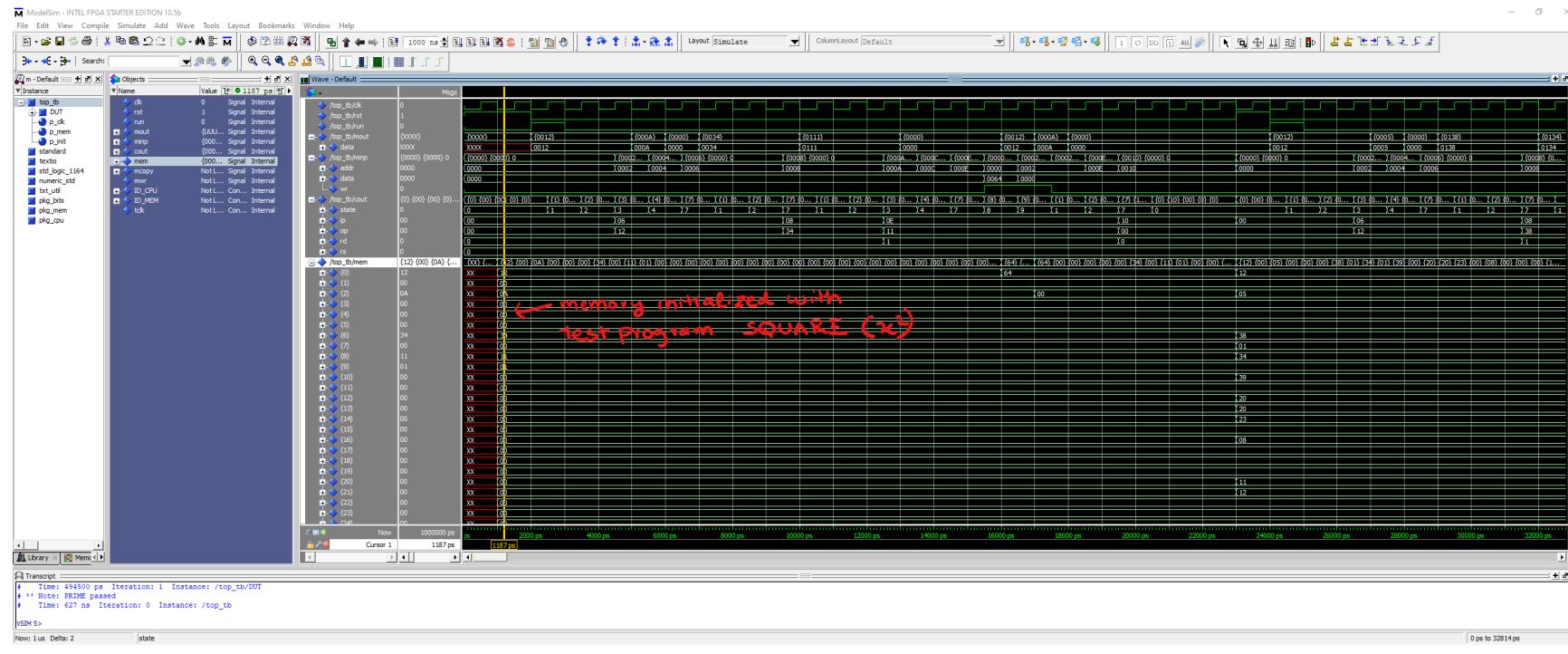
write program to memory & reset CPU

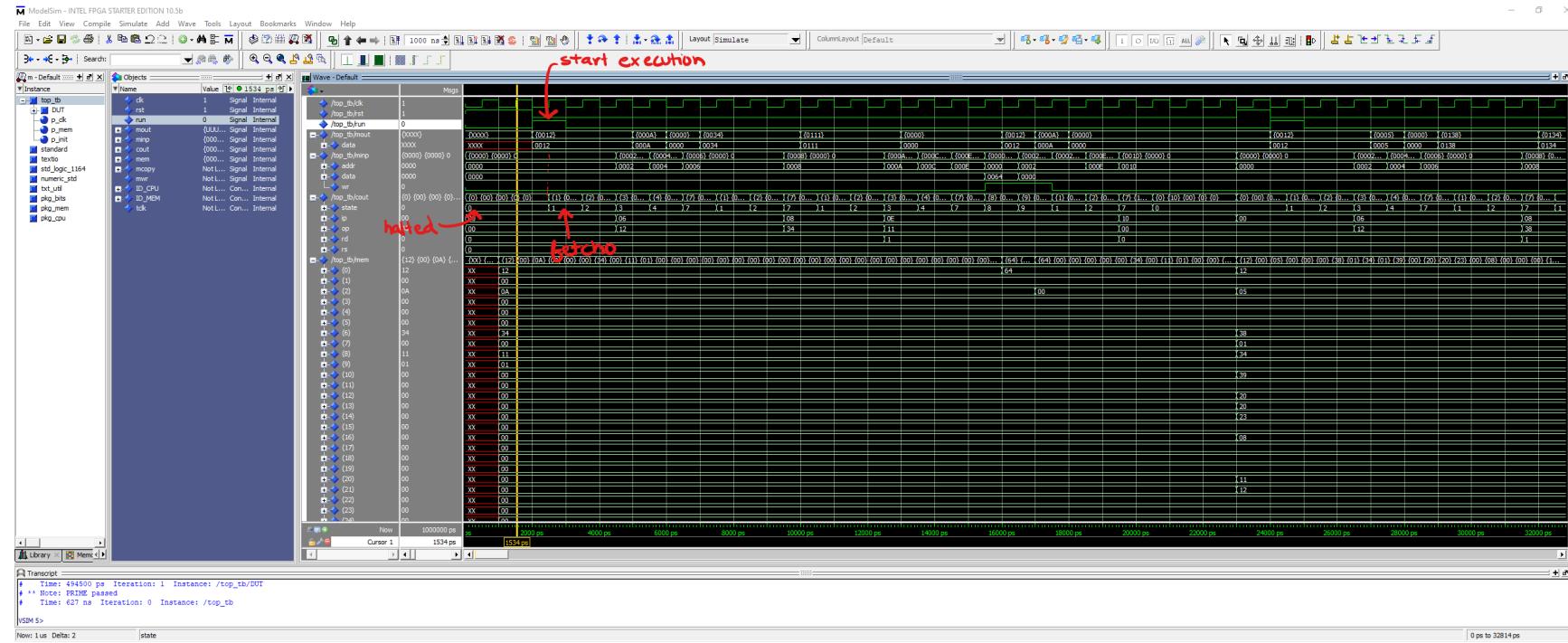
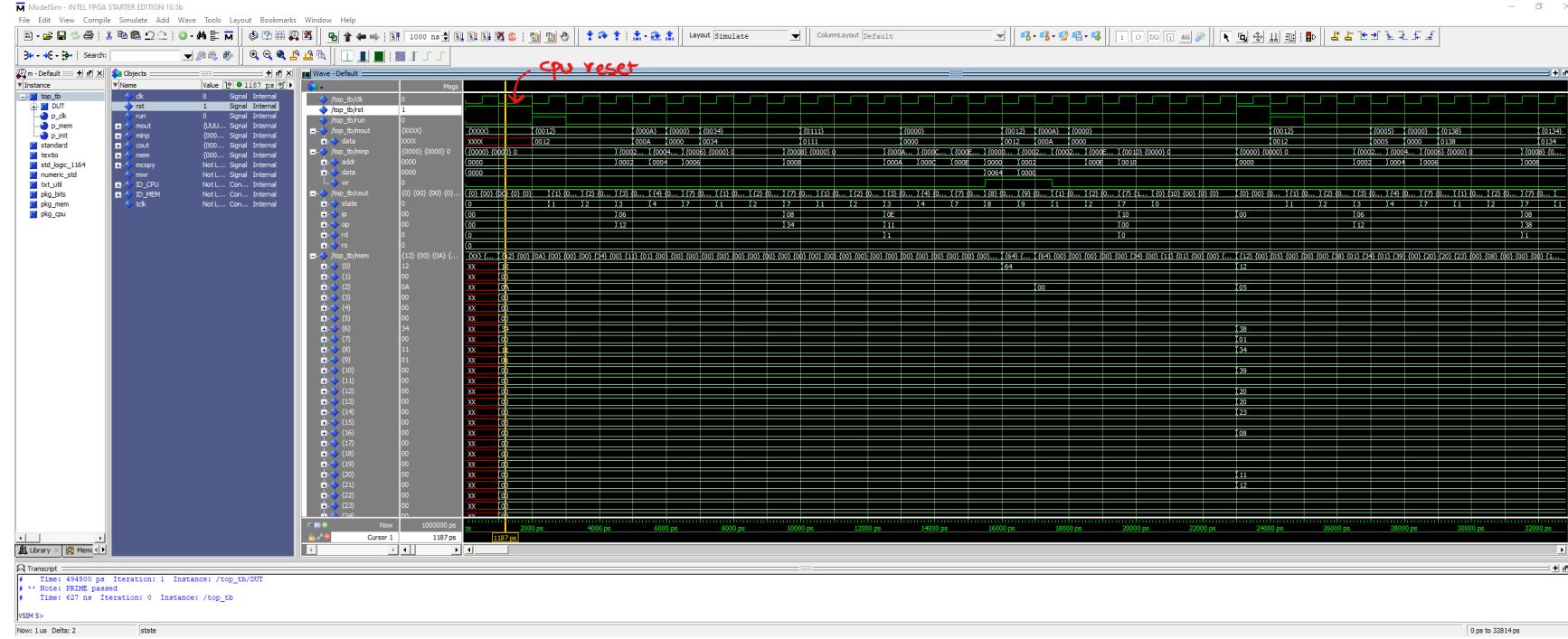
wait for result

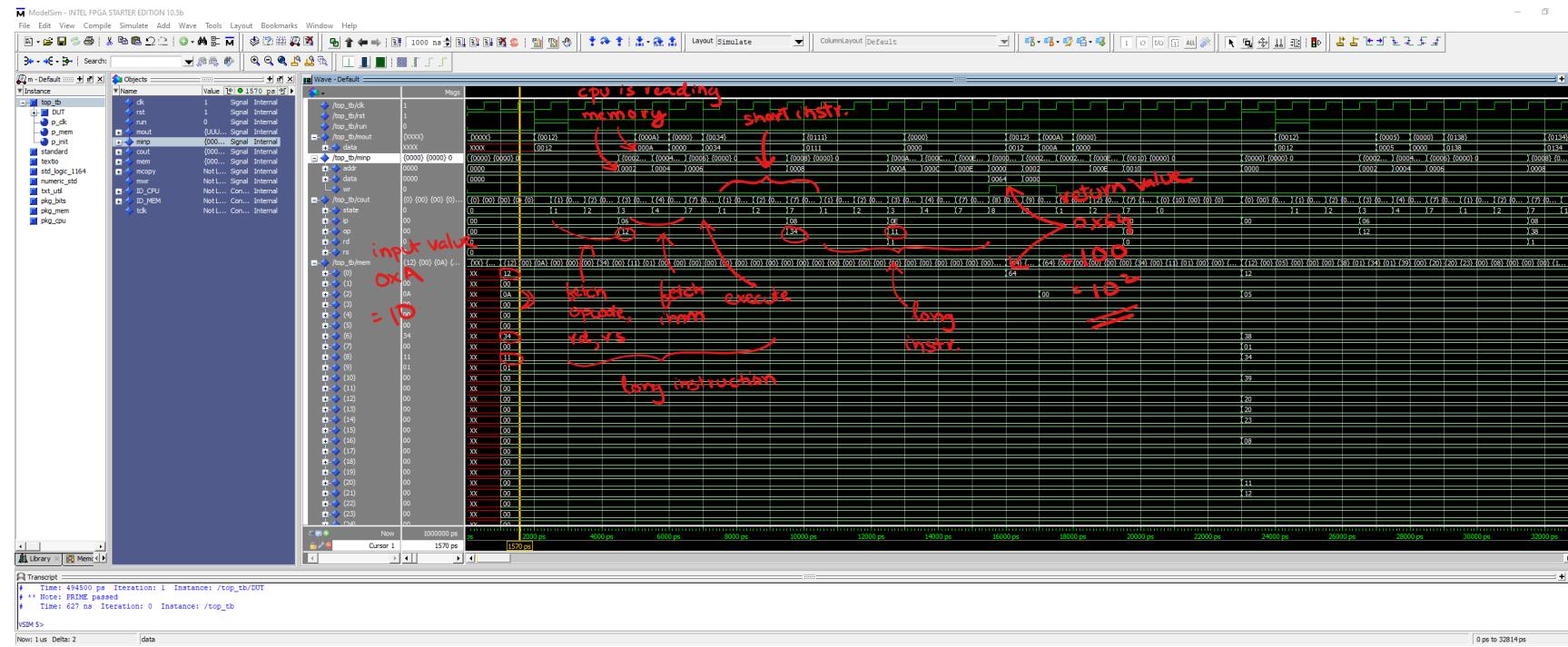
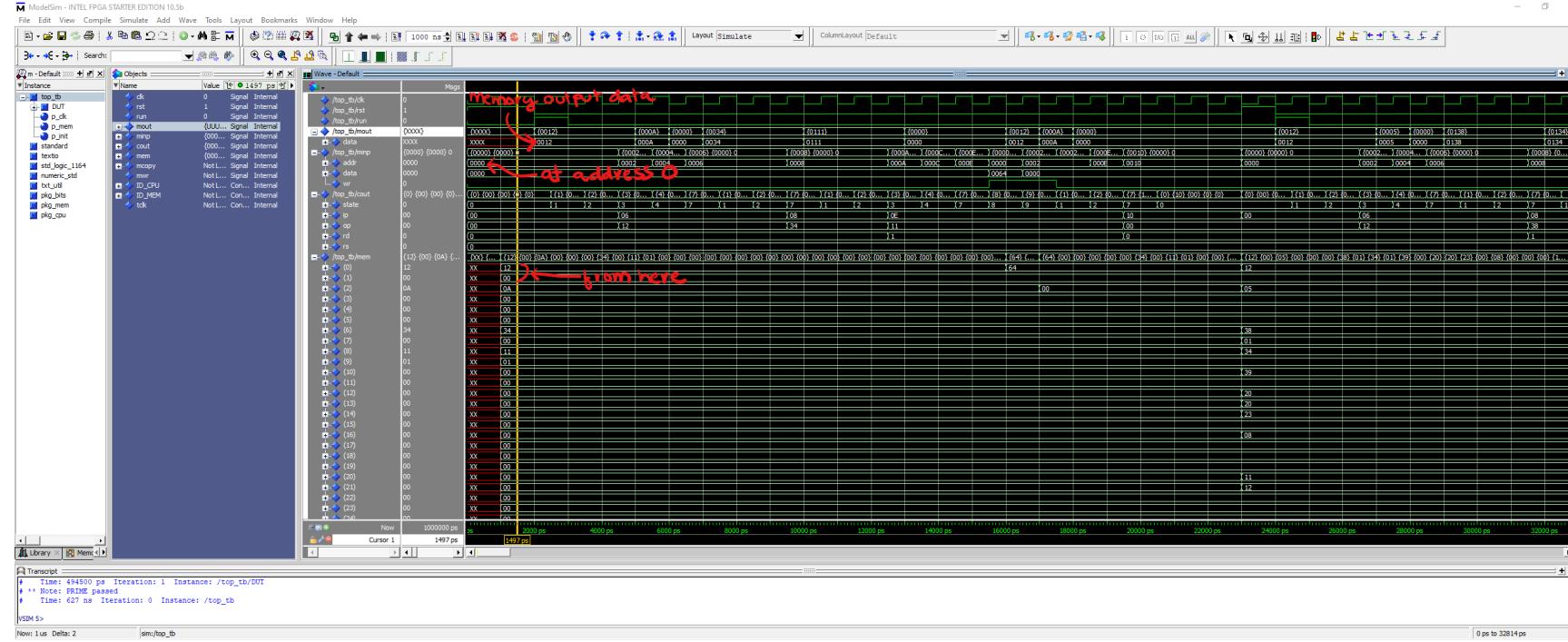
13 is prime

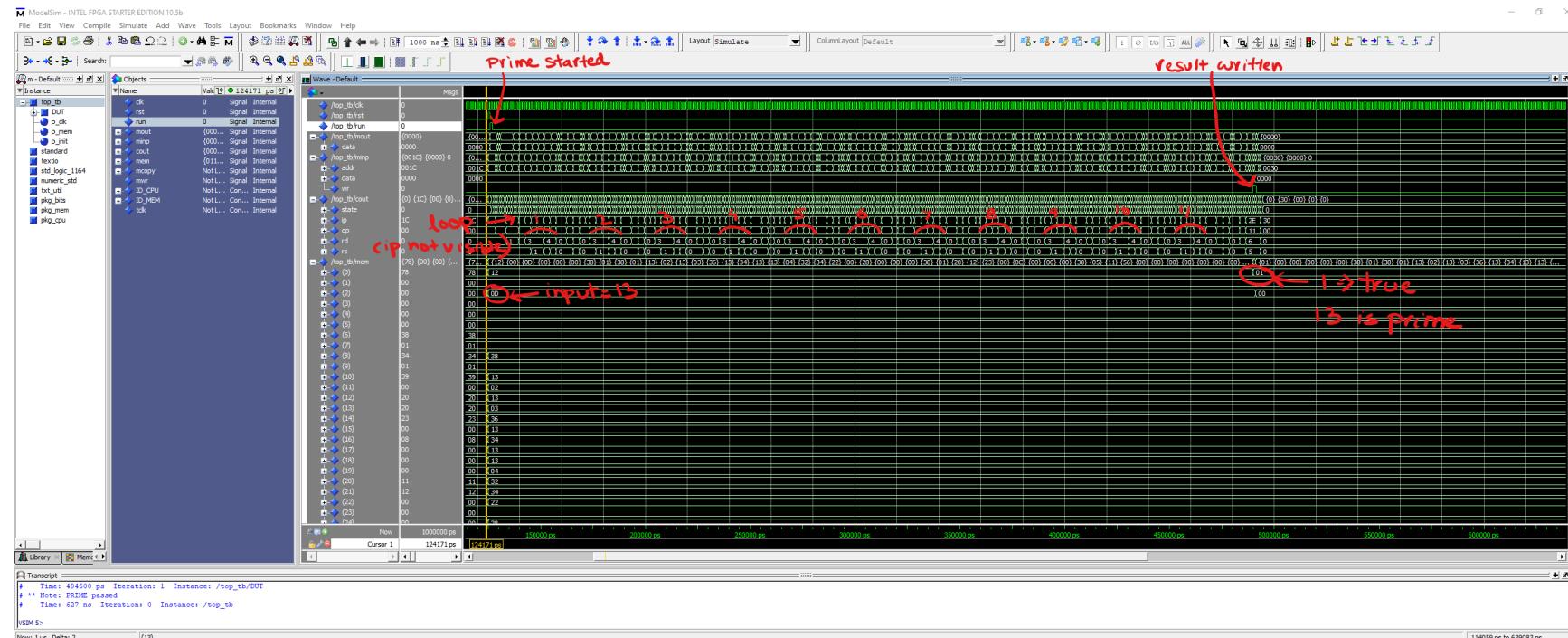
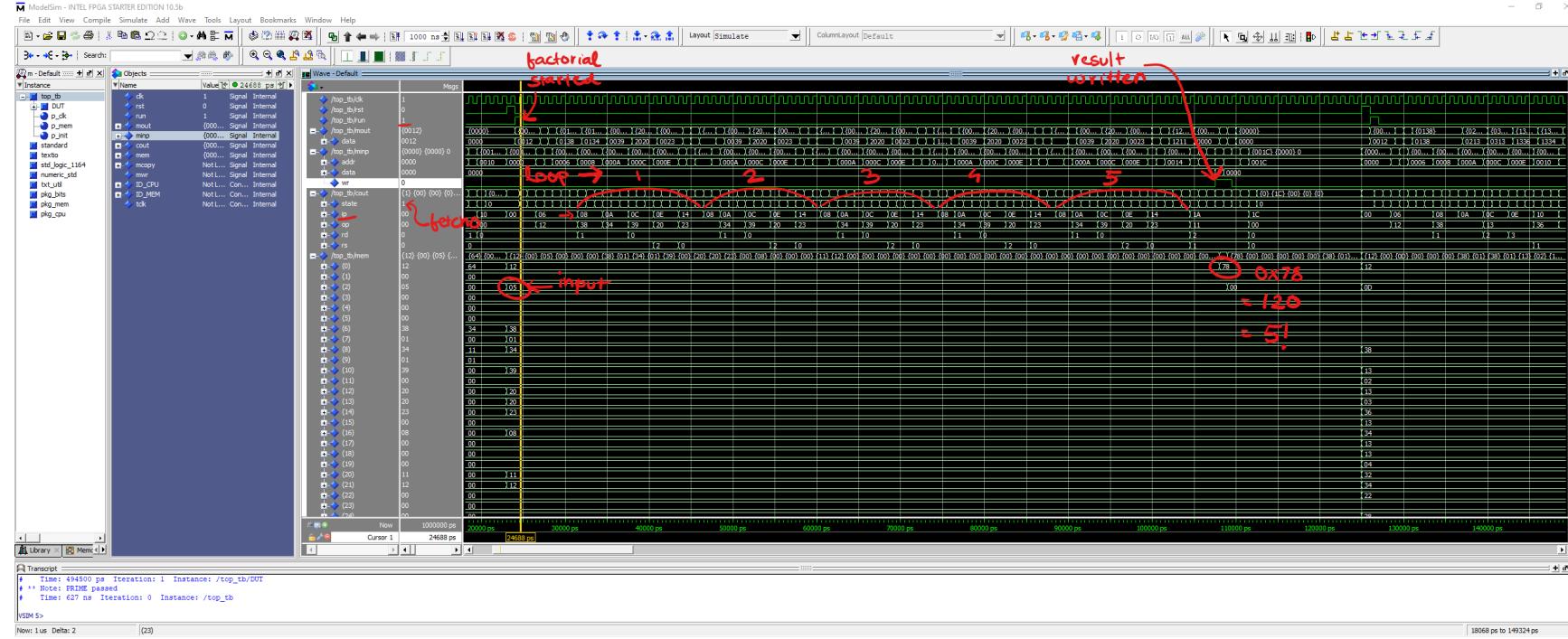
LN 21, COL 20 SPACES: 2 UTF-8 CRLF VHDL

simulation









report

Activities Visual Studio Code Sat 10:16 AM report.log - cpu_basic - Visual Studio Code

File Edit Selection View Go Run Terminal Help

OPEN EDITORS

- pkg_report.vhd
- top_tb.vhd
- report.log
- top.vhd

src > report.log

```
1  ../../src/ieee/v93/numeric_std-body.vhdl:2098:7:@0ms:(assertion warning): NUMERIC_STD.TO_INTEGER: me
2  top.vhd:39:5:@0ms:(report warning): RESET
3  ../../src/ieee/v93/numeric_std-body.vhdl:2098:7:@0ms:(assertion warning): NUMERIC_STD.TO_INTEGER: me
4  top.vhd:39:5:@0ms:(report warning): RESET
5  top.vhd:39:5:@500ps:(report warning): RESET
6  top.vhd:39:5:@1ns:(report warning): RESET
7  top.vhd:39:5:@1500ps:(report warning): RESET
8  top.vhd:184:13:@7500ps:(report note): movi r0, 0000000A
9  top.vhd:277:13:@10500ps:(report note): mul r0, r0
10 top.vhd:174:13:@15500ps:(report note): store [r1 + 00000000], r0
11 top_tb.vhd:61:7:@16ns:(report note): write @0: 0064
12 top_tb.vhd:61:7:@17ns:(report note): write @2: 0000
13 top.vhd:365:13:@20500ps:(report warning): INVALID OPCODE
14 top_tb.vhd:111:5:@23ns:(report note): SQUARE passed
15 top.vhd:39:5:@23ns:(report warning): RESET
16 top.vhd:39:5:@23ns:(report warning): RESET
17 top.vhd:39:5:@23500ps:(report warning): RESET
18 top.vhd:184:13:@29500ps:(report note): movi r0, 00000005
19 top.vhd:305:13:@32500ps:(report note): inc r1
20 top.vhd:277:13:@35500ps:(report note): mul r1, r0
21 top.vhd:312:13:@38500ps:(report note): dec r0
22 top.vhd:194:13:@41500ps:(report note): cmp r0, r2
23 top.vhd:212:13:@46500ps:(report note): jnz 00000008
24 top.vhd:277:13:@49500ps:(report note): mul r1, r0
25 top.vhd:312:13:@52500ps:(report note): dec r0
26 top.vhd:194:13:@55500ps:(report note): cmp r0, r2
27 top.vhd:212:13:@60500ps:(report note): jnz 00000008
28 top.vhd:277:13:@63500ps:(report note): mul r1, r0
29 top.vhd:312:13:@66500ps:(report note): dec r0
30 top.vhd:194:13:@69500ps:(report note): cmp r0, r2
31 top.vhd:212:13:@74500ps:(report note): jnz 00000008
32 top.vhd:277:13:@77500ps:(report note): mul r1, r0
33 top.vhd:312:13:@80500ps:(report note): dec r0
```

square started
input = 0xA = 10
result = 0x64 = 100 = 10²
halt

ghdl output

master 0 ▲ 0 △ 0

Ln 157, Col 1 Spaces: 2 UTF-8 LF Log

Activities Visual Studio Code Sat 10:16 AM report.log - cpu_basic - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER report.log top_tb.vhd top.vhd

OPEN EDITORS

- src > pkg_report.vhd
- top_tb.vhd
- report.log
- top.vhd

CPU_BASIC

- docs
- src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - report.log
 - top_tb.vhd
 - top.vhd
 - txt_util.vhd
 - work-obj93.cf
- .gitignore
- cpu_basic.qpf
- cpu_basic.qsf
- LICENSE
- README.md

factorial started
input = 5

```
14 top.vhd:111:5:@23ns:(report note): SQUARE passed
15 top.vhd:39:5:@23ns:(report warning): RESET
16 top.vhd:39:5:@23ns:(report warning): RESET
17 top.vhd:39:5:@23500ps:(report warning): RESET
18 top.vhd:184:13:@29500ps:(report note): movi r0, 00000005
19 top.vhd:305:13:@32500ps:(report note): inc r1
20 top.vhd:277:13:@35500ps:(report note): mul r1, r0
21 top.vhd:312:13:@38500ps:(report note): dec r0
22 top.vhd:194:13:@41500ps:(report note): cmp r0, r2
23 top.vhd:212:13:@46500ps:(report note): jnz 00000008
24 top.vhd:277:13:@49500ps:(report note): mul r1, r0
25 top.vhd:312:13:@52500ps:(report note): dec r0
26 top.vhd:194:13:@55500ps:(report note): cmp r0, r2
27 top.vhd:212:13:@60500ps:(report note): jnz 00000008
28 top.vhd:277:13:@63500ps:(report note): mul r1, r0
29 top.vhd:312:13:@66500ps:(report note): dec r0
30 top.vhd:194:13:@69500ps:(report note): cmp r0, r2
31 top.vhd:212:13:@74500ps:(report note): jnz 00000008
32 top.vhd:277:13:@77500ps:(report note): mul r1, r0
33 top.vhd:312:13:@80500ps:(report note): dec r0
34 top.vhd:194:13:@83500ps:(report note): cmp r0, r2
35 top.vhd:212:13:@88500ps:(report note): jnz 00000008
36 top.vhd:277:13:@91500ps:(report note): mul r1, r0
37 top.vhd:312:13:@94500ps:(report note): dec r0
38 top.vhd:194:13:@97500ps:(report note): cmp r0, r2
39 top.vhd:212:13:@102500ps:(report note): jnz 00000008
40 top.vhd:174:13:@107500ps:(report note): store [r2 + 00000000], r1
41 top_tb.vhd:61:7:@108ns:(report note): write @0: 0078 ← result : 0X78 = 120
42 top_tb.vhd:61:7:@109ns:(report note): write @2: 0000
43 top.vhd:365:13:@112500ps:(report warning): INVALID OPCODE
44 top_tb.vhd:153:7:@125ns:(report note): FACTORIAL passed
45 top.vhd:39:5:@125ns:(report warning): RESET
46 top.vhd:39:5:@125ns:(report warning): RESET
47 top.vhd:39:5:@125500ps:(report warning): RESET
```

halt = 5!

Ln 157, Col 1 Spaces: 2 UTF-8 LF Log

Activities Visual Studio Code Sat 10:16 AM report.log - cpu_basic - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER pkg_report.vhd top_tb.vhd report.log top.vhd

OPEN EDITORS src > report.log

CPU_BASIC

docs

src

- pkg_bits.vhd
- pkg_cpu.vhd
- pkg_mem.vhd
- pkg_report.vhd
- report.log
- top_tb.vhd
- top.vhd
- txt_util.vhd
- work-obj93.cf

.gitignore

cpu_basic.qpf

cpu_basic.qsf

LICENSE

README.md

OUTLINE

TIMELINE

Prime started

input: 0XD = 13

45 top.vhd:39:5:@125ns:(report warning): RESET
46 top.vhd:39:5:@125ns:(report warning): RESET
47 top.vhd:39:5:@125500ps:(report warning): RESET
48 top.vhd:184:13:@131500ps:(report note): movi r0, 0000000D
49 top.vhd:305:13:@134500ps:(report note): inc r1
50 top.vhd:305:13:@137500ps:(report note): inc r1
51 top.vhd:189:13:@140500ps:(report note): mov r2, r0
52 top.vhd:189:13:@143500ps:(report note): mov r3, r0
53 top.vhd:291:13:@146500ps:(report note): div r3, r1
54 top.vhd:277:13:@149500ps:(report note): mul r3, r1
55 top.vhd:189:13:@152500ps:(report note): mov r4, r0
56 top.vhd:262:13:@155500ps:(report note): sub r4, r3
57 top.vhd:205:13:@160500ps:(report note): jz 00000028
58 top.vhd:305:13:@163500ps:(report note): inc r1
59 top.vhd:194:13:@166500ps:(report note): cmp r2, r1
60 top.vhd:212:13:@171500ps:(report note): jnz 0000000C
61 top.vhd:189:13:@174500ps:(report note): mov r3, r0
62 top.vhd:291:13:@177500ps:(report note): div r3, r1
63 top.vhd:277:13:@180500ps:(report note): mul r3, r1
64 top.vhd:189:13:@183500ps:(report note): mov r4, r0
65 top.vhd:262:13:@186500ps:(report note): sub r4, r3
66 top.vhd:205:13:@191500ps:(report note): jz 00000028
67 top.vhd:305:13:@194500ps:(report note): inc r1
68 top.vhd:194:13:@197500ps:(report note): cmp r2, r1
69 top.vhd:212:13:@202500ps:(report note): jnz 0000000C
70 top.vhd:189:13:@205500ps:(report note): mov r3, r0
71 top.vhd:291:13:@208500ps:(report note): div r3, r1
72 top.vhd:277:13:@211500ps:(report note): mul r3, r1
73 top.vhd:189:13:@214500ps:(report note): mov r4, r0
74 top.vhd:262:13:@217500ps:(report note): sub r4, r3
75 top.vhd:205:13:@222500ps:(report note): jz 00000028
76 top.vhd:305:13:@225500ps:(report note): inc r1
77 top.vhd:194:13:@228500ps:(report note): cmp r2, r1

Ln 157, Col 1 Spaces: 2 UTF-8 LF Log

Activities Visual Studio Code Sat 10:16 AM 43% ▷ 🔍 ⌂ ⌂ ⌂

File Edit Selection View Go Run Terminal Help report.log - cpu_basic - Visual Studio Code

EXPLORER pkg_report.vhd top_tb.vhd report.log top.vhd

OPEN EDITORS src > report.log

CPU_BASIC

- docs
- src
 - pkg_bits.vhd
 - pkg_cpu.vhd
 - pkg_mem.vhd
 - pkg_report.vhd
 - report.log
 - top_tb.vhd
 - top.vhd
 - txt_util.vhd
 - work-obj93.cf
- .gitignore
- cpu_basic.qpf
- cpu_basic.qsf
- LICENSE
- README.md

OUTLINE TIMELINE

master ⌂ 0 △ 0

126 top.vhd:277:13:@397500ps:(report note): mul r3, r1
127 top.vhd:189:13:@400500ps:(report note): mov r4, r0
128 top.vhd:262:13:@403500ps:(report note): sub r4, r3
129 top.vhd:205:13:@408500ps:(report note): jz 00000028
130 top.vhd:305:13:@411500ps:(report note): inc r1
131 top.vhd:194:13:@414500ps:(report note): cmp r2, r1
132 top.vhd:212:13:@419500ps:(report note): jnz 0000000C
133 top.vhd:189:13:@422500ps:(report note): mov r3, r0
134 top.vhd:291:13:@425500ps:(report note): div r3, r1
135 top.vhd:277:13:@428500ps:(report note): mul r3, r1
136 top.vhd:189:13:@431500ps:(report note): mov r4, r0
137 top.vhd:262:13:@434500ps:(report note): sub r4, r3
138 top.vhd:205:13:@439500ps:(report note): jz 00000028
139 top.vhd:305:13:@442500ps:(report note): inc r1
140 top.vhd:194:13:@445500ps:(report note): cmp r2, r1
141 top.vhd:212:13:@450500ps:(report note): jnz 0000000C
142 top.vhd:189:13:@453500ps:(report note): mov r3, r0
143 top.vhd:291:13:@456500ps:(report note): div r3, r1
144 top.vhd:277:13:@459500ps:(report note): mul r3, r1
145 top.vhd:189:13:@462500ps:(report note): mov r4, r0
146 top.vhd:262:13:@465500ps:(report note): sub r4, r3
147 top.vhd:205:13:@470500ps:(report note): jz 00000028
148 top.vhd:305:13:@473500ps:(report note): inc r1
149 top.vhd:194:13:@476500ps:(report note): cmp r2, r1
150 top.vhd:212:13:@481500ps:(report note): jnz 0000000C
151 top.vhd:305:13:@484500ps:(report note): inc r5
152 top.vhd:174:13:@489500ps:(report note): store [r6 + 00000000], r5
153 top_tb.vhd:61:7:@490ns:(report note): write @0: 0001 ← result = 1 =>
154 top_tb.vhd:61:7:@491ns:(report note): write @2: 0000
155 top.vhd:365:13:@494500ps:(report warning): INVALID OPCODE
156 top_tb.vhd:213:5:@627ns:(report note): PRIME passed
157

Ln 157, Col 1 Spaces: 2 UTF-8 LF Log ↻

assert