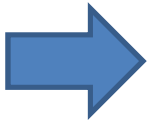
Loops in C – compilation stages in GCC – code in VHDL

K. Przygoda

Struktura partycji

C/C++

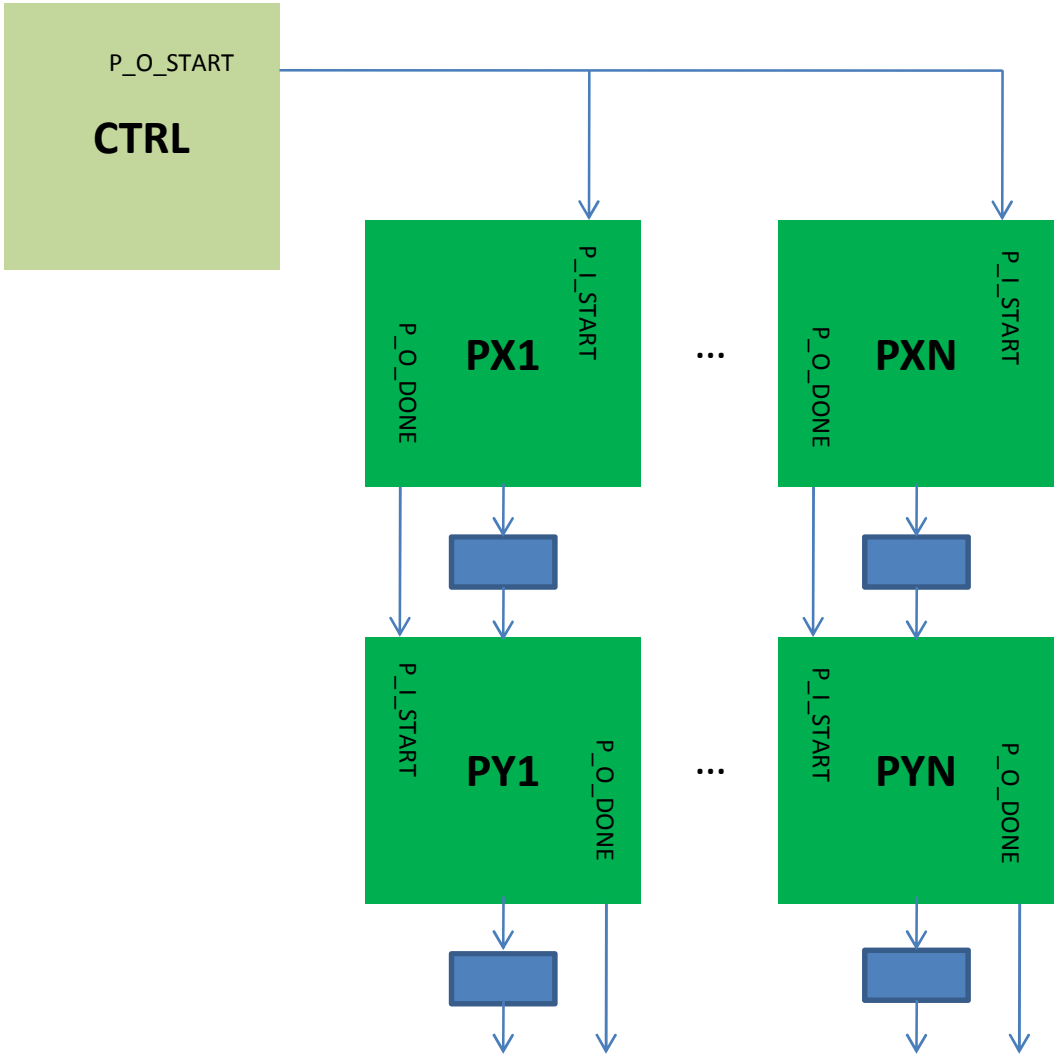
```
int p_o_arg = fun (int p_i_arg)
```



VHDL

```
entity fun is
generic
(
                GEN_PAR_WIDTH : natural := 18
);
port
(
--hardware interface
                P_I_CLK : in  STD_LOGIC;
                P_I_RST : in std_logic;
                P_I_START : in std_logic;
                P_O_DONE : out std_logic;
--loop interface
                P_I_ARG : in std_logic_vector(GEN_PAR_WIDTH-1 downto 0);
                P_O_ARG : out std_logic_vector(GEN_PAR_WIDTH-1 downto 0)

);
end entity ;
```

Komunikacja między partycjami



Typy danych HLL/VHDL

HLL Expression	VHDL Expression
variable representation	std_logic_vector
unsigned integer variable	signal name : unsigned (GEN_PAR_WIDTH-1 downto 0)
signed integer variable	signal name : signed (GEN_PAR_WIDTH-1 downto 0)
integer variable	signal name : integer
floating point variable	signal name : float (GEN_PAR_WIDTH-1 downto – (GEN_PAR_WDITH-1))
	conversion to std_logic_vector
unsigned integer variable	conv_std_logic_vector(signed, size) ⇔ conv_integer(slv)
signed integer variable	conv_std_logic_vector(unsigned, size) ⇔ conv_integer(slv)
integer variable	conv_std_logic_vector(integer, size) ⇔ conv_integer(signed/unsigned, slv)
floating point variable	to_slv(float) ⇔ to_float(slv, size)

```
library IEEE;
use IEEE.std_logic_signed.all
use IEEE.std_logic_usigned.all
use IEEE.float_pkg.all
```

Float_ug.pdf

Przykłady konwersji Gimple <=> VHDL

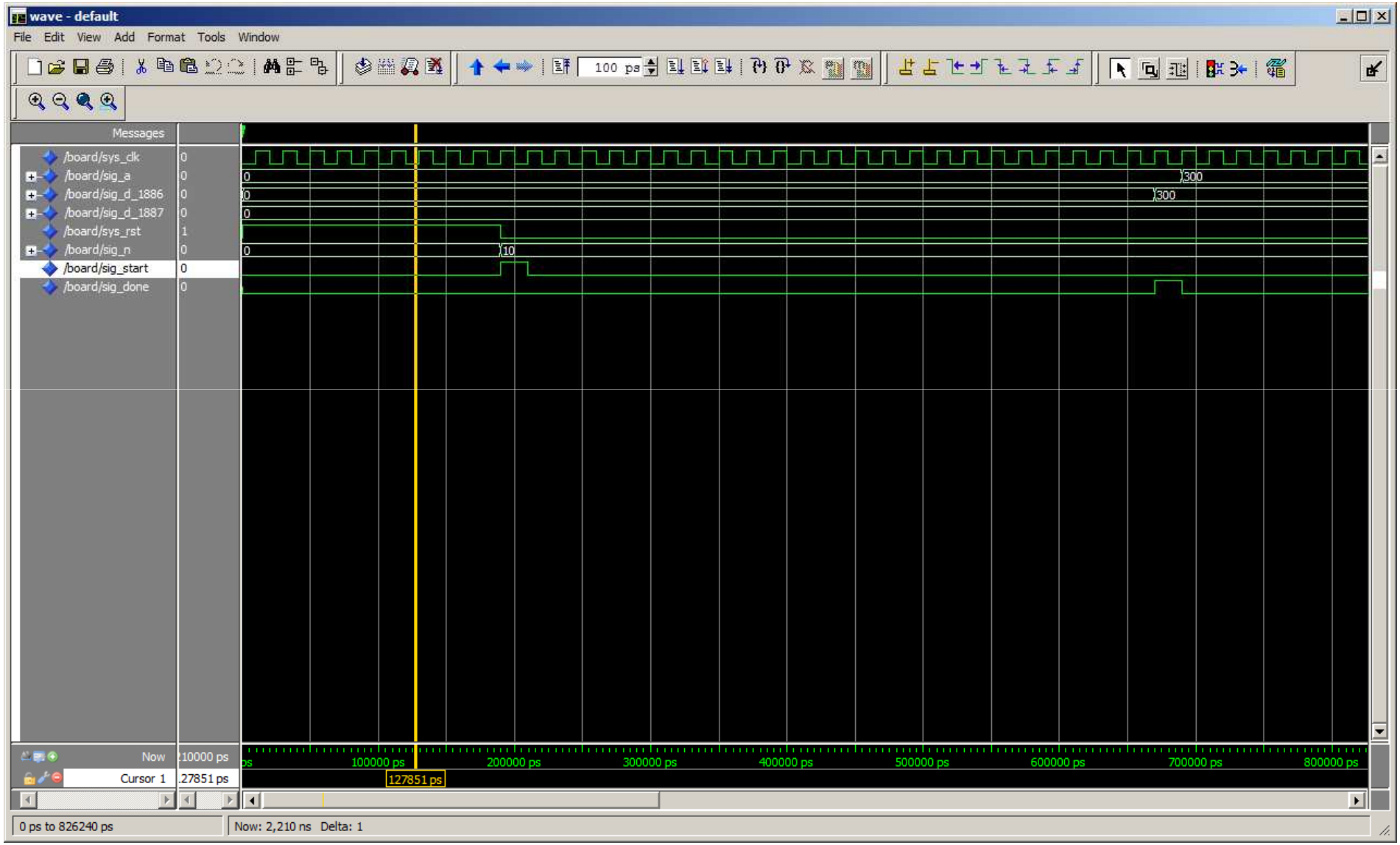
- Przykład 1 - pętle: while(cond){instr};
- Przykład 2 – primes

Przykład 1 VHDL

```
architecture a of fun is

type state is (start, bb_2, L1, L0, L2);

begin

loop1: process(P_I_CLK, P_I_RST)

variable arg3 : integer;
variable arg2 : integer;
variable arg1 : integer;
variable n : integer;
variable D_1880 : integer;
variable D_1879 : integer;
variable v_state: state;

begin
                If P_I_RST = '1' then
                v_state := start;
                P_O_ARG <= (others=>'0');
                P_O_DONE <= '0';
                elsif P_I_CLK'event and P_I_CLK = '1' then
                P_O_DONE <= '0';
```

```
case(v_state) is
                when start =>
                if P_I_START = '1' then
                                v_state := bb_2;
                end if;
                when bb_2 =>
                n := conv_integer(P_I_ARG);
                arg1 := 10;
                arg2 := 20;
                arg3 := 0;
                v_state := L1;
                when L0 =>
                D_1879 := arg1 + arg2;
                arg3 := D_1879 + arg3;
                n := n - 1;
                v_state := L1;
                when L1 =>
                if n /= 0 then
                                v_state := L0;
                else
                                v_state := L2;
                end if;

                when L2 =>
                D_1880 := arg3;
                P_O_ARG <= conv_std_logic_vector(D_1880, GEN_PAR_WIDTH);
                if P_I_START = '0' then
                                v_state := start;
                                P_O_DONE <= '1';
                end if;
                when others =>
                end case;
                end if;
end process;
end a;
```

Przykład 1 ModelSim



Przykład 2 VHDL

```
architecture a of primes_rev1 is

type arr is array(integer range<>) of integer;
type state is (start, bb_2, L_0, L_1, L_2, L_2_1, L_2_2, L_3, L_4, L_5, L_6, L_7);

Begin

process (P_I_CLK, P_I_RST)

                variable divisor : integer;
                variable number : integer;
                variable NUM : arr(0 to 10);
                variable D_2446 : integer;
                variable D_2448 : integer;
                variable number_0 : integer;
                variable mod_temp : integer;
                variable v_state : state;

Begin
                if P_I_RST = '1' then
                                v_state := start;
                                P_O_ARG <= (others=>'0');
                                P_O_DONE <= '0';
                                mod_temp := 0;
                elsif (P_I_CLK'event and P_I_CLK = '1') then

                P_O_DONE <= '0';
```

Środowisko do symulacji

- >compxlib
- >modelsim
- VSIM(paused)>do simulate_mti.do

- board_rtl.f
  - src/fun.vhd
  - src/primes_rev1.vhd
  - src/test.vhd

- simulate_mti.do
  - vlib work #create librart
  - vmap work #map physical and logical library
  - vcom -work work \ #compile the librart
  - -f board_rtl.f

  - vsim +notimingchecks \ #start modelsim
  - -L unisim -L work -L unisim work.board
  -
  - do wave.do #create waveform
  - run –all #run simulation