Аннотация

В рамках предложенной расчетно-графической работы необходимо создать аппаратный узел, выполняющий задачу преобразования цифровой информации с ее шифрованием по определенному алгоритму. Устройство необходимо спроектировать с использованием языка высокоуровневого описания аппаратуры VHDL.

Задание на разработку

В системе имеется два устройства: источник (И) и приемник (П) информации. Необходимо описать устройство являющееся посредником между устройствами И и П (рис. 1). Устройство И имеет две выходные однонаправленные шины данных: 32-х разрядную X и 4-х разрядную n, 8-и разрядную шину адреса Address, два выходных сигнала Write и Read и входной сигнал Ready. Устройство посредник вычисляет функцию

Y (0 : 31) =

f (X (0 : 32))

и по сигналу Write записывает вычисленные данные в ОЗУ по адресу, выставленному на шину адреса источника. По сигналу Read устройство посредник считывает байт по адресу, выставленному на шину адреса, и выдает его в последовательном коде приемнику предварительно пропустив через блок шифрования БШ. Выдача последовательного кода осуществляется, когда на сигнал готовности посредника к передаче Ready, приемник отвечает сигналом готовности приема Ask.

Блок шифрования

r1 = 6 r2 = 0

Функции

Bn cos x

где

∞ 1

Bk = ∑

π 2k − 1

p =1 p 2k

(2k )

Блок вычисления функции

Блок имеет две входные шины: 32-х разрядную X и 4-х разрядную n и выходную 32- х разрядную шину Y. Считать, что по шине X передается число с плавающей точкой в формате (рис. 3).

В блоке все операции производятся над вещественными типами данных. Для получения вещественных данных организовать функцию, осуществляющую перевод из типа STD\_LOGIC\_VECTOR в тип REAL, для получения двоичных данных из вещественных организовать функцию, осуществляющую обратный перевод. Аналогично организовать (или воспользоваться готовой, например, из пакета exemplar) функцию для перевода данных 4-х разрядной шины n в целый тип.

Устройство, вычисляет функцию Y = f ( X ) , посредством разложения в ряд с точностью определяемой количеством слагаемых равным значению n. При реализации описания использовать только стандартные арифметические операции "+", "-", "/" и собственную функцию возвращающую квадрат аргумента

rr :=b -1;

result:=mut(a,pwr(a,rr));

end if;

return result; \ъ

end pwr;

function toreal (X

:STD\_LOGIC\_VECTOR(31 downto 0))

return real is

variable result,res1,tt3,tt2,myn: real;

variable res2,tt1: integer;

begin

function toreal (X

:STD\_LOGIC\_VECTOR(31 downto 0))

return real is

variable result,res1,myn: real;

variable res2: integer;

begin

res1:= real(CONV\_INTEGER(X(20 downto 1)));

M1: while res1 >=1.0 loop res1:= res1/10.0;

end loop M1;

if X(0)= '1' then res1:=-res1;

end if;

res2:= CONV\_INTEGER(X(31 downto 22));

if X(20)= '1' then res2:=-res2;

end if;

myn := 2.0\*\*res2;

result:= mut(res1,myn) ;

return result;

end toreal;

function fakt (n :real) return real is variable rr : real;

variable result: real;

begin

if n =1.0 then result :=n;

elsif n = 0.0 then result :=0.0;

elsif n < 0.0 then

result := 0.0;

else

rr :=n -1.0;

result:=mut(n,fakt(rr));

end if;

return result;

end fakt;

res1:=0.0;

e1: for i in 20 downto 1 loop tt1 := i;

tt2 := 1.0/(pwr(2.0,tt1));

tt3 :=

mut(real(CONV\_INTEGER(X(i))),real(tt2)

);

res1:= res1 + real(tt3);

end loop e1;

if X(0)= '1' then res1:=-res1;

end if;

res2:= CONV\_INTEGER(X(31 downto

22));

myn := pwr(2.0,res2);

if X(21)= '1' then myn:=1.0/myn;

end if;

result:= mut(res1,myn) ;

return result;

end toreal;

function BK (X : real ; N : integer )

return real is

variable result, pi: real;

variable k1,k2,k3,k4: real;

begin pi:= 3.14;

for i in 1 to N loop k1 := mut(2.0,x); k2:=pwr(REAL(i),INTEGER(k1));

k3:=1.0/k2; k4:=(pwr(pi,INTEGER(k1))-

1.0)/fakt(k1); result:=k4; end loop;

return result;

end BK;

function MYFUN (X : real ; N : integer

) return real is

variable result, pi: real;

variable k2, k1:integer;

variable RR, l1,l2,l3,g1,g2,g3: real;

begin pi:= 3.14;

result := 0.0;

if abs(x) > (3.14/2) then return result;

end if;

for i in 1 to N loop

K1 := integer(pwr(2.0,i)); K2:= k1 - 1;

L1 := pwr(2.0,k2);

L2 := pwr(2.0,k1); L3 := pwr(x,k1);

G1 := fakt(real(k1));

G2 := mut(REAL(n),g1); Rr := mut(l1,(l2-1.0));

G3 := BK(x,i);

Rr := mut(rr,G3); Rr := mut(rr,l3); result:=rr/g2;

end loop;

return result;

end MYFUN;

function tostd (X1 : real ) return

STD\_LOGIC\_VECTOR is variable

result:STD\_LOGIC\_VECTOR ( 31 downto

0)

:="00000000000000000000000000000000"

;

variable X,myn,a,b,c: real; variable pr,w: integer; begin

X :=X1;

pr:=0;

M2: while abs(X) >= 1.0 loop

X := X /2.0 ;

pr:=pr +1;

end loop M2;

result(31 downto 22):= CONV\_STD\_LOGIC\_VECTOR(pr,10);

result (21) := '0';

if X < 0.0 then

result (0) := '1';

end if;

X := abs(X);

a:= real(integer(X));

b:= real(X); c:=b-a; w:=1;

M3: while c/=0.0 loop

X := X \*10.0 ;

a:= real(integer(X));

b:= real(X); c:=b-a; w:=w+1;

exit M3 when w > 6;

end loop M3;

result(20 downto 1):= CONV\_STD\_LOGIC\_VECTOR(integer(X

),20);

return result;

end tostd;

begin

process ( X,N )

variable mynum,res : real; variable count : integer; begin

mynum := toreal(X);

count := CONV\_INTEGER(N); res := MYFUN(mynum,count); Y<= tostd (res);

end process;

end F;

Блок шифрования

Блок шифрования организован в виде сдвигового регистра с обратными связями, в которые включены элементы суммирования по модулю два, генерирующего поток ключей Результат генерации суммируется по модуля два с последовательным кодом, полученным с выхода блока преобразования параллельного кода в последовательный и выдается приемнику. Блок шифрования однократно при начале работы устройства загружается начальным значением Key по сигналу Load.

Блок шифрования реализовать на основе 30-и битового сдвигового регистра с обратными связями.

VHDL КОД:

library IEEE;

use IEEE.std\_logic\_1164.all;

entity BCODE is port (

CLK : in std\_logic; LOAD : in std\_logic; DIN : in std\_logic;

DATA : in std\_logic\_vector(29 downto 0); SO : out std\_logic

);

end entity;

architecture BCODE of BCODE is

signal TEMP\_SO : std\_logic\_vector(29 downto 0);

begin process(CLK,LOAD) begin

if LOAD = '1' then

TEMP\_SO <= DATA;

ELSif rising\_edge(CLK) then

TEMP\_SO <= ((TEMP\_SO(0)XOR

TEMP\_SO(18))XOR TEMP\_SO(19)) & TEMP\_SO(29 downto 1);

end if;

end process;

SO <= TEMP\_SO(0)XOR DIN;

end architecture;

Блок ОЗУ

ОЗУ с раздельными шинами чтения и записи данных

|  |  |  |  |
| --- | --- | --- | --- |
| we | data | addr | Q |
| 1 | data | <=addr | Data |
| 0 | X | <=addr | dataaddr |

VHDL КОД

library IEEE;

use IEEE.std\_logic\_1164.all;

use IEEE.std\_logic\_unsigned.all;

entity ram is port (

WE : in STD\_LOGIC;

ADDR : in STD\_LOGIC\_VECTOR(7 downto 0);

DATA : in STD\_LOGIC\_VECTOR (31 downto 0);

Лист

Q : out STD\_LOGIC\_VECTOR (31 downto 0)

);

end entity;

architecture ram\_arch of ram is

Блок ОЗУ

type ram\_mem\_type is array (254 downto 0) of STD\_LOGIC\_VECTOR (31 downto 0);

signal ram\_mem : ram\_mem\_type;

begin

process (WE, ADDR, DATA)

variable ADDR\_TEMP: integer range 254 downto 0;

begin

if (WE = '1') then

ADDR\_TEMP := CONV\_INTEGER(ADDR);

ram\_mem(ADDR\_TEMP) <= DATA;

end if;

end process;

Q <= ram\_mem(CONV\_INTEGER(ADDR));

end architecture;

Блок-преобразователь параллельного кода в последовательный

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| clk | load | data | reg | so |
| 0 | 0 | X | data | data(0) |
| 1 | 1 | data | data | data(0) |
| 1 | 0 | X | 0.data | data(1) |

VHDL КОД

library IEEE;

use IEEE.std\_logic\_1164.all;

entity regpiso is port (

);

end entity;

CLK : in std\_logic; LOAD : in std\_logic;

DATA : in std\_logic\_vector(31 downto 0); SO : out std\_logic

architecture regpiso of regpiso is

signal TEMP\_SO : std\_logic\_vector(31 downto 0);

begin

process(CLK)

begin

if rising\_edge(CLK) then if LOAD = '1' then

TEMP\_SO <= DATA;

end if;

end process;

else end if;

TEMP\_SO <= '0' & TEMP\_SO(31 downto 1);

SO <= TEMP\_SO(0);

end architecture;

Блок устройства управления

Устройство управление реализовать управляющим а автоматом граф переходов, которого описать в редакторе FSM

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| read | ask | c | num | load | clk1 | clk2 | stb | ready |
| 0 | X | X | 0 | 0 | 0 | 0 | 0 | 1 |
| X | 0 | x | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| X | X | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| x | x | 1 | 2 | 0 | 1 | 0 | 0 | 0 |
| …. | … | .. | … | … | ….. | … | … | …. |
| x | x | 1 | 32 | 0 | 0 | 1 | 1 | 0 |
| X | X | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

VHDL Код

library IEEE;

use IEEE.std\_logic\_1164.all;

entity FUNC is port(

ASK : in STD\_LOGIC; LOAD : in STD\_LOGIC; READ : in STD\_LOGIC; WR : in STD\_LOGIC;

ADDR : in STD\_LOGIC\_VECTOR(7 downto 0); KEY : in STD\_LOGIC\_VECTOR(29 downto 0); N : in STD\_LOGIC\_VECTOR(3 downto 0);

X : in STD\_LOGIC\_VECTOR(31 downto 0); READY : out STD\_LOGIC;

READYO : out STD\_LOGIC; RESULT : out STD\_LOGIC; STB : out STD\_LOGIC

);

end FUNC;

architecture FUNC of FUNC is

---- Component declarations ----- component bcode

port (

CLK : in STD\_LOGIC;

DATA : in STD\_LOGIC\_VECTOR(29 downto 0); DIN : in STD\_LOGIC;

LOAD : in STD\_LOGIC; SO : out STD\_LOGIC

);

end component; component f port (

N : in STD\_LOGIC\_VECTOR(3 downto 0); X : in STD\_LOGIC\_VECTOR(31 downto 0); Y : out STD\_LOGIC\_VECTOR(31 downto 0)

);

end component; component kontroler port (

ASK : in STD\_LOGIC; C : in STD\_LOGIC; READ : in STD\_LOGIC; CLK1 : out STD\_LOGIC; CLK2 : out STD\_LOGIC;

LOAD : out STD\_LOGIC;

READY : out STD\_LOGIC; STB : out STD\_LOGIC

);

end component; component oscill port (

CLOCK : out STD\_LOGIC

);

end component; component ram port (

ADDR : in STD\_LOGIC\_VECTOR(7 downto 0); DATA : in STD\_LOGIC\_VECTOR(31 downto 0); WE : in STD\_LOGIC;

Q : out STD\_LOGIC\_VECTOR(31 downto 0)

);

end component;

component regpiso

port (

CLK : in STD\_LOGIC;

DATA : in STD\_LOGIC\_VECTOR(31 downto 0); LOAD : in STD\_LOGIC;

SO : out STD\_LOGIC

);

end component;

---- Signal declarations used on the diagram ----

signal CLK1 : STD\_LOGIC; signal CLK2 : STD\_LOGIC; signal LO1 : STD\_LOGIC; signal NET578 : STD\_LOGIC; signal NET908 : STD\_LOGIC; signal RED : STD\_LOGIC;

signal REDY : STD\_LOGIC; 11

signal BUS127 : STD\_LOGIC\_VECTOR (31 downto 0);

signal BUS534 : STD\_LOGIC\_VECTOR (31 downto 0);

begin

U1 : f

port map(

N => N, X => X,

Y => BUS127

);

U2 : ram port map(

ADDR => ADDR, DATA => BUS127,

Q => BUS534, WE => WR

);

U3 : regpiso port map(

CLK => CLK1, DATA => BUS534, LOAD => LO1,

SO => NET578

);

U4 : bcode port map(

CLK => CLK2,

DATA => KEY,

DIN => NET578,

LOAD => LOAD, SO => RESULT

);

U5 : kontroler port map(

ASK => ASK, C => NET908, CLK1 => CLK1, CLK2 => CLK2, LOAD => LO1, READ => RED, READY => REDY, STB => STB

);

U6 : oscill port map(

CLOCK => NET908

);

RED <= READ;

-- Output\buffer terminals READY <= REDY; READYO <= REDY;

end FUNC;

Временная диаграмма

Диаграмма потока данных:

KEY

LOA

Лист

N(3: X(31

func

f

mut

pwr

mut

pwr

ADR(7

WR READ READ

a

tore l

a a\*

myn

a^

myn

a

a\* MYF

UN

n re

re

a\* a tost

d

BK MYF

RESU ST

READ

AS

X(31

X(31

PROCESS

mynum,res : real count : integer;

Y(31:0

N(3:

UN

fakt

ram

Y(31

regpiso

Y(31 Y(0)

mu

BCODE

result

adr(7:

process (WE, ADDR, DATA) ram\_mem

process(CLK) TEMP\_SO

process(CLK, LOAD) TEMP SO

CLK CLK

KONTROLER

process (C) NUM

Заключение

Структура шифрующе-вычисляющего устройства была описана на языке VHDL и отлажена с использованием пакета Aldec Active-HDL.

Описание предложенного алгоритма на языке описания аппаратуры оказалось несложным и повторяет подходы и принципы, используемые при подобных устройств на обычных высокоуровневых языках программировании программирования.

Использованный программный пакет в свою очередь предоставляет разработчику мощный арсенал средств для отладки, моделирования и верификации описанного устройства.

Список литературы

1. Синтез логических схем с использованием языка VHDL. Бибило П.Н.

2. Проектирование цифровых систем на VHDL. Суворова E. А., Шейнин Ю. Е.

3. Fundamentals of Digital Logic with VHDL. Brown S.