Joseph Thornton

Professor Faroughi

CSC 137/ Section 3

Project 4

DATA path and Control Unit

1. Data path circular logic

2 to 1 mux

Add/Sub

A B C D

Register

2 to 1 mux

2 to 1 mux

Modee

S0

S2

S1

//report A Control unit block diagram

Q0 d0

q(bar) r

Q1 d1

q(bar) r

Clk reset mode start

OG

NSG

cs

d0

d1

Done

//report a FSD with RTNS

Reset

Start=0,Mode=d/Do nothing

Start =1

Mode=d/R←A

Start =d

Mode=0/R←R+B

Start =d

Mode=1/R←R-B

Start =d

Mode=1/R←R+C

Start =d

Mode=0/R←R-C

Start =d

Mode=d/R←R-D

Start =d

Mode=d/Do Nothing

//Reprt a FSD with Control signals

Start =d

Mode=d/e=0,m=d,s0=d,s1=d,s2=d

Reset

Start =0

Mode=d/e=0,m=d,s0=d,s1=d,s2=d

Start =1

Mode=d/e=1,m=d,s0=0,s1=d,s2=d,done=0

Start =d

Mode=0/e=1,m=0,s0=1,s1=0,s2=1,done=0

Start =d

Mode=1/e=1,m=1,s0=1,s1=0,s2=1,done=0

Start =d

Mode=1/e=1,m=0,s0=1,s1=d,s2=0,done=0

Start =d

Mode=0/e=1,m=1,s0=1,s1=d,s2=1,done=0

Start =d

Mode=d/e=0,m=d,s0=d,s1=d,s2=d,done=0

//Report B Verilog models

//Control unit

module controlunit(

input clock,reset,start,mode,

output reg [5:0] cs //m,s0,s1,s2,done,e,

);

parameter A=3'b000,

B=3'b001,

C=3'b010,

D=3'b011,

F=3'b100;

reg [2:0] current\_state,next\_state;

always@\*

begin

casex (current\_state)

A: if(start==1'b0)

next\_state=A;

else

next\_state=B;

B: next\_state=C;

C: next\_state=D;

D: next\_state=F;

F: next\_state=F;

endcase

end

//OG

always@\*

begin

casex (current\_state)

A:if(start==0)

cs=6'b000000;

else

cs=6'b100000;

B: if(mode==1)

cs=6'b101011;

else

cs=6'b101010;

C:if(mode==1)

cs=6'b101110;

else

cs=6'b101111;

D: cs=6'b100111;

F: cs=6'b010000;

endcase

end

//flip flop

always@ (posedge clock,reset)

begin

if (reset==1)begin

current\_state<=A;

//$display("executed");

end else

current\_state<=next\_state;

//$display("%d %b",$time,current\_state);

end

//assign state=current\_state;

Endmodule

//2 to 1 Mux

module mux(

input s,

input [7:0] a,b,

output reg [7:0] o

);

always@\*

begin

if (s==1'b0)

o=a;

else

o=b;

end

endmodule

// combine modules note: im not putting the adder modules in this file to save paper

`include "/gaia/class/student/thorntjl/csc137/project4/2to1mux.v"

`include "/gaia/class/student/thorntjl/csc137/project4/controlunit.v"

`include "/gaia/class/student/thorntjl/csc137/project4/AddSub.v"

module combine(

input clock,reset,start,mode,//clock,reset,start,mode

input [7:0] a,b,c,d,

output reg [7:0] result,

output done

);

wire [5:0] cs;

wire [7:0] answer;

wire [7:0] o1,o2,o3;

wire ovf;

controlunit c1(clock,reset,start,mode,cs);

mux m1(cs[3],d,o2,o3);

mux m2(cs[2],b,c,o2);

AddSub add1(result,o3,cs[0],answer,ovf);

mux m3(cs[1],a,answer,o1);

assign done=cs[4];

//register

always@ (posedge clock, reset)

begin

if(reset==1)

result<=0;

else if(cs[5]==1)

result<=o1;

else

result<=result;

//$display("%d %b %8h %b",$time,cs,result,reset);

end

endmodule

// report C output

Chronologic VCS simulator copyright 1991-2014

Contains Synopsys proprietary information.

Compiler version I-2014.03-2; Runtime version I-2014.03-2; May 4 09:41 2017

Time result done

a=01 b=02 c=ff d=fe

clock1: 15 01 0

clock2: 25 03 0

clock3: 35 04 0

answer: 45 06 1

reset: 75 00 0

a=fe b=01 c=01 d=04

clock1: 85 fe 0

clock2: 95 fd 0

clock3: 105 fe 0

answer: 115 fa 1

reset: 145 00 0

a=01 b=ff c=ff d=ff

clock1: 155 01 0

clock2: 165 00 0

clock3: 175 01 0

answer: 185 02 1

reset : 215 00 0

a=ff b=01 c=ff d=01

clock1: 225 ff 0

clock2: 235 fe 0

clock3: 245 fd 0

clock4: 255 fc 1

reset : 285 00 0

$finish called from file "testbench.v", line 43.

$finish at simulation time 350

V C S S i m u l a t i o n R e p o r t

Time: 350

CPU Time: 0.300 seconds; Data structure size: 0.0Mb

Thu May 4 09:41:23 2017

[thorntjl@athena:21]> exit