Add Source Files

Look in:
250228_adder

250228_adder.cache

250228_adder.hw

250228_adder.ip_user_files

250228_adder.runs

250228_adder.sim

250228 adder.srcs

Source"

File name:
C:/Working/FPGA_Harman_25_1/250228_adder

Files of type: Design Source Files (.vhd, vhdl, vhf, vhdp, vho, v, vf, verilog, vr, v
Add Sources

Add or Create Design Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source file on disk and add it to your project.

Index

1
Name

adder.v

Library

xil_defaultlib
Location

2
fnd_controller.v xil_defaultlib
C:/Working/FPGA_Harman_25_1/250228_adder/250228_adder.srcs/sources_1

C:/Working/FPGA_Harman_25_1/250228_adder/250228_adder.srcs/sources_1

>

Add Files
Add Directories
Create File

Scan and add RTL include files into project

Copy sources into project

Add sources from subdirectories

Recurse all subdirectories and add found HDL files to your project.
?
< Back
Next >
Finish
Cancel
X

Btn

JAY.

Cululator

FAD Controlle

sel.

seblomm

Btu

2x4

Мих

Sel

8

Sum

L

W_Sunk

Addors

4.

BCD

b

to seg se Data

3日日日日

a t

b

\+ Actor

C

Calulator

хол

End_Controller

Thit -9bit

4w_digit

`digit digit_|`

Plitter digit- to

`100` `~-TODO`

Sel

2x `Mux`

лих

4 BCD

Suk

seslomm

노

日日日日

tosey Seg Data 1000

100

1001

255 (22) 255/10=

14/hey

10 0/2121 255/10 %.10=

1512

256.

Sum

10

의자리

[14]

[3:0]

100 +42021 255/100/10

1000

0/242/255/1000 / 210 =

10000/2414.

10092621

46rt,

4

W

Carry

abit

On ъ12,

15

acq; t] ↑

// 9비트 {carry 1bit, sum 8bit}

-MSB

LSB =

*결합 연산자(MSB

LOBS

Sam[4:0] Ад

a

:0] ac1:0]

ата

461 FA Carry 4bifFA

upper

44

な

acsit

a[:0]

Lower

160

b

та

[achit]

a [3:0]

w_seg_sel

44

```
module counter_4 (
```

45

46

```
input

input
clk,

reset,
```

CLK-

47
```
output [1:0] o_sel
```
48
```
);
```
49

50

51

52

Fb

53

54

55

56

```
reg [1:0] r_counter;

assign o_sel = r_counter;
```

OLIMEREN

```
always @(kosedge (1k posedge
```

reset) begin

if (reset) begin

r_counter <= 0;

end else begin

57

C

r counter =

r_counter + 1;

58

end

59

end

60

61

62

endmodule

123

+569

4

7

8

9

10

11

1233

14

15

16

17

18

19

20

21

1222

```
`timescale 1ns/1ps
// calculator 를 완성하시오.module calculator (









);
// counter_4
input clk,
input reset,
// adder_8
input [7:0] a,
input [7:0] b,
```

```
// 7-seg 2
output [7:0] seg,
output [3:0] seg_comm

wire [7:0] w_sum;
wire w_carry;

fnd_controller U_fnd_ctrl (
.clk(clk),
.reset(reset),
.bcd({w_carry, w_sum}), // 9bit w_carry, w_sum[7:0]
.seg(seg),
.seg_comm (seg_comm)


adder_8 U_fa_8 (
.a(a), // 4bit vertorg
.b(b),
sum(w_sum),
.carry (w_carry
23
);
24
25
26
27
28
29
```

```
30
);
31
endmodule
32
```

w_seg_sel

```
reg [$clog2(1_000_000)-1:0]
r_counter;
reg r_clk;
assign o_clk = r_clk;
```

```
53
54
55
56
module clk_divider (
input clk,
input reset,
```

$clog2:

수익 필요한 비트수 계산,

```
ㄴ


57 v );
output o_clk

//$clog2 숫자를 나타내는데 필요한 비트수 계산
reg [$clog2(1_000_000)-1:0] r_counter; reg_r_clk;
assign o_clk = r_clk;
58

59

60

61

62
```

```
63 v
always
64 v
65
66
(posedge clk, posedge reset) begin
if (reset) begin
r_counter <= 0;
r_clk <= 1'b0;
67 v
end else begin
68
// clock divide,100Mhz -> 100hz
69 v
70
71
72 v
73
74
75
76
end
77
end
if (r_counter
==
1_000_000
```

```
1) begin
r_counter <= 0;
r_clk <= 1'b1; // r_clk : 0->1
end else begin


end
r_counter <= r_counter + 1;
r_clk <= 1'b0; // r_clk : 0 |.;
```

폴러 관리„

4=1421

Datu Path

CCK

100MH2

reset

설계

CLK reser

0~9999

Counter

100002 Counter

?

HELP

Top_upConverter.

FND_Controller"

Cck 423

4진

Diver counts

Test

кошка

순차

조합

3/ Digit /

14

Count 32 14 SPIT

W_Count

014

1004

1000

Iter

4x1

мих

$

12x4

мих

み

BER

toseg

Sey_comm

Seg-Data

Fund_Controller

SW

0

сек

Reset

10

run clear

N

Stop

A

구르샤

铡

CLK Divider

코드사진

LOHE

ж
'10000집

Counter

FND_counter

run_Stop: 1:run

0: Stop

clear

1: dear

counter :"0"

0: