

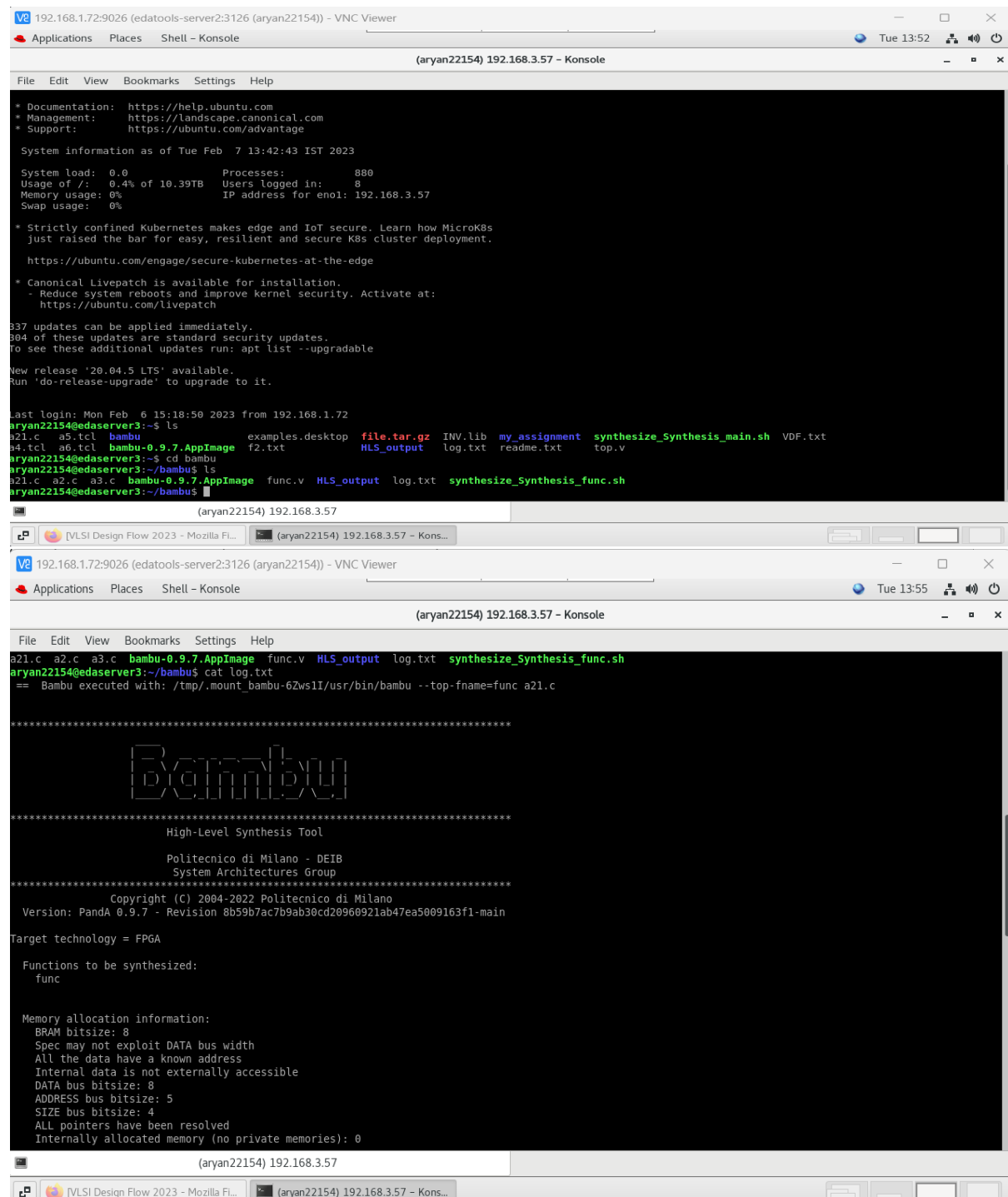
## ASSIGNMENT -2

Name: - Aryan Gupta

Roll No- MT22154

VLSI Design Flow (Submitted to Prof. Sneha Saurabh)

Q1)- Show a screenshot or snippet of the log file to prove that you have indeed run the tool [If this is not given, then your assignment will not be evaluated].



```
192.168.1.72:9026 (edatools-server2:3126 (aryan22154)) - VNC Viewer
Applications Places Shell - Konsole
(aryan22154) 192.168.3.57 - Konsole
File Edit View Bookmarks Settings Help
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information as of Tue Feb 7 13:42:43 IST 2023
System load: 0.0 Processes: 880
Usage of /: 0.4% of 10.39TB Users logged in: 8
Memory usage: 0% IP address for eno1: 192.168.3.57
Swap usage: 0%
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge
* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch
337 updates can be applied immediately.
884 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Mon Feb 6 15:18:50 2023 from 192.168.1.72
aryan22154@edaserver3:~$ ls
a21.c a5.tcl bambu examples.desktop file.tar.gz INV.lib my_assignment synthesize_Synthesis_main.sh VOF.txt
a4.tcl a6.tcl bambu-0.9.7.AppImage f2.txt HLS_output log.txt readme.txt top.v
aryan22154@edaserver3:~$ cd bambu
aryan22154@edaserver3:~/bambu$ ls
a21.c a2.c a3.c bambu-0.9.7.AppImage func.v HLS_output log.txt synthesize_Synthesis_func.sh
aryan22154@edaserver3:~/bambu$
(aryan22154) 192.168.3.57
[VLSI Design Flow 2023 - Mozilla Firefox] (aryan22154) 192.168.3.57 - Konsole
192.168.1.72:9026 (edatools-server2:3126 (aryan22154)) - VNC Viewer
Applications Places Shell - Konsole
(aryan22154) 192.168.3.57 - Konsole
File Edit View Bookmarks Settings Help
a21.c a2.c a3.c bambu-0.9.7.AppImage func.v HLS_output log.txt synthesize_Synthesis_func.sh
aryan22154@edaserver3:~/bambu$ cat log.txt
== Bambu executed with: /tmp/.mount_bambu-6Zws1I/usr/bin/bambu --top-fname=func a21.c
*****
Bambu
*****
High-Level Synthesis Tool
Politecnico di Milano - DEIB
System Architectures Group
*****
Copyright (C) 2004-2022 Politecnico di Milano
Version: Panda 0.9.7 - Revision 8b59b7ac7b9ab30cd20960921ab47ea5009163f1-main
Target technology = FPGA
Functions to be synthesized:
func
Memory allocation information:
BRAM bitsize: 8
Spec may not exploit DATA bus width
All the data have a known address
Internal data is not externally accessible
DATA bus bitsize: 8
ADDRESS bus bitsize: 5
SIZE bus bitsize: 4
ALL pointers have been resolved
Internally allocated memory (no private memories): 0
```

192.168.1.72:9026 (edatools-server2:3126 (aryan22154)) - VNC Viewer

Applications Places Shell - Konsole

(aryan22154) 192.168.3.57 - Konsole

File Edit View Bookmarks Settings Help

```
Internally allocated memory: 0
Time to perform memory allocation: 0.00 seconds

Module allocation information for function func:
  Number of complex operations: 0
  Number of complex operations: 0
Time to perform module allocation: 0.01 seconds

Scheduling Information of function func:
  Number of control steps: 3
  Minimum slack: 6.5193999989999982
  Estimated max frequency (MHz): 287.3067861037444
Time to perform scheduling: 0.00 seconds

State Transition Graph Information of function func:
  Number of states: 1
  Minimum number of cycles: 1
  Maximum number of cycles 1
Time to perform creation of STG: 0.00 seconds

Easy binding information for function func:
  Bound operations:8/10
Time to perform easy binding: 0.00 seconds

Storage Value Information of function func:
  Number of storage values inserted: 0
Time to compute storage value information: 0.00 seconds

Register binding information for function func:
  Register allocation algorithm obtains an optimal result: 0 registers
Time to perform register binding: 0.00 seconds
```

(aryan22154) 192.168.3.57

[VLSI Design Flow 2023 - Mozilla Fi... (aryan22154) 192.168.3.57 - Kons...

192.168.1.72:9026 (edatools-server2:3126 (aryan22154)) - VNC Viewer

Applications Places Shell - Konsole

(aryan22154) 192.168.3.57 - Konsole

File Edit View Bookmarks Settings Help

```
Easy binding information for function func:
  Bound operations:8/10
Time to perform easy binding: 0.00 seconds

Storage Value Information of function func:
  Number of storage values inserted: 0
Time to compute storage value information: 0.00 seconds

Register binding information for function func:
  Register allocation algorithm obtains an optimal result: 0 registers
Time to perform register binding: 0.00 seconds

Module binding information for function func:
  Number of modules instantiated: 10
  Number of possible conflicts for possible false paths introduced by resource sharing: 0
  Estimated resources area (no Muxes and address logic): 191
  Estimated area of MUX21: 0
  Total estimated area: 191
  Estimated number of DSPs: 0
  Slack computed in 0.00 seconds
  False-loop computation completed in 0.00 seconds
  Weight computation completed in 0.00 seconds
  Clique covering computation completed in 0.00 seconds
Time to perform module binding: 0.00 seconds

Register binding information for function func:
  Register allocation algorithm obtains an optimal result: 0 registers
Time to perform register binding: 0.00 seconds

Total number of flip-flops in function func: 0
aryan22154@edaserver3:~/bamBUS
```

(aryan22154) 192.168.3.57

[VLSI Design Flow 2023 - Mozilla Fi... (aryan22154) 192.168.3.57 - Kons...

Q2 ) Obtain the Verilog RTL for the following C code using the HLS tool.

```
latools-server2:3126 (aryan22154) - VNC Viewer
Applications Places Shell - Konsole

(aryan22154) 192.168.3.57 - Konsole

File Edit View Bookmarks Settings Help
a21.c a2.c a3.c bambu-0.9.7.AppImage func.v HLS_output log.txt synthesize_Synthesis_func.sh
aryan22154@edaserver3:~/bambu$ cat func.v
//
// Politecnico di Milano
// Code created using PAndA - Version: PAndA 0.9.7 - Revision 8b59b7ac7b9ab30cd20960921ab47ea5009163f1-main - Date 2023-02-06T15:28:32
// /tmp/.mount_bambu-6ZwsII/usr/bin/bambu executed with: /tmp/.mount_bambu-6ZwsII/usr/bin/bambu --top-fname=func a21.c
//
// Send any bug to: panda-info@polimi.it
// *****
// The following text holds for all the components tagged with PANDA_LGPLv3.
// They are all part of the BAMBU/PANDA IP LIBRARY.
// This library is free software; you can redistribute it and/or
// modify it under the terms of the GNU Lesser General Public
// License as published by the Free Software Foundation; either
// version 3 of the License, or (at your option) any later version.
//
// This library is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
// Lesser General Public License for more details.
//
// You should have received a copy of the GNU Lesser General Public
// License along with the PAndA framework; see the files COPYING.LIB
// If not, see <http://www.gnu.org/licenses/>.
// *****

#ifdef ICARUS
#define _SIM_HAVE_CLOG2
#endif
#ifdef VERILATOR
#define _SIM_HAVE_CLOG2
#endif
#ifdef MODEL_TECH
#define _SIM_HAVE_CLOG2
#endif
#ifdef VCS
#define _SIM_HAVE_CLOG2
#endif
```

```
latools-server2:3126 (aryan22154) - VNC Viewer
Applications Places Shell - Konsole

(aryan22154) 192.168.3.57 - Konsole

File Edit View Bookmarks Settings Help
`endif
`ifdef VCS
`define _SIM_HAVE_CLOG2
`endif
`ifdef NCVERILOG
`define _SIM_HAVE_CLOG2
`endif
`ifdef XILINX_SIMULATOR
`define _SIM_HAVE_CLOG2
`endif
`ifdef XILINX_ISIM
`define _SIM_HAVE_CLOG2
`endif

// This component is part of the BAMBU/PANDA IP LIBRARY
// Copyright (C) 2004-2022 Politecnico di Milano
// Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>, Christian Pilato <christian.pilato@polimi.it>
// License: PANDA LGPLv3
`timescale 1ns / 1ps
module constant_value(out1);
    parameter BITSIZE_out1=1,
        value=1'b0;
    // OUT
    output [BITSIZE_out1-1:0] out1;
    assign out1 = value;
endmodule

// This component is part of the BAMBU/PANDA IP LIBRARY
// Copyright (C) 2016-2022 Politecnico di Milano
// Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
// License: PANDA LGPLv3
`timescale 1ns / 1ps
module lut_expr_FU(in1,
    in2,
    in3,
    in4,
    in5,
```

```

in5,
in6,
in7,
in8,
in9,
out1);
parameter BITSIZE_in1=1,
        BITSIZE_out1=1;
// IN
input [BITSIZE_in1-1:0] in1;
input in2;
input in3;
input in4;
input in5;
input in6;
input in7;
input in8;
input in9;
// OUT
output [BITSIZE_out1-1:0] out1;
reg[7:0] cleaned_in0;
wire [7:0] in0;
wire[BITSIZE_in1-1:0] shifted_s;
assign in0 = {in9, in8, in7, in6, in5, in4, in3, in2};
generate
    genvar i0;
    for (i0=0; i0<8; i0=i0+1)
    begin : L0
        always @(*)
        begin
            if (in0[i0] == 1'b1)
                cleaned_in0[i0] = 1'b1;
            else
                cleaned_in0[i0] = 1'b0;
        end
    end
endgenerate

```

(aryan22154) 192.168.3.57

```

endgenerate
assign shifted_s = in1 >> cleaned_in0;
assign out1[0] = shifted_s[0];
generate
    if(BITSIZE_out1 > 1)
        assign out1[BITSIZE_out1-1:1] = 0;
    endgenerate
endmodule

// This component is part of the BAMBU/PANDA IP LIBRARY
// Copyright (C) 2004-2022 Politecnico di Milano
// Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
// License: PANDA_LGPLv3
`timescale 1ns / 1ps
module cond_expr_FU(in1,
    in2,
    in3,
    out1);
    parameter BITSIZE_in1=1,
            BITSIZE_in2=1,
            BITSIZE_in3=1,
            BITSIZE_out1=1;
    // IN
    input [BITSIZE_in1-1:0] in1;
    input signed [BITSIZE_in2-1:0] in2;
    input signed [BITSIZE_in3-1:0] in3;
    // OUT
    output signed [BITSIZE_out1-1:0] out1;
    assign out1 = in1 != 0 ? in2 : in3;
endmodule

// This component is part of the BAMBU/PANDA IP LIBRARY
// Copyright (C) 2004-2022 Politecnico di Milano
// Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
// License: PANDA_LGPLv3
`timescale 1ns / 1ps

```

(aryan22154) 192.168.3.57

File Edit View Bookmarks Settings Help

```

`timescale 1ns / 1ps
module gt_expr_FU(in1,
    in2,
    out1);
    parameter BITSIZE_in1=1,
        BITSIZE_in2=1,
        BITSIZE_out1=1;
    // IN
    input signed [BITSIZE_in1-1:0] in1;
    input signed [BITSIZE_in2-1:0] in2;
    // OUT
    output [BITSIZE_out1-1:0] out1;
    assign out1 = in1 > in2;
endmodule

// This component is part of the BAMBU/PANDA IP LIBRARY
// Copyright (C) 2004-2022 Politecnico di Milano
// Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
// License: PANDA LGPLv3
`timescale 1ns / 1ps
module le_expr_FU(in1,
    in2,
    out1);
    parameter BITSIZE_in1=1,
        BITSIZE_in2=1,
        BITSIZE_out1=1;
    // IN
    input signed [BITSIZE_in1-1:0] in1;
    input signed [BITSIZE_in2-1:0] in2;
    // OUT
    output [BITSIZE_out1-1:0] out1;
    assign out1 = in1 <= in2;
endmodule

// This component is part of the BAMBU/PANDA IP LIBRARY
// Copyright (C) 2004-2022 Politecnico di Milano
// Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>

```

(aryan22154) 192.168.3.57



[VLSI Design Flow 2023 - Mozilla Fi...

(aryan22154) 192.168.3.57 - Kons...

File Edit View Bookmarks Settings Help

```

// Copyright (C) 2004-2022 Politecnico di Milano
// Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
// License: PANDA LGPLv3
`timescale 1ns / 1ps
module minus_expr_FU(in1,
    in2,
    out1);
    parameter BITSIZE_in1=1,
        BITSIZE_in2=1,
        BITSIZE_out1=1;
    // IN
    input signed [BITSIZE_in1-1:0] in1;
    input signed [BITSIZE_in2-1:0] in2;
    // OUT
    output signed [BITSIZE_out1-1:0] out1;
    assign out1 = in1 - in2;
endmodule

// This component is part of the BAMBU/PANDA IP LIBRARY
// Copyright (C) 2004-2022 Politecnico di Milano
// Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
// License: PANDA LGPLv3
`timescale 1ns / 1ps
module plus_expr_FU(in1,
    in2,
    out1);
    parameter BITSIZE_in1=1,
        BITSIZE_in2=1,
        BITSIZE_out1=1;
    // IN
    input signed [BITSIZE_in1-1:0] in1;
    input signed [BITSIZE_in2-1:0] in2;
    // OUT
    output signed [BITSIZE_out1-1:0] out1;
    assign out1 = in1 + in2;
endmodule

```

(aryan22154) 192.168.3.57



[VLSI Design Flow 2023 - Mozilla Fi...

(aryan22154) 192.168.3.57 - Kons...

datools-server2:3126 (aryan22154) - VNC Viewer

Applications Places Shell - Konsole

(aryan22154) 192.168.3.57 - Konsole

```
File Edit View Bookmarks Settings Help

// Datapath RTL description for func
// This component has been derived from the input source code and so it does not fall under the copyright of Panda framework, but it follows the input source
// code copyright, and may be aggregated with components of the BAMBU/PANDA IP LIBRARY.
// Author(s): Component automatically generated by bambu
// License: THIS COMPONENT IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHAN
// TIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
timescale 1ns / 1ps
module datapath_func(clock,
    reset,
    in_port_j,
    in_port_k,
    in_port_c,
    in_port_d,
    return_port);
    // IN
    input clock;
    input reset;
    input signed [31:0] in_port_j;
    input signed [31:0] in_port_k;
    input signed [31:0] in_port_c;
    input signed [31:0] in_port_d;
    // OUT
    output signed [31:0] return_port;
    // Component and signal declarations
    wire signed [31:0] out_cond_expr_FU_32_32_32_32_4_i0_fu_func_33672_33752;
    wire signed [31:0] out_cond_expr_FU_32_32_32_32_4_i1_fu_func_33672_33754;
    wire [2:0] out_const_0;
    wire [3:0] out_const_1;
    wire [4:0] out_const_2;
    wire [2:0] out_const_3;
    wire out_gt_expr_FU_32_0_32_5_i0_fu_func_33672_33740;
    wire out_le_expr_FU_32_0_32_6_i0_fu_func_33672_33742;
    wire out_lut_expr_FU_2_i0_fu_func_33672_33749;
    wire signed [31:0] out_minus_expr_FU_32_32_32_7_i0_fu_func_33672_33704;
    wire signed [31:0] out_plus_expr_FU_32_32_32_8_i0_fu_func_33672_33717;

    constant_value #(BITSIZE_out1(3),
        .value(3'b010)) const_0 (out1(out_const_0));
    constant_value #(BITSIZE_out1(4),
        .value(4'b0100)) const_1 (out1(out_const_1));
    constant_value #(BITSIZE_out1(5),
        .value(5'b01100)) const_2 (out1(out_const_2));
    constant_value #(BITSIZE_out1(3),
        .value(3'b100)) const_3 (out1(out_const_3));
    minus_expr FU #(BITSIZE_in1(32),
        .BITSIZE_in2(32),
        .BITSIZE_out1(32)) fu_func_33672_33704 (out1(out_minus_expr_FU_32_32_32_7_i0_fu_func_33672_33704),
        .in1(in_port_j),
        .in2(in_port_k));
    plus_expr FU #(BITSIZE_in1(32),
        .BITSIZE_in2(32),
        .BITSIZE_out1(32)) fu_func_33672_33717 (out1(out_plus_expr_FU_32_32_32_8_i0_fu_func_33672_33717),
        .in1(in_port_j),
        .in2(in_port_k));
    gt_expr FU #(BITSIZE_in1(32),
        .BITSIZE_in2(3),
        .BITSIZE_out1(1)) fu_func_33672_33740 (out1(out_gt_expr_FU_32_0_32_5_i0_fu_func_33672_33740),
        .in1(in_port_c),
        .in2(out_const_0));
    le_expr FU #(BITSIZE_in1(32),
        .BITSIZE_in2(4),
        .BITSIZE_out1(1)) fu_func_33672_33742 (out1(out_le_expr_FU_32_0_32_6_i0_fu_func_33672_33742),
        .in1(in_port_d),
        .in2(out_const_1));
    lut_expr FU #(BITSIZE_in1(3),
        .BITSIZE_out1(1)) fu_func_33672_33749 (out1(out_lut_expr_FU_2_i0_fu_func_33672_33749),
        .in1(out_const_3),
        .in2(out_gt_expr_FU_32_0_32_5_i0_fu_func_33672_33740),
        .in3(out_le_expr_FU_32_0_32_6_i0_fu_func_33672_33742),
        .in4(1'b0),
        .in5(1'b0),
        .in6(1'b0),
        .in7(1'b0),
        .in8(1'b0),
        .in9(1'b0),
        .in10(1'b0),
        .in11(1'b0),
        .in12(1'b0),
        .in13(1'b0),
        .in14(1'b0),
        .in15(1'b0),
        .in16(1'b0),
        .in17(1'b0),
        .in18(1'b0),
        .in19(1'b0),
        .in20(1'b0),
        .in21(1'b0),
        .in22(1'b0),
        .in23(1'b0),
        .in24(1'b0),
        .in25(1'b0),
        .in26(1'b0),
        .in27(1'b0),
        .in28(1'b0),
        .in29(1'b0),
        .in30(1'b0),
        .in31(1'b0));
endmodule
```

(aryan22154) 192.168.3.57

[VLSI Design Flow 2023 - Mozilla Fi... (aryan22154) 192.168.3.57 - Kons...

datools-server2:3126 (aryan22154) - VNC Viewer

Applications Places Shell - Konsole

(aryan22154) 192.168.3.57 - Konsole

```
File Edit View Bookmarks Settings Help

constant_value #(BITSIZE_out1(3),
    .value(3'b010)) const_0 (out1(out_const_0));
constant_value #(BITSIZE_out1(4),
    .value(4'b0100)) const_1 (out1(out_const_1));
constant_value #(BITSIZE_out1(5),
    .value(5'b01100)) const_2 (out1(out_const_2));
constant_value #(BITSIZE_out1(3),
    .value(3'b100)) const_3 (out1(out_const_3));
minus_expr FU #(BITSIZE_in1(32),
    .BITSIZE_in2(32),
    .BITSIZE_out1(32)) fu_func_33672_33704 (out1(out_minus_expr_FU_32_32_32_7_i0_fu_func_33672_33704),
    .in1(in_port_j),
    .in2(in_port_k));
plus_expr FU #(BITSIZE_in1(32),
    .BITSIZE_in2(32),
    .BITSIZE_out1(32)) fu_func_33672_33717 (out1(out_plus_expr_FU_32_32_32_8_i0_fu_func_33672_33717),
    .in1(in_port_j),
    .in2(in_port_k));
gt_expr FU #(BITSIZE_in1(32),
    .BITSIZE_in2(3),
    .BITSIZE_out1(1)) fu_func_33672_33740 (out1(out_gt_expr_FU_32_0_32_5_i0_fu_func_33672_33740),
    .in1(in_port_c),
    .in2(out_const_0));
le_expr FU #(BITSIZE_in1(32),
    .BITSIZE_in2(4),
    .BITSIZE_out1(1)) fu_func_33672_33742 (out1(out_le_expr_FU_32_0_32_6_i0_fu_func_33672_33742),
    .in1(in_port_d),
    .in2(out_const_1));
lut_expr FU #(BITSIZE_in1(3),
    .BITSIZE_out1(1)) fu_func_33672_33749 (out1(out_lut_expr_FU_2_i0_fu_func_33672_33749),
    .in1(out_const_3),
    .in2(out_gt_expr_FU_32_0_32_5_i0_fu_func_33672_33740),
    .in3(out_le_expr_FU_32_0_32_6_i0_fu_func_33672_33742),
    .in4(1'b0),
    .in5(1'b0),
    .in6(1'b0),
    .in7(1'b0),
    .in8(1'b0),
    .in9(1'b0),
    .in10(1'b0),
    .in11(1'b0),
    .in12(1'b0),
    .in13(1'b0),
    .in14(1'b0),
    .in15(1'b0),
    .in16(1'b0),
    .in17(1'b0),
    .in18(1'b0),
    .in19(1'b0),
    .in20(1'b0),
    .in21(1'b0),
    .in22(1'b0),
    .in23(1'b0),
    .in24(1'b0),
    .in25(1'b0),
    .in26(1'b0),
    .in27(1'b0),
    .in28(1'b0),
    .in29(1'b0),
    .in30(1'b0),
    .in31(1'b0));
endmodule
```

(aryan22154) 192.168.3.57

[VLSI Design Flow 2023 - Mozilla Fi... (aryan22154) 192.168.3.57 - Kons...

```
datools-server2:3126 (aryan22154)) - VNC Viewer
Applications Places Shell - Konsole
(aryan22154) 192.168.3.57 - Konsole
File Edit View Bookmarks Settings Help
.in5(1'b0),
.in6(1'b0),
.in7(1'b0),
.in8(1'b0),
.in9(1'b0));
cond_expr FU #(.BITSIZE_in1(1),
.BITSIZE_in2(32),
.BITSIZE_in3(5),
.BITSIZE_out1(32)) fu_func 33672_33752 (.out1(out_cond_expr_FU_32_32_32_4_i0_fu_func_33672_33752),
.in1(out_gt_expr_FU_32_0_32_5_i0_fu_func_33672_33740),
.in2(out_minus_expr_FU_32_32_32_7_i0_fu_func_33672_33704),
.in3(out_const 2));
cond_expr FU #(.BITSIZE_in1(1),
.BITSIZE_in2(32),
.BITSIZE_in3(32),
.BITSIZE_out1(32)) fu_func 33672_33754 (.out1(out_cond_expr_FU_32_32_32_4_i1_fu_func_33672_33754),
.in1(out_lut_expr_FU_2_i0_fu_func_33672_33749),
.in2(out_plus_expr_FU_32_32_32_8_i0_fu_func_33672_33717),
.in3(out_cond_expr_FU_32_32_32_4_i0_fu_func_33672_33752));
// io-signal post fix
assign return_port = out_cond_expr_FU_32_32_32_4_i1_fu_func_33672_33754;

endmodule

// FSM based controller description for func
// This component has been derived from the input source code and so it does not fall under the copyright of PandA framework, but it follows the input source
code copyright, and may be aggregated with components of the BAMBU/PANDA IP LIBRARY.
// Author(s): Component automatically generated by bambu
// License: THIS COMPONENT IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHAN
TIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
timescale 1ns / 1ps
module controller_func(done_port,
clock,
reset,
start_port);
// IN
input clock;

// OUT
output done_port;
parameter [0:0] S_0 = 1'b1;
reg [0:0] _present_state=S_0, _next_state;
reg done_port;

always @(posedge clock)
if (reset == 1'b0) _present_state <= S_0;
else _present_state <= _next_state;

always @(*)
begin
done_port = 1'b0;
case (_present_state)
S_0 :
if(start_port == 1'b1)
begin
_next_state = S_0;
done_port = 1'b1;
end
else
begin
_next_state = S_0;
end
default :
begin
_next_state = S_0;
end
endcase
end
endmodule

// Top component for func
```

```
datools-server2:3126 (aryan22154)) - VNC Viewer
Applications Places Shell - Konsole
(aryan22154) 192.168.3.57 - Konsole
File Edit View Bookmarks Settings Help
// IN
input clock;
input reset;
input start_port;
// OUT
output done_port;
parameter [0:0] S_0 = 1'b1;
reg [0:0] _present_state=S_0, _next_state;
reg done_port;

always @(posedge clock)
if (reset == 1'b0) _present_state <= S_0;
else _present_state <= _next_state;

always @(*)
begin
done_port = 1'b0;
case (_present_state)
S_0 :
if(start_port == 1'b1)
begin
_next_state = S_0;
done_port = 1'b1;
end
else
begin
_next_state = S_0;
end
default :
begin
_next_state = S_0;
end
endcase
end
endmodule

// Top component for func
```

```
// Top component for func
// This component has been derived from the input source code and so it does not fall under the copyright of PandA framework, but it follows the input source
code copyright, and may be aggregated with components of the BAMBU/PANDA IP LIBRARY.
// Author(s): Component automatically generated by bambu
// License: THIS COMPONENT IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHAN
TIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
timescale 1ns / 1ps
module func(clock,
    reset,
    start_port,
    done_port,
    j,
    k,
    c,
    d,
    return_port);
    // IN
    input clock;
    input reset;
    input start_port;
    input signed [31:0] j;
    input signed [31:0] k;
    input signed [31:0] c;
    input signed [31:0] d;
    // OUT
    output done_port;
    output signed [31:0] return_port;
    // Component and signal declarations

    controller func Controller_i (.done_port(done_port),
        .clock(clock),
        .reset(reset),
        .start_port(start_port));
    datapath func Datapath_i (.return_port(return_port),
        .clock(clock),
        .reset(reset),
        .in_port_j(j),
        .in_port_k(k),
        .in_port_c(c),
        .in_port_d(d));
endmodule
```

(aryan22154) 192.168.3.57

```
.in_port_j(j),
.in_port_k(k),
.in_port_c(c),
.in_port_d(d));

endmodule

// This component is part of the BAMBU/PANDA IP LIBRARY
// Copyright (C) 2004-2022 Politecnico di Milano
// Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
// License: PANDA LGPLv3
timescale 1ns / 1ps
module view_convert_expr_FU(in1,
    out1);
    parameter BITSIZE_in1=1,
        BITSIZE_out1=1;
    // IN
    input signed [BITSIZE_in1-1:0] in1;
    // OUT
    output [BITSIZE_out1-1:0] out1;
    assign out1 = in1;
endmodule

// Minimal interface for function: func
// This component has been derived from the input source code and so it does not fall under the copyright of PandA framework, but it follows the input source
code copyright, and may be aggregated with components of the BAMBU/PANDA IP LIBRARY.
// Author(s): Component automatically generated by bambu
// License: THIS COMPONENT IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHAN
TIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
timescale 1ns / 1ps
module func(clock,
    reset,
    start_port,
    j,
    k,
    c,
    d,
    return_port);
    // IN
    input clock;
    input reset;
    input start_port;
    input signed [31:0] j;
    input signed [31:0] k;
    input signed [31:0] c;
    input signed [31:0] d;
    // OUT
    output done_port;
    output signed [31:0] return_port;
    // Component and signal declarations

    controller func Controller_i (.done_port(done_port),
        .clock(clock),
        .reset(reset),
        .start_port(start_port));
    datapath func Datapath_i (.return_port(return_port),
        .clock(clock),
        .reset(reset),
        .in_port_j(j),
        .in_port_k(k),
        .in_port_c(c),
        .in_port_d(d));
endmodule
```

(aryan22154) 192.168.3.57



```

j,
k,
c,
d,
done_port,
return_port);
// IN
input clock;
input reset;
input start_port;
input [31:0] j;
input [31:0] k;
input [31:0] c;
input [31:0] d;
// OUT
output done_port;
output [31:0] return_port;
// Component and signal declarations
wire signed [31:0] out_return_port_view_convert_expr_FU;

_func _func_i0 (.done_port(done_port),
               .return_port(out_return_port_view_convert_expr_FU),
               .clock(clock),
               .reset(reset),
               .start_port(start_port),
               .j(j),
               .k(k),
               .c(c),
               .d(d));
view convert_expr_FU #(.BITSIZE in1(32),
                      .BITSIZE_out1(32)) return_port_view_convert_expr_FU (.out1(return_port),
                                   .in1(out_return_port_view_convert_expr_FU));

endmodule

```

aryan22154@edaserver3:~/bambu

(aryan22154) 192.168.3.57

[VLSI Design Flow 2023 - Mozilla Fi... (aryan22154) 192.168.3.57 - Kons...

```

Terminal Help  < ->  IITD_MTECH
funcv x
funcv
1 //
2 // Politecnico di Milano
3 // Code created using Panda - Version: Panda 0.9.7 - Revision 8b59b7ac7b9ab30cd20960921ab47ea5009163f1-main - Date 2023-02-07T14:33:39
4 // /tmp/.mount_bambu-0VihYR/usr/bin/bambu executed with: /tmp/.mount_bambu-0VihYR/usr/bin/bambu --top-fname=func a21.c
5 //
6 // Send any bug to: panda-info@polimi.it
7 // *****
8 // The following text holds for all the components tagged with PANDA_LGPLv3.
9 // They are all part of the BAMBU/PANDA IP LIBRARY.
10 // This library is free software; you can redistribute it and/or
11 // modify it under the terms of the GNU Lesser General Public
12 // License as published by the Free Software Foundation; either
13 // version 3 of the License, or (at your option) any later version.
14 //
15 // This library is distributed in the hope that it will be useful,
16 // but WITHOUT ANY WARRANTY; without even the implied warranty of
17 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
18 // Lesser General Public License for more details.
19 //
20 // You should have received a copy of the GNU Lesser General Public
21 // License along with the Panda framework; see the files COPYING.LIB
22 // If not, see <http://www.gnu.org/licenses/>.
23 // *****
24
25 `ifndef __ICARUS__
26   `define _SIM_HAVE_CLOG2
27 `endif
28 `ifdef VERILATOR
29   `define _SIM_HAVE_CLOG2
30 `endif
31 `ifdef MODEL_TECH
32   `define _SIM_HAVE_CLOG2
33 `endif
34 `ifdef VCS
35   `define _SIM_HAVE_CLOG2
36 `endif
37 `ifndef NCVERILOG

```

```
n Terminal Help ← → IIITD_MTECH

func.v x
func.v
37 `ifdef NCVERILOG
38 `define _SIM_HAVE_CLOG2
39 `endif
40 `ifdef XILINX_SIMULATOR
41 `define _SIM_HAVE_CLOG2
42 `endif
43 `ifdef XILINX_ISIM
44 `define _SIM_HAVE_CLOG2
45 `endif
46
47 // This component is part of the BAMBU/PANDA IP LIBRARY
48 // Copyright (C) 2004-2022 Politecnico di Milano
49 // Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>, Christian Pilato <christian.pilato@polimi.it>
50 // License: PANDA_LGPLV3
51 `timescale 1ns / 1ps
52 module constant_value(out1);
53     parameter BITSIZE_out1=1,
54         value=1'b0;
55     // OUT
56     output [BITSIZE_out1-1:0] out1;
57     assign out1 = value;
58 endmodule
59
60 // This component is part of the BAMBU/PANDA IP LIBRARY
61 // Copyright (C) 2016-2022 Politecnico di Milano
62 // Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
63 // License: PANDA_LGPLV3
64 `timescale 1ns / 1ps
65 module lut_expr_FU(in1,
66     in2,
67     in3,
68     in4,
69     in5,
70     in6,
71     in7,
72     in8,
73     in9,
```

Ln 448, Col 10

```
Terminal Help ← → IIITD_MTECH

func.v x
func.v
73     in9,
74     out1);
75     parameter BITSIZE_in1=1,
76         BITSIZE_out1=1;
77     // IN
78     input [BITSIZE_in1-1:0] in1;
79     input in2;
80     input in3;
81     input in4;
82     input in5;
83     input in6;
84     input in7;
85     input in8;
86     input in9;
87     // OUT
88     output [BITSIZE_out1-1:0] out1;
89     reg[7:0] cleaned_in0;
90     wire [7:0] in0;
91     wire[BITSIZE_in1-1:0] shifted_s;
92     assign in0 = {in9, in8, in7, in6, in5, in4, in3, in2};
93     generate
94         genvar i0;
95         for (i0=0; i0<8; i0=i0+1)
96             begin : L0
97                 always @(*)
98                     begin
99                         if (in0[i0] == 1'b1)
100                             cleaned_in0[i0] = 1'b1;
101                         else
102                             cleaned_in0[i0] = 1'b0;
103                     end
104             end
105     endgenerate
106     assign shifted_s = in1 >> cleaned_in0;
107     assign out1[0] = shifted_s[0];
108     generate
109         if(BITSIZE_out1 > 1)
```

```
un Terminal Help ← → IITD_MTECH
func.v
func.v
109     if(BITSIZE_out1 > 1)
110         assign out1[BITSIZE_out1-1:1] = 0;
111     endgenerate
112
113 endmodule
114
115 // This component is part of the BAMBU/PANDA IP LIBRARY
116 // Copyright (C) 2004-2022 Politecnico di Milano
117 // Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
118 // License: PANDA_LGPLv3
119 `timescale 1ns / 1ps
120 module cond_expr_FU(in1,
121     in2,
122     in3,
123     out1);
124     parameter BITSIZE_in1=1,
125         BITSIZE_in2=1,
126         BITSIZE_in3=1,
127         BITSIZE_out1=1;
128     // IN
129     input [BITSIZE_in1-1:0] in1;
130     input signed [BITSIZE_in2-1:0] in2;
131     input signed [BITSIZE_in3-1:0] in3;
132     // OUT
133     output signed [BITSIZE_out1-1:0] out1;
134     assign out1 = in1 != 0 ? in2 : in3;
135 endmodule
136
137 // This component is part of the BAMBU/PANDA IP LIBRARY
138 // Copyright (C) 2004-2022 Politecnico di Milano
139 // Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
140 // License: PANDA_LGPLv3
141 `timescale 1ns / 1ps
142 module gt_expr_FU(in1,
143     in2,
144     out1);
145     parameter BITSIZE_in1=1,
```

```
un Terminal Help ← → IITD_MTECH
func.v
func.v
146     BITSIZE_in2=1,
147     BITSIZE_out1=1;
148     // IN
149     input signed [BITSIZE_in1-1:0] in1;
150     input signed [BITSIZE_in2-1:0] in2;
151     // OUT
152     output [BITSIZE_out1-1:0] out1;
153     assign out1 = in1 > in2;
154 endmodule
155
156 // This component is part of the BAMBU/PANDA IP LIBRARY
157 // Copyright (C) 2004-2022 Politecnico di Milano
158 // Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
159 // License: PANDA_LGPLv3
160 `timescale 1ns / 1ps
161 module le_expr_FU(in1,
162     in2,
163     out1);
164     parameter BITSIZE_in1=1,
165         BITSIZE_in2=1,
166         BITSIZE_out1=1;
167     // IN
168     input signed [BITSIZE_in1-1:0] in1;
169     input signed [BITSIZE_in2-1:0] in2;
170     // OUT
171     output [BITSIZE_out1-1:0] out1;
172     assign out1 = in1 <= in2;
173 endmodule
174
175 // This component is part of the BAMBU/PANDA IP LIBRARY
176 // Copyright (C) 2004-2022 Politecnico di Milano
177 // Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
178 // License: PANDA_LGPLv3
179 `timescale 1ns / 1ps
180 module minus_expr_FU(in1,
181     in2,
182     out1);
```

```
Terminal  Help  ← →  IITD_MTECH

func.v  X
func.v
~~~~~
183     parameter BITSIZE_in1=1,
184           BITSIZE_in2=1,
185           BITSIZE_out1=1;
186     // IN
187     input signed [BITSIZE_in1-1:0] in1;
188     input signed [BITSIZE_in2-1:0] in2;
189     // OUT
190     output signed [BITSIZE_out1-1:0] out1;
191     assign out1 = in1 - in2;
192 endmodule
193
194 // This component is part of the BAMBU/PANDA IP LIBRARY
195 // Copyright (C) 2004-2022 Politecnico di Milano
196 // Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
197 // License: PANDA_LGPLV3
198 `timescale 1ns / 1ps
199 module plus_expr_FU(in1,
200     in2,
201     out1);
202     parameter BITSIZE_in1=1,
203           BITSIZE_in2=1,
204           BITSIZE_out1=1;
205     // IN
206     input signed [BITSIZE_in1-1:0] in1;
207     input signed [BITSIZE_in2-1:0] in2;
208     // OUT
209     output signed [BITSIZE_out1-1:0] out1;
210     assign out1 = in1 + in2;
211 endmodule
212
213 // Datapath RTL description for func
214 // This component has been derived from the input source code and so it does not fall under the copyright of Panda framework, but it follows
215 // Author(s): Component automatically generated by bambu
216 // License: THIS COMPONENT IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARR
217 `timescale 1ns / 1ps
218 module datapath_func(clock,
219     reset,
```

```
Terminal  Help  ← →  IITD_MTECH

func.v  X
func.v
219     reset,
220     in_port_j,
221     in_port_k,
222     in_port_c,
223     in_port_d,
224     return_port);
225     // IN
226     input clock;
227     input reset;
228     input signed [31:0] in_port_j;
229     input signed [31:0] in_port_k;
230     input signed [31:0] in_port_c;
231     input signed [31:0] in_port_d;
232     // OUT
233     output signed [31:0] return_port;
234     // Component and signal declarations
235     wire signed [31:0] out_cond_expr_FU_32_32_32_32_4_i0_fu_func_33672_33752;
236     wire signed [31:0] out_cond_expr_FU_32_32_32_32_4_i1_fu_func_33672_33754;
237     wire [2:0] out_const_0;
238     wire [3:0] out_const_1;
239     wire [4:0] out_const_2;
240     wire [2:0] out_const_3;
241     wire out_gt_expr_FU_32_0_32_5_i0_fu_func_33672_33740;
242     wire out_le_expr_FU_32_0_32_6_i0_fu_func_33672_33742;
243     wire out_lut_expr_FU_2_i0_fu_func_33672_33749;
244     wire signed [31:0] out_minus_expr_FU_32_32_32_7_i0_fu_func_33672_33704;
245     wire signed [31:0] out_plus_expr_FU_32_32_32_8_i0_fu_func_33672_33717;
246
247     constant_value #(.BITSIZE_out1(3),
248         .value(3'b010)) const_0 (.out1(out_const_0));
249     constant_value #(.BITSIZE_out1(4),
250         .value(4'b0100)) const_1 (.out1(out_const_1));
251     constant_value #(.BITSIZE_out1(5),
252         .value(5'b01100)) const_2 (.out1(out_const_2));
253     constant_value #(.BITSIZE_out1(3),
254         .value(3'b100)) const_3 (.out1(out_const_3));
255     minus_expr_FU #(.BITSIZE_in1(32),
```

```
Terminal Help  ← →  IIITD_MTECH

func.v  X
func.v
256     .BITSIZE_in2(32),
257     .BITSIZE_out1(32)) fu_func_33672_33704 (.out1(out_minus_expr_FU_32_32_32_7_i0_fu_func_33672_33704),
258     .in1(in_port_j),
259     .in2(in_port_k));
260 plus_expr_FU #(.BITSIZE_in1(32),
261     .BITSIZE_in2(32),
262     .BITSIZE_out1(32)) fu_func_33672_33717 (.out1(out_plus_expr_FU_32_32_32_8_i0_fu_func_33672_33717),
263     .in1(in_port_j),
264     .in2(in_port_k));
265 gt_expr_FU #(.BITSIZE_in1(32),
266     .BITSIZE_in2(3),
267     .BITSIZE_out1(1)) fu_func_33672_33740 (.out1(out_gt_expr_FU_32_0_32_5_i0_fu_func_33672_33740),
268     .in1(in_port_c),
269     .in2(out_const_0));
270 le_expr_FU #(.BITSIZE_in1(32),
271     .BITSIZE_in2(4),
272     .BITSIZE_out1(1)) fu_func_33672_33742 (.out1(out_le_expr_FU_32_0_32_6_i0_fu_func_33672_33742),
273     .in1(in_port_d),
274     .in2(out_const_1));
275 lut_expr_FU #(.BITSIZE_in1(3),
276     .BITSIZE_out1(1)) fu_func_33672_33749 (.out1(out_lut_expr_FU_2_i0_fu_func_33672_33749),
277     .in1(out_const_3),
278     .in2(out_gt_expr_FU_32_0_32_5_i0_fu_func_33672_33740),
279     .in3(out_le_expr_FU_32_0_32_6_i0_fu_func_33672_33742),
280     .in4(1'b0),
281     .in5(1'b0),
282     .in6(1'b0),
283     .in7(1'b0),
284     .in8(1'b0),
285     .in9(1'b0));
286 cond_expr_FU #(.BITSIZE_in1(1),
287     .BITSIZE_in2(32),
288     .BITSIZE_in3(5),
289     .BITSIZE_out1(32)) fu_func_33672_33752 (.out1(out_cond_expr_FU_32_32_32_32_4_i0_fu_func_33672_33752),
290     .in1(out_gt_expr_FU_32_0_32_5_i0_fu_func_33672_33740),
291     .in2(out_minus_expr_FU_32_32_32_7_i0_fu_func_33672_33704),
292     .in3(out_const_2));
```

```
Terminal Help  ← →  IIITD_MTECH

func.v  X
func.v
293     cond_expr_FU #(.BITSIZE_in1(1),
294     .BITSIZE_in2(32),
295     .BITSIZE_in3(32),
296     .BITSIZE_out1(32)) fu_func_33672_33754 (.out1(out_cond_expr_FU_32_32_32_32_4_i1_fu_func_33672_33754),
297     .in1(out_lut_expr_FU_2_i0_fu_func_33672_33749),
298     .in2(out_plus_expr_FU_32_32_32_8_i0_fu_func_33672_33717),
299     .in3(out_cond_expr_FU_32_32_32_32_4_i0_fu_func_33672_33752));
300 // io-signal post fix
301 assign return_port = out_cond_expr_FU_32_32_32_32_4_i1_fu_func_33672_33754;
302
303 endmodule
304
305 // FSM based controller description for func
306 // This component has been derived from the input source code and so it does not fall under the copyright of PandA framework, but it follows
307 // Author(s): Component automatically generated by bambu
308 // License: THIS COMPONENT IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARR
309 `timescale 1ns / 1ps
310 module controller_func(done_port,
311     clock,
312     reset,
313     start_port);
314 // IN
315 input clock;
316 input reset;
317 input start_port;
318 // OUT
319 output done_port;
320 parameter [0:0] S_0 = 1'b1;
321 reg [0:0] _present_state=S_0, _next_state;
322 reg done_port;
323
324 always @(posedge clock)
325     if (reset == 1'b0) _present_state <= S_0;
326     else _present_state <= _next_state;
327
328 always @(*)
329     begin
```

```
Terminal  Help  ← →  IIIITD_MTECH

func.v  X
func.v
329  begin
330  done_port = 1'b0;
331  case (_present_state)
332  S_0 :
333      if(start_port == 1'b1)
334      begin
335          _next_state = S_0;
336          done_port = 1'b1;
337      end
338      else
339      begin
340          _next_state = S_0;
341      end
342  default :
343      begin
344          _next_state = S_0;
345      end
346  endcase
347  end
348  endmodule
349
350  // Top component for func
351  // This component has been derived from the input source code and so it does not fall under the copyright of Panda framework, but it follows
352  // Author(s): Component automatically generated by bambu
353  // License: THIS COMPONENT IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARF
354  `timescale 1ns / 1ps
355  module _func(clock,
356      reset,
357      start_port,
358      done_port,
359      j,
360      k,
361      c,
362      d,
363      return_port);
364      // IN
365      input clock;
```

```
Terminal  Help  ← →  IIIITD_MTECH

func.v  X
func.v
366  input reset;
367  input start_port;
368  input signed [31:0] j;
369  input signed [31:0] k;
370  input signed [31:0] c;
371  input signed [31:0] d;
372  // OUT
373  output done_port;
374  output signed [31:0] return_port;
375  // Component and signal declarations
376
377  controller_func Controller_i (done_port(done_port),
378      .clock(clock),
379      .reset(reset),
380      .start_port(start_port));
381  datapath_func Datapath_i (return_port(return_port),
382      .clock(clock),
383      .reset(reset),
384      .in_port_j(j),
385      .in_port_k(k),
386      .in_port_c(c),
387      .in_port_d(d));
388
389  endmodule
390
391  // This component is part of the BAMBU/PANDA IP LIBRARY
392  // Copyright (C) 2004-2022 Politecnico di Milano
393  // Author(s): Fabrizio Ferrandi <fabrizio.ferrandi@polimi.it>
394  // License: PANDA_LGPLv3
395  `timescale 1ns / 1ps
396  module view_convert_expr_FU(in1,
397      out1);
398      parameter BITSIZE_in1=1,
399          BITSIZE_out1=1;
400      // IN
401      input signed [BITSIZE_in1-1:0] in1;
402      // OUT
```

```
Terminal Help  ← →  IIITD_MTECH  08 --
func.v x
func.v
403     output [BITSIZE_out1-1:0] out1;
404     assign out1 = in1;
405 endmodule
406
407 // Minimal interface for function: func
408 // This component has been derived from the input source code and so it does not fall under the copyright of PandA framework, but it follows
409 // Author(s): Component automatically generated by bambu
410 // License: THIS COMPONENT IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARR
411 `timescale 1ns / 1ps
412 module func(clock,
413     reset,
414     start_port,
415     j,
416     k,
417     c,
418     d,
419     done_port,
420     return_port);
421     // IN
422     input clock;
423     input reset;
424     input start_port;
425     input [31:0] j;
426     input [31:0] k;
427     input [31:0] c;
428     input [31:0] d;
429     // OUT
430     output done_port;
431     output [31:0] return_port;
432     // Component and signal declarations
433     wire signed [31:0] out_return_port_view_convert_expr_FU;
434
435     _func_func_i0 (.done_port(done_port),
436         .return_port(out_return_port_view_convert_expr_FU),
437         .clock(clock),
438         .reset(reset),
439         .start_port(start_port),
```

```
Terminal Help  ← →  IIITD_MTECH
func.v x
func.v
417     c,
418     d,
419     done_port,
420     return_port);
421     // IN
422     input clock;
423     input reset;
424     input start_port;
425     input [31:0] j;
426     input [31:0] k;
427     input [31:0] c;
428     input [31:0] d;
429     // OUT
430     output done_port;
431     output [31:0] return_port;
432     // Component and signal declarations
433     wire signed [31:0] out_return_port_view_convert_expr_FU;
434
435     _func_func_i0 (.done_port(done_port),
436         .return_port(out_return_port_view_convert_expr_FU),
437         .clock(clock),
438         .reset(reset),
439         .start_port(start_port),
440         .j(j),
441         .k(k),
442         .c(c),
443         .d(d));
444     view_convert_expr_FU #(.BITSIZE_in1(32),
445         .BITSIZE_out1(32)) return_port_view_convert_expr_FU (.out1(return_port),
446         .in1(out_return_port_view_convert_expr_FU));
447
448 endmodule
```

Q3) - Explain the role of each module in the generated RTL.

The FSM here contains a single state  $S_0=1'b1$ .

And if the reset  $=1'b0$ , then the  $\_present\_state \leq S_0$ . Otherwise, the  $\_present\_state \leq next\_state$ .

Initially, the  $done\_port=1'b0$ .

```
case (_present_state)
  S_0 :
    if(start_port == 1'b1)
      begin
        _next_state = S_0;
        done_port = 1'b1;
      end
    else
      begin
        _next_state = S_0;
      end
  default :
    begin
      _next_state = S_0;
    end
endcase
```

Top module:  $\_func \rightarrow$  data path  $\rightarrow$  Controller  $\rightarrow$  helper modules to operate.

1)- module constant\_value () – This component is part of the BAMBU/PANDA IP LIBRARY. it generates the constant value with a size of [BITSIZE\_out1-1:0].

2)- module lut\_expr\_FU () – This component is part of the BAMBU/PANDA IP LIBRARY. This module takes the 9 inputs and gives the 1 output. The output depends on the following factors using the loop and the output is the shifted + several values. (If the BITSIZE\_out1 >1).

3)- module cond\_expr\_FU () – This component is part of the BAMBU/PANDA IP LIBRARY. This component is part of the BAMBU/PANDA IP LIBRARY. This module performs the conditional operation.

assign out1 = in1! = 0? in2: in3.

4)- module gt\_expr\_FU () – This component is part of the BAMBU/PANDA IP LIBRARY. This module performs the greater operation.

assign out1 = in1 > in2.

5)- module le\_expr\_FU () – This component is part of the BAMBU/PANDA IP LIBRARY. This is the module assign out1 = in1 <= in2.

6)- module minus\_expr\_FU () – This component is part of the BAMBU/PANDA IP LIBRARY. This module performs the minus operation between two inputs. assign out1 = in1 - in2.



7)- module plus\_expr\_FU () – This component is part of the BAMBU/PANDA IP LIBRARY. This module performs the plus operation between two inputs. Assign  $out1 = in1 + in2$ .

8)- module datapath\_func () –Datapath RTL description for func.This module calls the other function with the parameters and is responsible for handling the Datapath of func function of c.

9)- module controller\_func () – This module deals with the FSM based controller description for func.

10)- module \_func () –Top component for func.

11)- module view\_convert\_expr\_FU () – This component is part of the BAMBU/PANDA IP LIBRARY.To view the convert, assign  $out1 = in1$ .

12)- module func() - Minimal interface for function: func.

Q4) - Explain how + and – are computed in the data path in the generated Verilog model.

The module plus\_expr\_FU and module minus\_expr\_FU perform the + and the – operation in the Verilog model.

Plus\_expr\_FU - assign  $out1 = in1 + in2$ .

Minus\_expr\_FU - assign  $out1 = in1 - in2$ .

So, now let's take an example from the code in the datapath

```
minus_expr_FU #(.BITSIZE_in1(32),
    .BITSIZE_in2(32),
    .BITSIZE_out1(32)) fu_func_33672_33704
(.out1(out_minus_expr_FU_32_32_32_7_i0_fu_func_33672_33704),
    .in1(in_port_j),
    .in2(in_port_k));
plus_expr_FU #(.BITSIZE_in1(32),
    .BITSIZE_in2(32),
    .BITSIZE_out1(32)) fu_func_33672_33717
(.out1(out_plus_expr_FU_32_32_32_8_i0_fu_func_33672_33717),
    .in1(in_port_j),
    .in2(in_port_k));
```

This is the code snippet of the datapath in the module.

So, the data path will call these modules when + and – are needed to perform with the parameters.