

3.1 Bus Operation

```

18 # ----- #
19 #
20 # Quartus Prime
21 # Version 16.1.2 Build 203 01/16/2017 SJ Standard Edition
22 # Date created = 14:04:54 December 03, 2020
23 #
24 # ----- #
25 #
26 # Notes:
27 #
28 # 1) The default values for assignments are stored in the file:
29 #     playbus0_assignment_defaults.qdf
30 #     If this file doesn't exist, see file:
31 #     assignment_defaults.qdf
32 #
33 # 2) Altera recommends that you do not modify this file. This
34 #     file is updated automatically by the Quartus Prime software
35 #     and any changes you make may be lost or overwritten.
36 #
37 # ----- #
38 #
39 #
40 set_global_assignment -name FAMILY "Cyclone V"
41 set_global_assignment -name DEVICE 5CSEMA5F31C6
42 set_global_assignment -name TOP_LEVEL_ENTITY playbus0
43 set_global_assignment -name ORIGINAL_QUARTUS_VERSION 16.1.2
44 set_global_assignment -name PROJECT_CREATION_TIME_DATE "14:04:54 DECEMBER 03, 2020"
45 set_global_assignment -name LAST_QUARTUS_VERSION "16.1.2 Standard Edition"
46 set_global_assignment -name SYSTEMVERILOG_FILE playbus0.sv
47 set_global_assignment -name SYSTEMVERILOG_FILE components.sv
48 set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output_files
49 set_global_assignment -name MIN_CORE_JUNCTION_TEMP 0
50 set_global_assignment -name MAX_CORE_JUNCTION_TEMP 85
51 set_global_assignment -name ERROR_CHECK_FREQUENCY_DIVISOR 256
52
53
54 set_location_assignment PIN_AE26 -to disp0[0]
55 set_location_assignment PIN_AE27 -to disp0[1]
56 set_location_assignment PIN_AE28 -to disp0[2]
57 set_location_assignment PIN_AG27 -to disp0[3]
58 set_location_assignment PIN_AF28 -to disp0[4]
59 set_location_assignment PIN_AG28 -to disp0[5]
60 set_location_assignment PIN_AH28 -to disp0[6]
61 set_location_assignment PIN_AJ29 -to displ[0]
62 set_location_assignment PIN_AH29 -to displ[1]
63 set_location_assignment PIN_AH30 -to displ[2]
64 set_location_assignment PIN_AG30 -to displ[3]
65 set_location_assignment PIN_AF29 -to displ[4]
66 set_location_assignment PIN_AF30 -to displ[5]
67 set_location_assignment PIN_AD27 -to displ[6]
68
69 set_location_assignment PIN_AB12 -to sw0[0]
70 set_location_assignment PIN_AC12 -to sw0[1]
71 set_location_assignment PIN_AF9 -to sw0[2]
72 set_location_assignment PIN_AF10 -to sw0[3]
73 set_location_assignment PIN_AD11 -to RAMO
74 set_location_assignment PIN_AD12 -to ROMO
75 set_location_assignment PIN_AE11 -to SWBEN
76 set_location_assignment PIN_AC9 -to RAMW
77 set_location_assignment PIN_AD10 -to LEDLTCH
78 set_location_assignment PIN_Y16 -to n_clk
79 set_instance_assignment -name PARTITION_HIERARCHY root_partition -to | -section_id Top
80 set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id Top
81 set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL PLACEMENT_AND_ROUTING -section_id Top
82 set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top

```

Diagram of Code with pin configuration added on

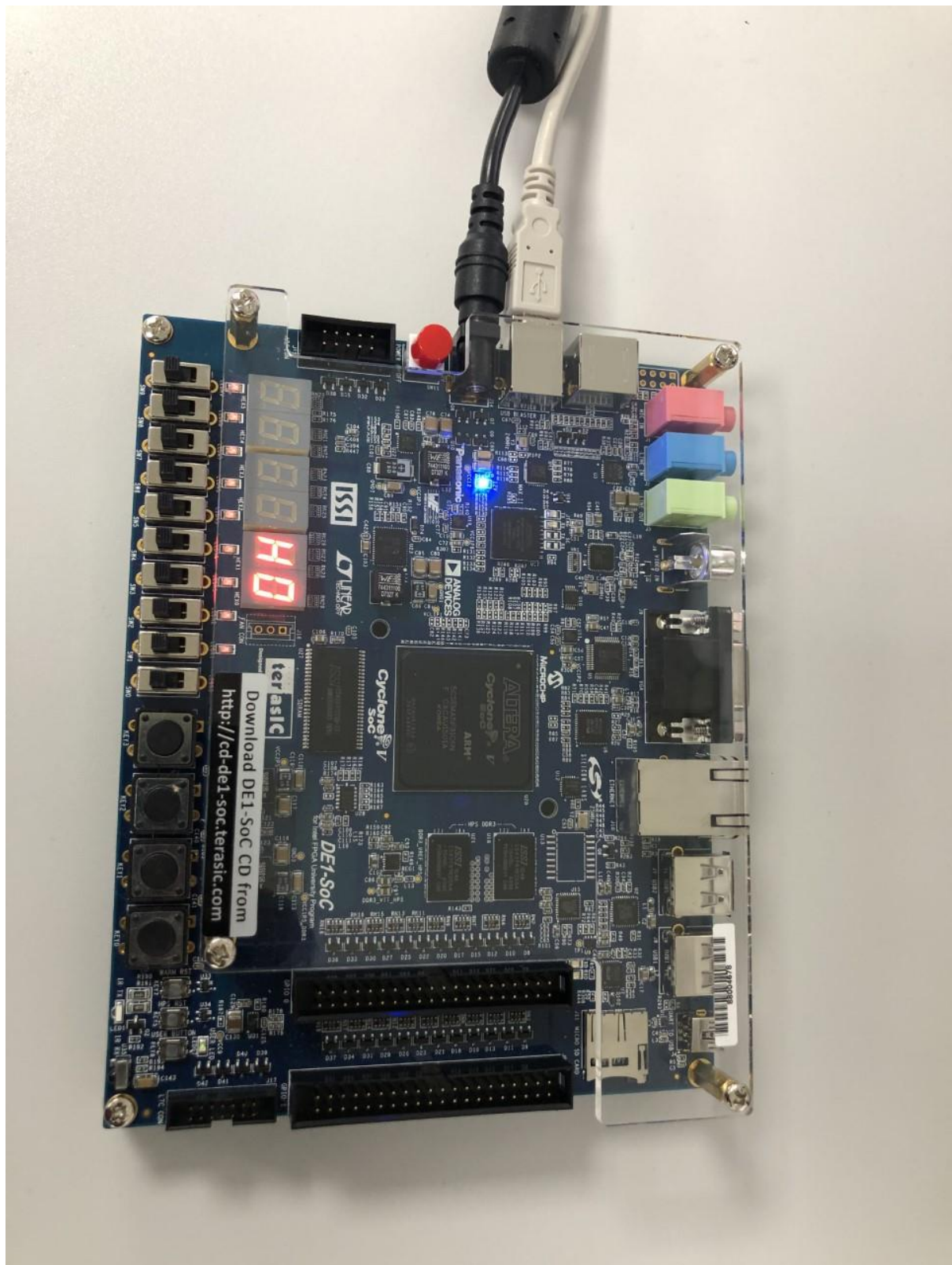


Diagram of FPGA with 'HO' displayed

Hardware Setup... DE-SoC [USB-1] Mode: JTAG Progress: 100% (Successful)

☐ Enable real-time ISP to allow background programming when available

Start Stop Auto Detect Delete Add File... Change File... Save File Add Device... Up Down

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
<none>	SOCVHPS	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H:/t4/playbus0/Files/p...	5CSEMA5F31	00AFE608	00AFE608	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Diagram illustrating the JTAG connection between the SOC VHPS and the 5CSEMA5F31 device:

```

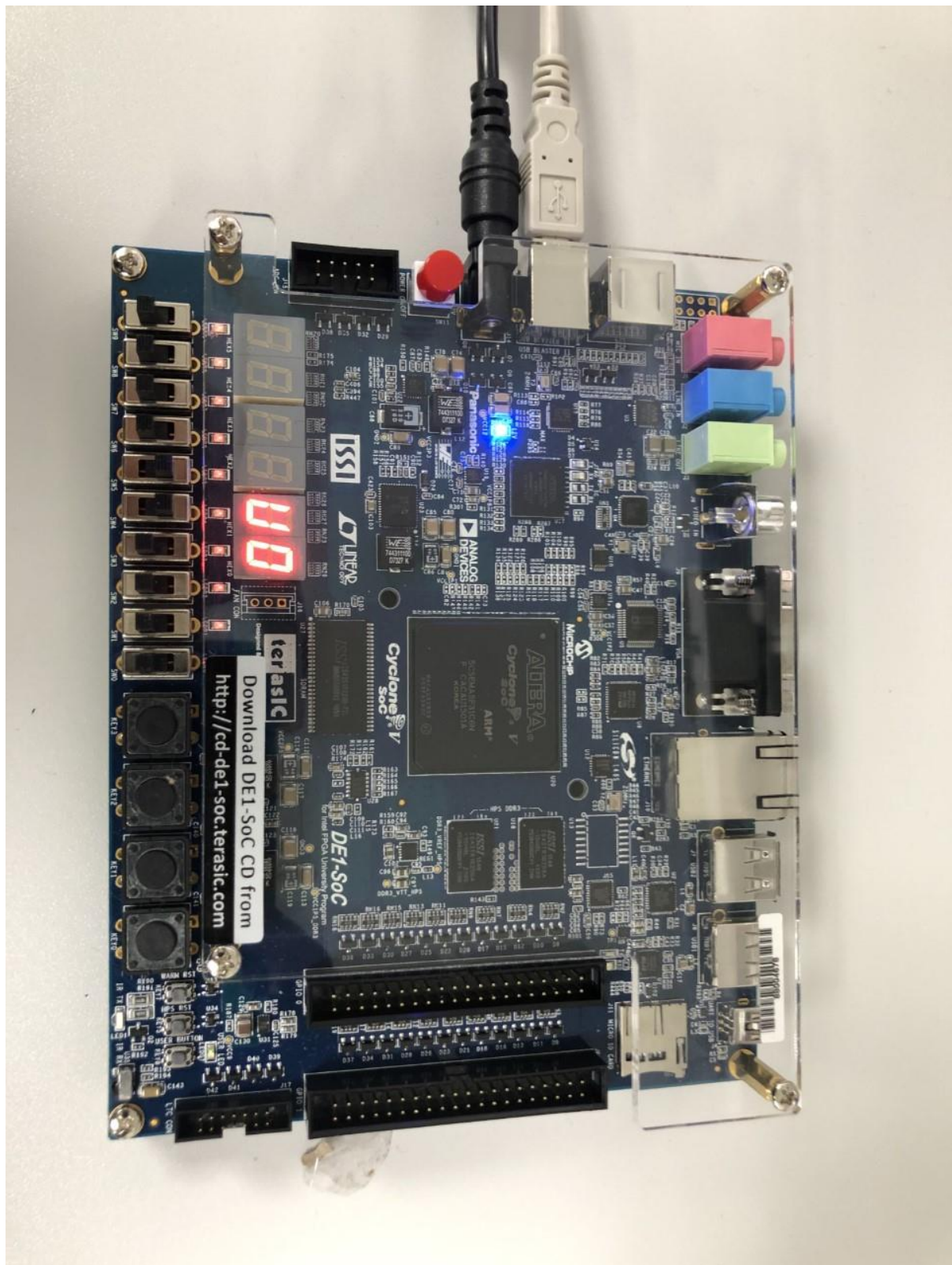
graph LR
    TDI --> SOCVHPS
    SOCVHPS --> 5CSEMA5F31
    5CSEMA5F31 -- TDO --> TDO
  
```

Messages:

Type	ID	Message
Info	209061	Ended Programmer operation at Thu Dec 03 14:09:23 2020

System (6) Processing

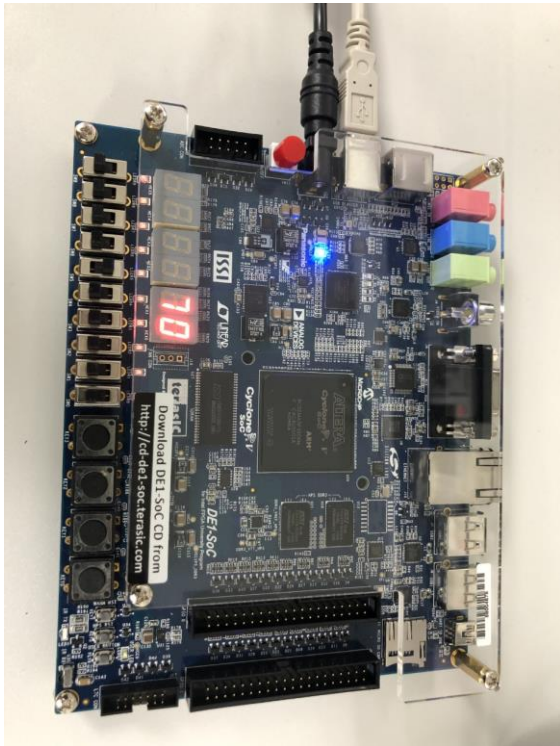
Diagram of Quartus Programmer with the code written to FPGA



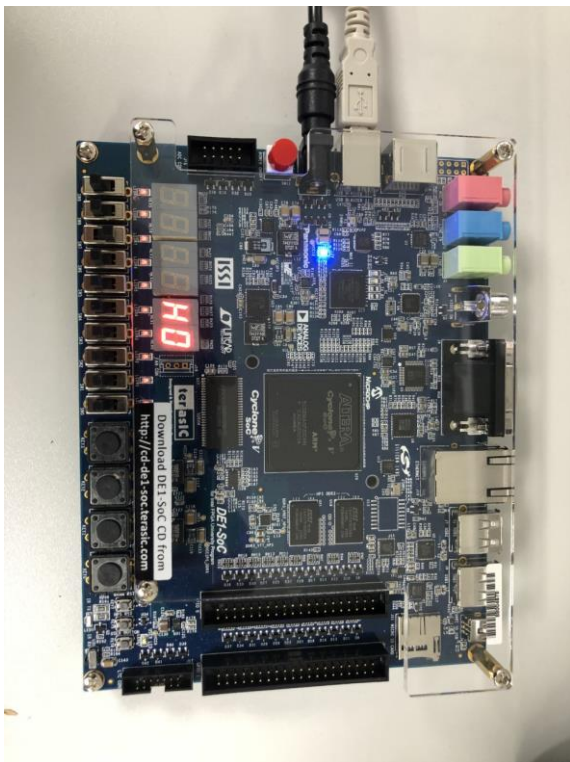
All the operations listed in TABLE 3 work correctly. I verified this by using the switches on the board SW0 – SW9 to copy data and read data.

Operation

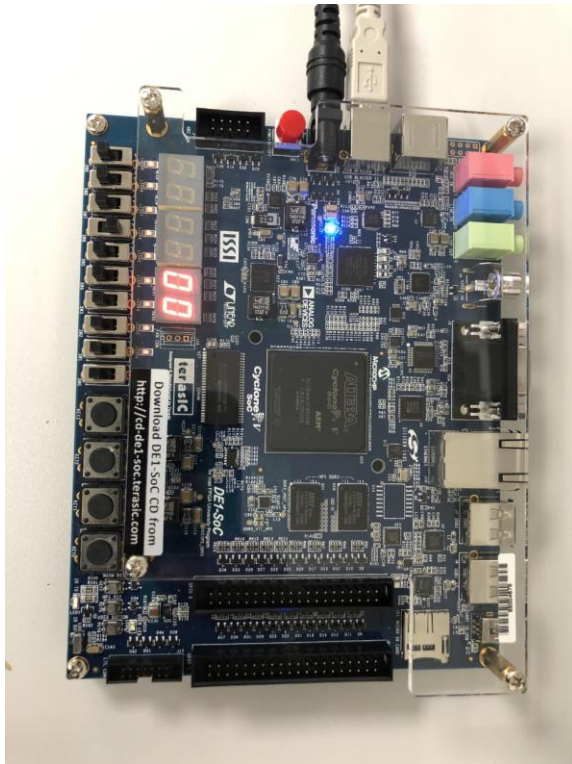
Read data from EPROM at address 5 onto bus - By flipping ROMO on the FPGA as shown below



Read data from RAM at address 5 onto bus - By flipping RAMW on the FPGA as shown below



Read data from Switches onto bus - By flipping SWBEN on the FPGA as shown below



Copy data from Switches into RAM at address 5 – By flipping on SWBEN and then RAMW. Then using RAMO to view the data

Copy data from EPROM into RAM at address 5 - By flipping on ROMO and then RAMW. Then using RAMO to view the data

Copy data from Switches into LEDs - By flipping on SWBEN and then LEDLTCH.

Copy data from EPROM at address into LEDs - By flipping on ROMO and then LEDLTCH.

Copy data from RAM at address into LEDs - By flipping on RAMW and then LEDLTCH.

Q: How does the FPGA system determine when the data bus is in the 'Z' state?

The Z state is represented by a 'H' on the FPGA.

Q: SystemVerilog and ModelSim can represent 'Z' and 'X' states, but are these states “real”? In other words, is it possible to design logic that would detect 'Z' or 'X' states and show them on a 7-segment display?

The states are not real and merely a simulation of the states represented on the 7 segment display. A 'H' state would not be possible as it involves taking the FPGA to either low nor high state. This is not possible as resistors will draw current.

3.2 Controller Operation

Hardware Setup... DE-SoC [USB-1] Mode: JTAG Progress: 100% (Successful)

☐ Enable real-time ISP to allow background programming when available

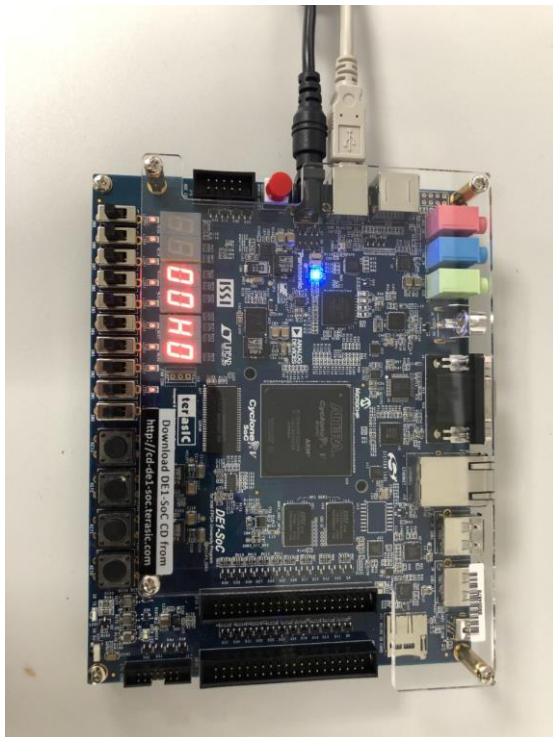
Start Stop Auto Detect Delete Add File... Change File... Save File Add Device... Up Down

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
<none>	SOCVHPS	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H:/t4/playbus1/output...	5CSEMA5F31	00B04262	00B04262	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

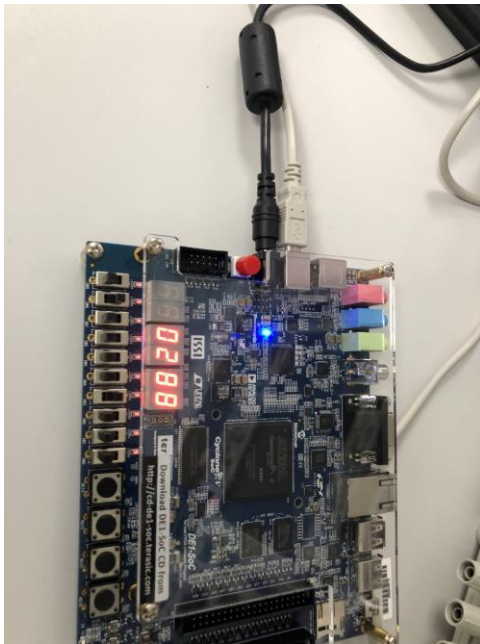
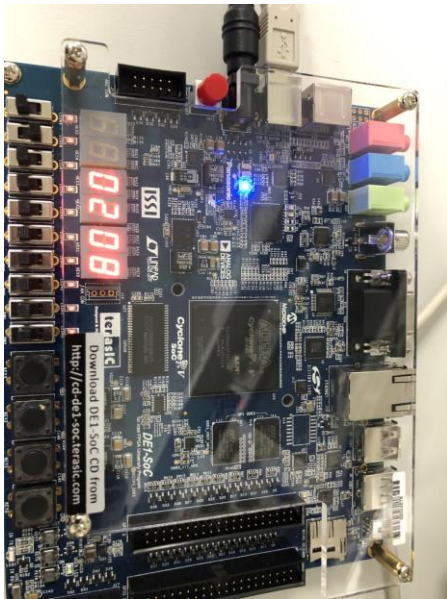
Diagram illustrating the JTAG connection between the DE-SoC (SOCVHPS) and the target device (5CSEMA5F31). The TDI (Test Data In) signal is connected to the DE-SoC, and the TDO (Test Data Out) signal is connected to the target device.

Log messages:

Type	ID	Message
Info	209061	Ended Programmer operation at Thu Dec 03 15:25:46 2020



Function 2: Read data from Switches onto bus - Flip the switches in the switches section of the board



Function 5: Before –



After implemented –



Q: Are the signals that you observe consistent with the waveforms from your simulation? If not, why not?

Yes they are.

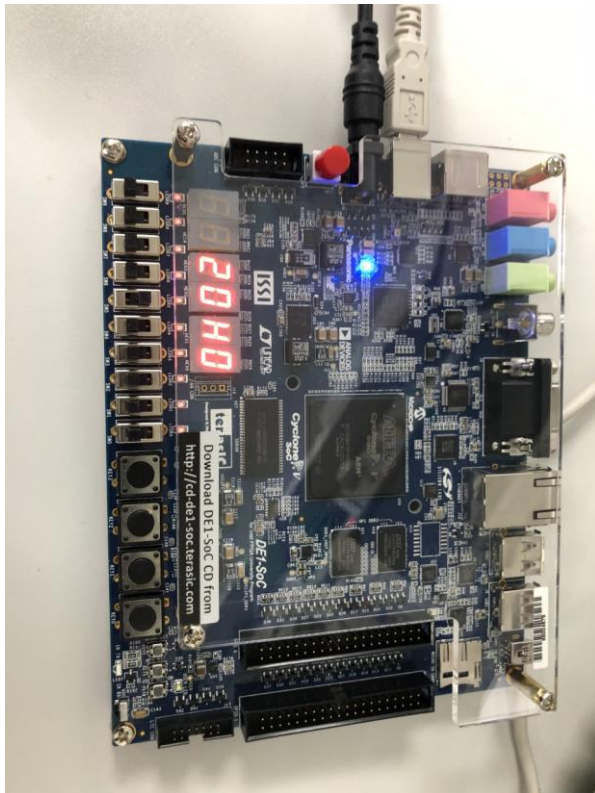
3.3

Q: Why is it a good idea to simulate code before downloading it onto the FPGA? Hint: How easy is it to observe what is happening inside your design on the FPGA compared with in the simulator

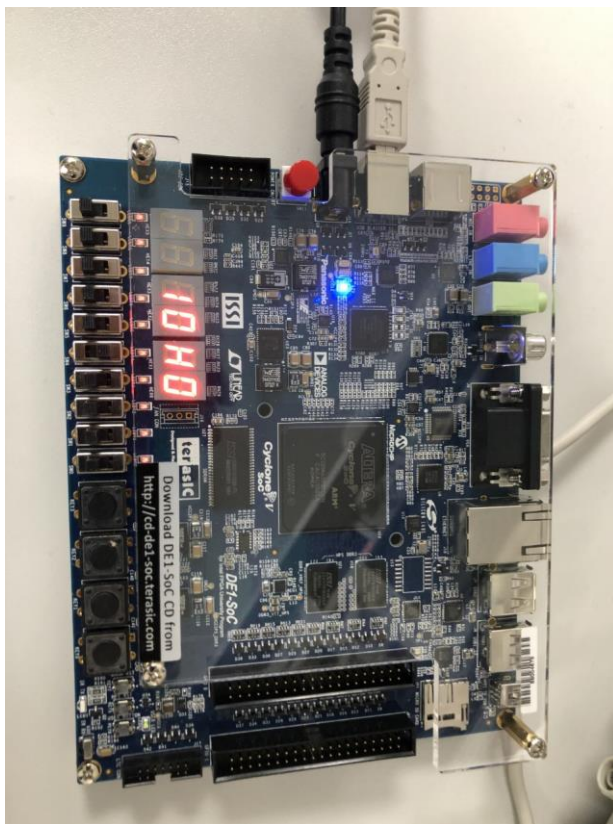
The simulation enables me to see the waveforms and actual working of the functions which will be harder to see on the FPGA

Operation

Read data from EPROM at address onto bus

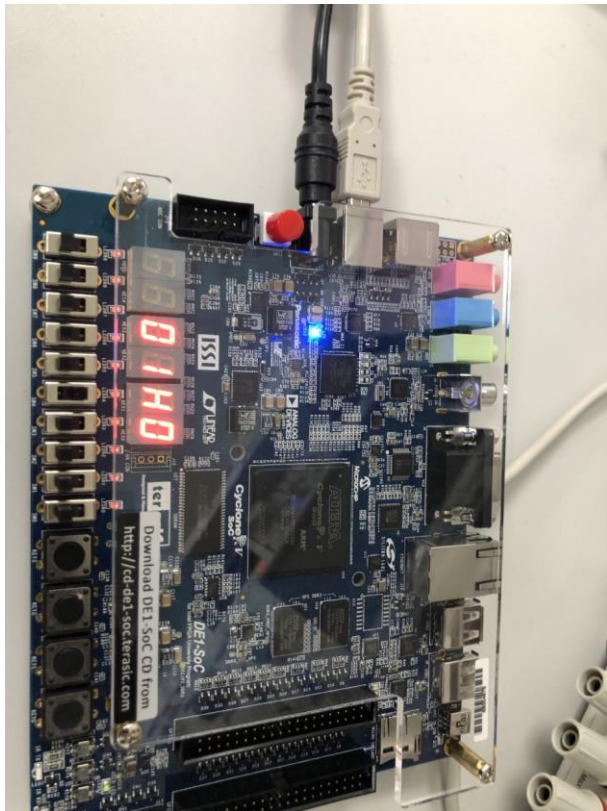


Read data from RAM at address onto bus

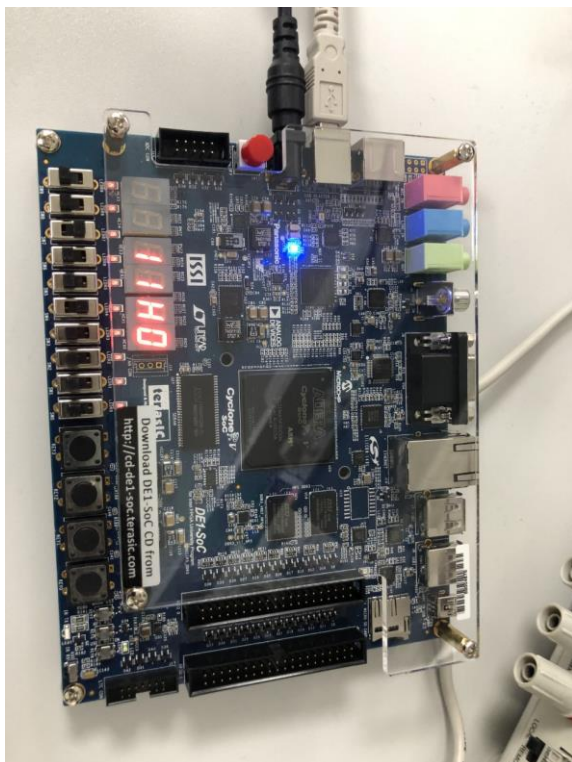


Copy data from Switches into RAM at address

Switches before-



Switches after-



Copy data from EPROM into RAM at address



- RAM before



- Data at EPROM



Copy data from EPROM at address into LEDs

Copy data from RAM at address into LEDs

Q: For each of the functions, is the observed behaviour consistent with the simulations? If not, why not?

Yes

Q: For these new functions, can you read from and write to different addresses?

Yes, as shown above