

Lab no 02: Simulate 2-to-1 Multiplexer & Implement 7-Segment Decoder

The purpose of this Lab is to learn to:

- Simulate 2-to-1 multiplexer on ModelSim and verify multiplexer function using testbench. In this lab, you will build the multiplexer using logic gates (refer to Lab 01).
- Implement the seven-segment decoder on FPGA. You will write the logic equation for each segment using Verilog bitwise operators.

Objective: a seven-segment run on FPGA (<u>Here</u>).

Refer to assignment 2, to review the logic equations of the seven-segment decoder.

Parts: -

- 1. Simulate 2-1 multiplexer using logic gates.
- 2. Implement the seven-segment decoder using logic equations and run it on the FPGA.



Part 1. Simulate 2-1 multiplexer using logic gates (Lab01).



The multiplexer chooses between the two data inputs based on the select:

if S = 0, Y = D0, and if S = 1, Y = D1.

Verilog code for multiplexer 2-to-1

```
// -----// Mux 2-1 -----//
module mux 2 1(S,D0,D1,Y);
input S, D0, D1;
output Y;
wire not S, out G2, out G3;
notGate G1(S, not S);
andGate G2(D0, not S, out G2);
andGate G3(D1, S, out G3);
orGate G4(out G2, out G3, Y);
endmodule
// ----- AND Gate -----
                                              -//
module andGate (a,b,c);
input a, b;
output c;
assign c = a \& b;
endmodule
```

Faculty of Computers and Artificial Intelligence

CS222: Computer Architecture



```
// ----- Or Gate -----//
module orGate (a,b,c);
input a, b;
output c;
assign c = a | b;
endmodule
// ----- Not Gate -----//
module notGate (a,c);
input a;
output c;
assign c = ~a;
endmodule
```

Testbench of the multiplexer 2-to-1

```
module mux_tb;
reg s, d0, d1;
wire y;
mux_2_1 mux_dut(s,d0,d1,y);
initial
begin
s= 0; d0 = 0; d1 =0;
#10 s=0; d0=0; d1=1; // select d0=0
#10 s=0; d0=1; d1=0; // select d0=1
#10 s=1; d0=0; d1=1; // select d1=1
#10 s=1; d0=1; d1=0; // select d1=0
end
endmodule
```



Part 2. Implement the seven-segment decoder using logic equations.

A seven-segment display decoder takes a 4-bit data input A, B, C, D and produces seven outputs to control light-emitting diodes to display a digit from 0 to 9. The seven outputs are often called segments a through g, as defined in Figure.



The truth table of the seven-segment decoder:

Digit	A	В	С	D	а	b	с	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0



The logic equations of the seven-segment decoder

 $a = A + C + BD + \overline{B} \overline{D}$ $b = \overline{B} + \overline{C} \overline{D} + CD$ $c = B + \overline{C} + D$ $d = \overline{B} \overline{D} + C \overline{D} + B \overline{C} D + \overline{B} C + A$ $e = \overline{B} \overline{D} + C \overline{D}$ $f = A + \overline{C} \overline{D} + B \overline{C} + B \overline{D}$

 $\mathbf{g} = \mathbf{A} + \mathbf{B} \ \overline{\mathbf{C}} + \overline{\mathbf{B}} \ \mathbf{C} + \mathbf{C} \ \overline{\mathbf{D}}$

• Bitwise operators

&	Bitwise and / reduction and
1	Bitwise or / reduction or
^	Bitwise xor / reduction xor
~	Bitwise not



Verilog code for Decoder to 7 segments

```
module decoder 7seg (A, B, C, D, led a, led b, led c,
led d, led e, led f, led g);
       input A, B, C, D;
       output led a, led b, led c, led d, led e, led f,
                 led g;
       assign led a = \sim (A \mid C \mid B\&D \mid \sim B\&\sim D);
       assign led b = \sim (\sim B \mid \sim C\&\sim D \mid C\&D);
       assign led c = \sim (B | \sim C | D);
       assign led d = \sim (\simB&\simD | C&\simD | B&\simC&D | \simB&C |A);
       assign led e = \sim (\sim B\&\sim D | C\&\sim D);
       assign led f = ~(A | ~C\&~D | B\&~C | B\&~D);
       assign led q = \sim (A \mid B\&\sim C \mid \sim B\&C \mid C\&\sim D);
```

endmodule

Run the seven-segment decoder on FPGA.

We will use a DE-10lite kit, Altera MAX 10 based FPGA board, check here. Check DE10-lite user manual (Here). You will use **Quartus** to program the FPGA.



Quartus – Seven Segment Decoder Project Steps

Step 1: Open Quartus.



Step 2: Open a New Project Wizard.

* N	ew Project Wizard	Oper	n Project
Compare Editions	Buy Software	Documentation	
Training	Support	What's New	
Notifications			
Close page after p	roject load reen again		(intel)



Step 3: Select Next

The No	lew Project Wizard helps you create a new project and preliminary project settings, including the ring:
•	Project name and directory Name of the top-level design entity Project files and libraries Target device family and device EDA tool settings
Cotting	are command (Assignments menu). You can use the various page of the Settings dialogs but the
Setting	an endinge and existing in outsing project and specify additional project wide settings with the gis command (Assignments menu). You can use the various pages of the Settings dialog box to add onality to the project.

Step 4: Choose a directory to put your project under. you can place it wherever you want. Name the project as the name of the top-level module. You will name it <u>"decoder_7seg"</u>, Select Next.

What is the name of	f this project?		
what is the name o	this project		
What is the name o match the entity na	the top-level design entity for this project me in the design file.	ct? This name is case	sensitive and must ex
blink			
Lice Existing Project	t Settings		
Use Existing Project			
Use Existing Project			
Use Existing Projec	and the second	station as	-
Use Existing Project	and an and and	antes an annes a	for many
Use Existing Project	and an and an and an	antes an antes of	par mana



Step5: Select Empty Project, and then click Next.

Project Type
Select the type of project to create.
Empty project
Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.
○ Project template
Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the <u>Design Store</u> .
and for her of the same for a grand the same
< Back Next > Finish Cancel Help

Step 6: If You want to add any files here. <u>Add the seven-</u> segment decoder Verilog file And Click Next.

Select the design files you want to include in directory to the project.	n the project. Click Add All to add all design files in the	proje
Note: you can always add design files to the	project later.	
<u>F</u> ile name:		Α
	×	Ad
File Name	Type Library Design Entry/Synthesis Toc	Ren
D:/BFCAI/Architecture/code/	Ve	
		D
		Pror
		(1 , 0)
•	► •	



Step7: Board Settings, select our board <u>DE10-lite</u>, <u>unmark</u> <u>"create top-level design file"</u>

electi	he board/devel	opment kit you v	want to target for (compila	ation.		\checkmark		
amily:	MAX 10			-	Development Kit:	MAX 10 DE10	- Lite		•
<u>v</u> ailab	le boards:								
	Name	Version	Family		Device	Vendor	LES	Total I/Os	
в м	AX 10 DE10	1.0	MAX 10	10M	150DAF484C6GES	Altera	49760	360	3

Step 8: EDA Tool Settings then finish.

ool Type	Tool Name	Format(s)	Run Tool Automatically
esign Entry/S	<none></none>	None>	 Run this tool automatically to synthesize the current design
imulation	<none></none>	<none></none>	Run gate-level simulation automatically after compilation
Board-Level	Timing	<none></none>	•
	Symbol	<none></none>	•
	Signal Integrity	<none></none>	•
	Boundary Scan	<none></none>	•



Step 9: compile the design

Project Navigator 🍐 Hierarchy 🔹 🤉 💷 🖉	♦ Decoder7segment.v ×	Compilation Report - Decoder7segmer	n ×	IP Catalog
Entity:Instance	Table of Contents	Flow Summary		X =
MAX 10: 10MSODAF484C6GES Procoder7segment 1 Task Compilation Task Fisks Compile Design Fisks Fitter (Place & Route) Fitter	Flow Settings Flow Settings Flow Settings Flow Settings Flow Lon-Delati Global Setti Flow Cloped Time Flow Log Flow Log Flow Log Flow Log Flow Resages Flow Assages Flow Assages Flow Assages Flow Assages Flow Assages Flow Assages	Control of the second s	Successful - Wed Feb 23 17:58:21 2022 21.1.0 Build 842 10/21/2021 SJ Lite Edition Decoder Segment MAX 10 10MSDDAF484.66GES Preliminary 8 / 49,760 (-1 %) 0 / 0 / 75,7312 (0 %) 0 / 288 (0 %) 0 / 40 %) 0 / 2 (0 %)	 Installed IP Project Directory No Selection Available Ubrary Bask Functions DSP Initeface Protocols Memory Interfaces and Controllers Processors and Peripherals University Program Search for Partner IP
4	4			+ Add
Image: Stress of the	ort report th paths to report constrained for hold requi Analyzer was successful. Analyzer was successful.	d_ ØFind Negt irements rements 0 errors, 6 warnings 0 errors, 17 warnings		

Step 10: Pin assignment on FPGA

In this step, you will assign inputs (A,B,C,D) to the switches, And the outputs (led_a, led_b, led_c, led_d, led_e, led_f, led_g) to the first segment display.





To assign pins, refer to DE10-lite <u>FPGA user manual</u>. <u>User-Defined Slide Switch Section</u>



Figure 3-15 Connections between the slide switches and MAX 10 FPGA

Signal Name	FPGA Pin No.	Description	I/O Standard
SW0	PIN_C10	Slide Switch[0]	3.3-V LVTTL
SW1	PIN_C11	Slide Switch[1]	3.3-V LVTTL
SW2	PIN_D12	Slide Switch[2]	3.3-V LVTTL
SW3	PIN_C12	Slide Switch[3]	3.3-V LVTTL
SW4	PIN_A12	Slide Switch[4]	3.3-V LVTTL
SW5	PIN_B12	Slide Switch[5]	3.3-V LVTTL
SW6	PIN_A13	Slide Switch[6]	3.3-V LVTTL
SW7	PIN_A14	Slide Switch[7]	3.3-V LVTTL
SW8	PIN_B14	Slide Switch[8]	3.3-V LVTTL
SW9	PIN F15	Slide Switch[9]	3.3-V LVTTL

7-segment displays section



Figure 3-17 Connections between the 7-segment display HEX0 and the MAX 10 FPGA

Table 3-6 Pin A	ssignment of 7-	segment Displays	
Signal Name	FPGA Pin No.	Description	I/O Standard
HEX00	PIN_C14	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX01	PIN_E15	Seven Segment Digit 0[1]	3.3-V LVTTL
HEX02	PIN_C15	Seven Segment Digit 0[2]	3.3-V LVTTL
HEX03	PIN_C16	Seven Segment Digit 0[3]	3.3-V LVTTL
HEX04	PIN_E16	Seven Segment Digit 0[4]	3.3-V LVTTL
HEX05	PIN_D17	Seven Segment Digit 0[5]	3.3-V LVTTL
HEX06	PIN_C17	Seven Segment Digit 0[6]	3.3-V LVTTL
HEX07	PIN D15	Seven Segment Digit 0[7], DP	3.3-V LVTTL

 ${\rm Page}~12~{\rm of}~14$



Assign pins on Quartus, open the assignment tab, click on pin planner, and assign pins as figure below.



Step 11: Compile all project after pin assignment, like step 9. Program the FPGA. Step 12: load the program to the FPGA.

	· · · ·				
Entity:Instance Logic C	ells Table of Contents	Flow Summary			
MAX 10: 10M50DAF484C6GES	Elow Summary	< <filter>></filter>			
Image: Second state of the se	Flow Summary Flow Settings Flow Non-Default Global Setting Flow Ketapsed Time Flow Log Analysis & Synthesis Flow Log Flitter Flow Analysis & Synthesis Flow Suppressed Messages Flow Suppressed Messages Masembler Timing Analyzer	Content of the second sec	Succesful - Wed Feb 23 17:58:21 2022 21.1.0 Build 842 10/21/2021 SJ Lite Edition Decoder7segment MAX 10 10MS0DAF484C66ES Preliminary 8 / 49,760 (< 1 %) 0 0 / 11 / 360 (3 %) 0 0 / 1677,312 (0 %) 0 / 288 (0 %) 0 / 1 (0 %) 0 / 1 (0 %)		
Program Device (Open Programmer)	▼				
All Image: Construct of the state of	rt t ths to report rained for setup requirements rained for hold requirements rained for hold requirements the requirements rained for hold requirements rained for hold requirements rained for low requirements	nd Negt ; s, 6 warnings rs, 17 warnings			



Press "Start"

Hint: you may have a problem with the FPGA driver. Check the "device manager" and update the USB driver.

<u>File E</u> dit <u>V</u> iew	<u>adit View Processing Tools Window Help</u>					Search Intel FPGA						
着 Hardware Se	etup B-Blaster [US	B-0] Mode:	JTAG		•	Progre	ess:					
Enable real-time ISP to allow background programming when available												
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine				
Stop	output_files/Deco	10M50DAF48	00271EE9	00271EE9	\checkmark							
Auto Detect												
× Delete												
📥 Add File												
Change File.								F				
Save File												
Add Device.	TDI											
t [™] Up												
J [™] Down	10M50DAF4	84ES										
	4											

