MILESTONE 2

SKEE2263 Term Project

Group work

(5% final grade)

OBJECTIVES:

- Design using high-level modules
- Using primitives from the Altera library
- Interfacing CPLD with simple input/output devices

PART 2A: 4-BIT ACCUMULATOR-BASED COUNTER

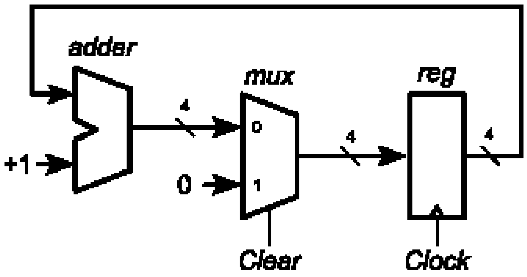


Fig.1

Accumulator based counter

Step 1.

Implement a 4-bit binary counter based on Fig. 1 with the following modules:

- **Adder**: use the 4-bit ripple adder from Milestone 1.
- **Mux**: use four 2:1 multiplexers from the Altera library
- **Reg**: use four DFF from the Altera library

Step 2.

For each new module (mux and reg), follow the general procedure of Milestone 1:

1. Compile until there are no errors.
2. Test/simulate using carefully selected data inputs.
3. Convert to symbol

Step 3.

Combine all modules based on Fig. 1.

Step 3:

Simulate by driving the circuit with a clock signal for 24 cycles. Use bus grouping to display the output in hex. Your simulation results must contain only three waveforms: clock, clear and the 4-bit output group. Assert the Clear signal at clock cycle 20. The output should start a 0 at clock cycle 0, increments by 1 until clock cycle 16 when it resets automatically. It will increment by 1 again until it is reset manually at clock cycle 20.

Step 4.

Save the circuit as a symbol. Name it ezcounter.bsf.

Step 5.

Print the following pages as PDF in landscape format.

Page 1: Circuit schematic

Page 2: Compilation report

Page 3: Simulation output waveforms.

PART 2B: BINARY TO 7-SEGMENT DECODER

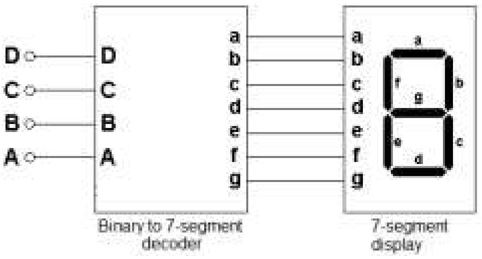
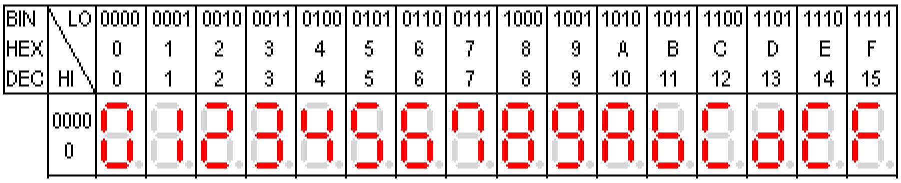


Fig. 2

Binary to 7segment decoder

Step 1.

Derive the complete truth table for binary to 7-segment display.



BIN	LO	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
HEX		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DEC	HI	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0000 0	0	1	2	3	4	5	6	7	8	9	A	b	C	d	E	F

Input				Output						
D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
…				…						
1	1	1	1	1	0	0	0	1	1	1

Step 2.

Implement the decoder using any method: gates, decoders, multiplexers, or Verilog code.

Step 3.

Simulate by giving all 16 input combinations 0000 through 1111.

Step 4.

Save the circuit as a module. Name it bin7seg.bsf.

Step 5.

Connect this module with counter module from Part 2A. Let the counter module generate the values 0000 through 1111 Use **hexadecimal** bus grouping to simplify your work. Your simulation results must contain only 4 waveforms: clock, clear, 4-bit output group from the counter (in hex), and 7-bit output from the decoder (in hex).

Step 6.

Print the following pages as PDF in landscape format.

Page 4: Binary to 7-segment truth table and LED output patterns

Page 5: Circuit schematic for decoder module

Page 6: Compilation report

Page 7: Simulation output waveform.

Step 7.

Combine all 6 PDF pages into 1 PDF document. Add a front cover. Submit.

Note: Plagiarism is easy to detect for this section. If plagiarism is detected, you will get 0 for the whole assignment.

PART 2C: 1 HZ CLOCK

Step 1.

Find the circuit or Verilog code which produces a 1 Hz pulse. Use your ingenuity or Googling skills.

Step 2.

Enter the design using Quartus.

Step 3.

Assign a pin on the CPLD as output. Make other assignments as necessary to make the module run.

Step 4.

Connect a resistor and an LED (on a breadboard) to the pin on the CPLD (assigned in Step 3).

Step 5.

Program the circuit into the CPLD. The LED should blink once every second.

Step 6.

Record a 10- to 15-second video showing that it works.

Step 6.

Print the following pages as PDF in landscape format.

Page 8: Schematic

Page 9: Compilation report

PART 2D: COMPLETE SYSTEM

Step 1.

In Quartus, connect the 1Hz clock to the counter, and connect the counter to the binary-to-7-segment display decoder. This is your top level entity.

Step 2.

Assign 7 pins on the CPLD to the output of the circuit from Step 1. Assign 1 pin as the input to the Clear wire with goes to the mux.

Step 3.

On the breadboard, wire up a 7-segment display and a switch. Connect to the pins that was assigned in Step 2.

Step 4.

Program the CPLD with the circuit from Step 1.

Step 5.

Run the CPLD. The display should display 0..9,A..F. It resets to 0 automatically when reaching F, OR when the switch is pressed.

Step 5.

Record a 20- to 30-second video showing that it works. Combine with the video from Part 2C Step 6. Upload to YouTube.

Step 6.

Print the following pages as PDF in landscape format.

Page 10: Schematic

Page 11: Compilation report

Page 12: References. Item [1] is the URL to your YouTube video.

Step 7.

Combine all 12 PDF pages into 1 PDF document. Add the plagiarism declaration as the front cover. Each team member must sign the declaration. Upload to elearning.utm.my.

PLAGIARISM DECLARATION

1. I know that plagiarism means taking and using the ideas, writings, works or inventions of another as if they were one's own. I know that plagiarism not only includes verbatim copying, but also the extensive use of another person's ideas without proper acknowledgement (which includes the proper use of quotation marks). I know that plagiarism covers this sort of use of material found in textual sources and from the Internet.
2. I acknowledge and understand that plagiarism is wrong.
3. I understand that my research must be accurately referenced. I have followed the rules and conventions concerning referencing and citation.
4. I declare that all material in this assignment is my own work and does not involve plagiarism.
5. I have not allowed, nor will I in the future allow, anyone to copy my work with the intention of passing it off as their own work.

Team Name	

Name	
Matric Number	
Signed	
Date	

Name	
Matric Number	
Signed	
Date	

Name	
Matric Number	
Signed	
Date	