|  |
| --- |
| Name: \_\_\_\_\_Servando\_Olvera\_\_\_\_\_\_\_\_ ID# \_\_\_\_\_\_\_\_1001909287\_\_\_\_\_\_\_    Date Submitted: \_\_\_\_\_02-20-2024\_\_\_\_ Time Submitted \_\_\_\_\_9:00\_pm\_\_\_\_\_\_    CSE 3341 Digital Logic Design II    CSE 5357 Advanced Digital Logic Design    Spring Semester 2024    **Lab 2 – Registered High-Speed Adder Subtractor**  **200 points**  Due Date – February 20, 2024, 11:59 PM    Submit on Canvas Assignments |

**DESIGN REQUIREMENTS**

Your assignment is to design an eight-bit registered high-speed adder subtractor that also produces carry-out, overflow, zero, and negative condition code outputs. The adder/subtractor component must use group carry lookahead architecture. You will code your design in SystemVerilog, simulate to verify its correctness, and test its functionality on a DE10-Lite development board. You will also perform a timing analysis on Quartus and compute the add/subtract time of your device.

Assume A, B, and R are eight-bit signed binary numbers using a two’s complement number system. CarryOut, OVR, ZERO, and NEG are condition codes determined by the result of the last operation performed. InA and InB load registers A and B, respectively. Out loads register R with the result of the last operation and the register CC with the last condition codes. Clear loads all zeros in all registers.

**ORGANIZATION DIAGRAM**

A diagram of a system

Description automatically generated

**HIRERARCHY DIAGRAM**

**A diagram of a computer

Description automatically generated**

**SYSTEM-VERILOG CODE**

**Part Top Module**

**A screenshot of a computer

Description automatically generated**

**Register Module**

**A screenshot of a computer

Description automatically generated**

**Four Bit Carry Look-Ahead Adder Module**

**A screenshot of a computer program

Description automatically generated**

**Group Carry Look-Ahead Adder Module**

**A screenshot of a computer program

Description automatically generated**

**Condition Code Module**

**A screenshot of a computer program

Description automatically generated**

**Binary to Seven Segment Display Module**

**A screenshot of a computer program

Description automatically generated**

**MUX/Controller Module**

**A screenshot of a computer program

Description automatically generated**

**Finite State Machine Module**

**A computer screen shot of a computer code

Description automatically generated**

**Four to One Decoder Module**

**A screenshot of a computer program

Description automatically generated**

**Clock Divider Module**

**A screen shot of a computer code

Description automatically generated**

**SIMULATION RESULTS WAVEFORM**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**PIN ASSIGMENTS**

**A screenshot of a computer

Description automatically generated**

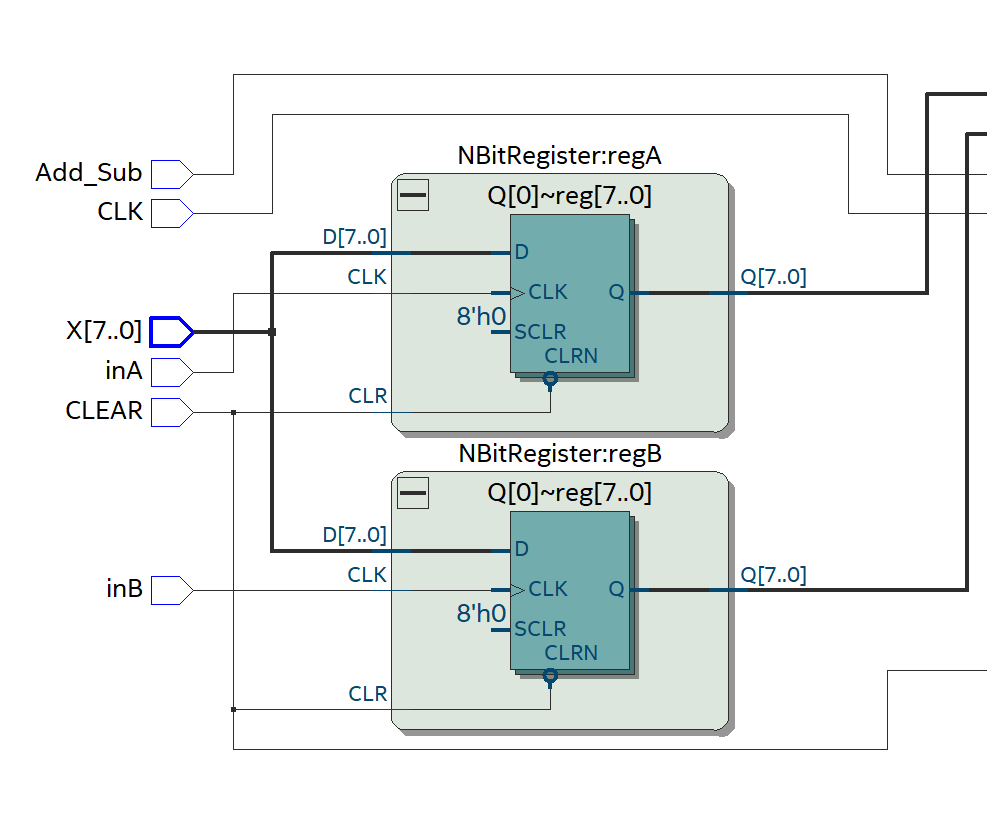
**RTL DIAGRAMS**

**Top Module**

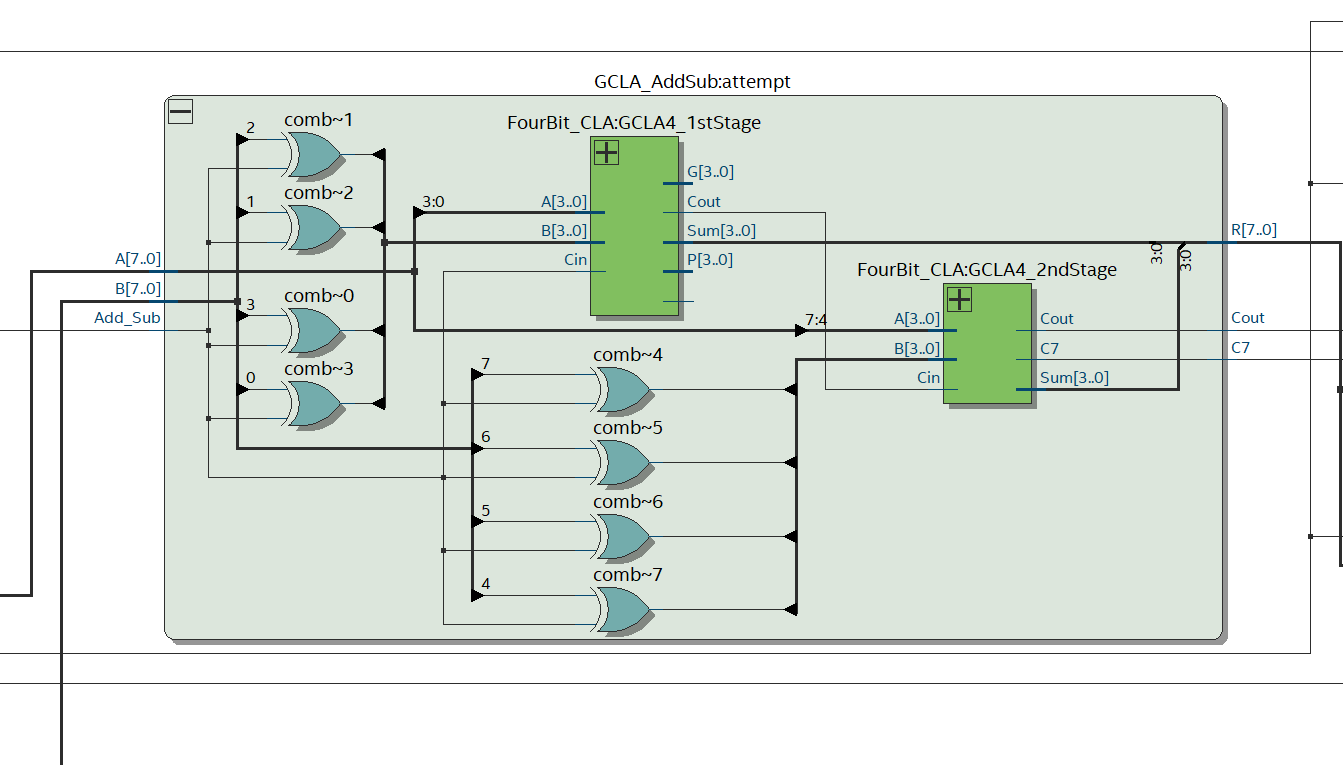
**A diagram of a computer program

Description automatically generated**

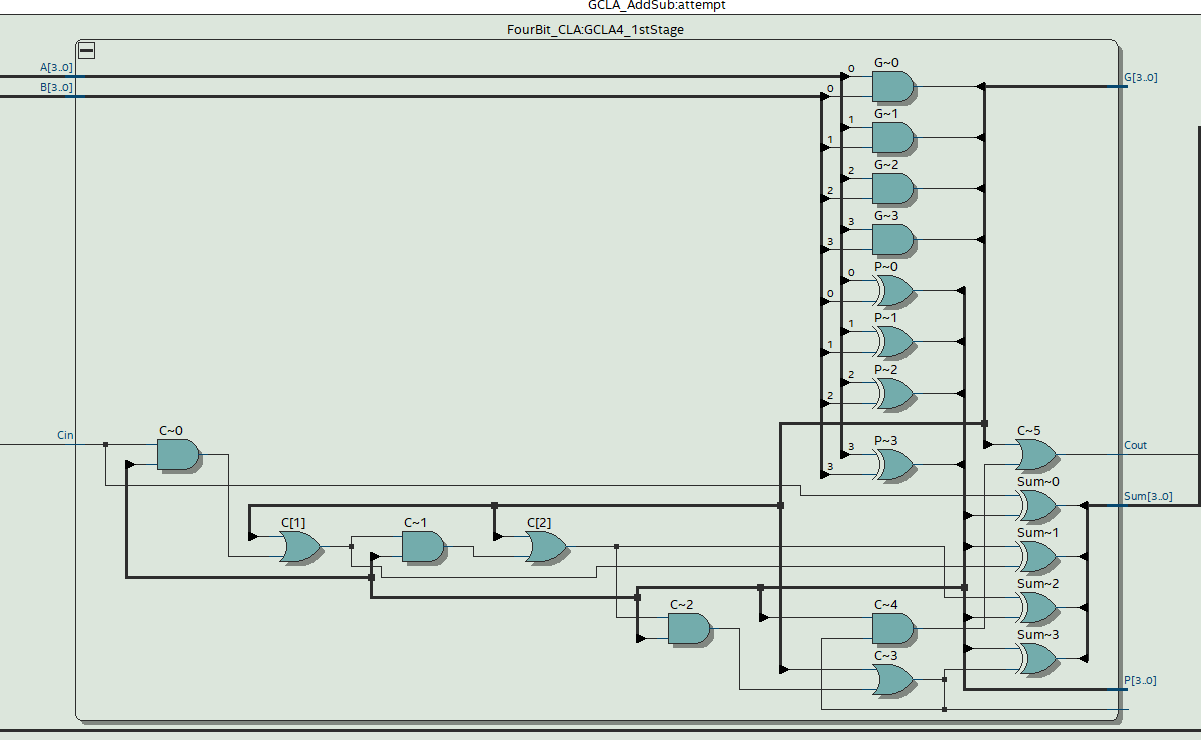
**Registers A & B**

****

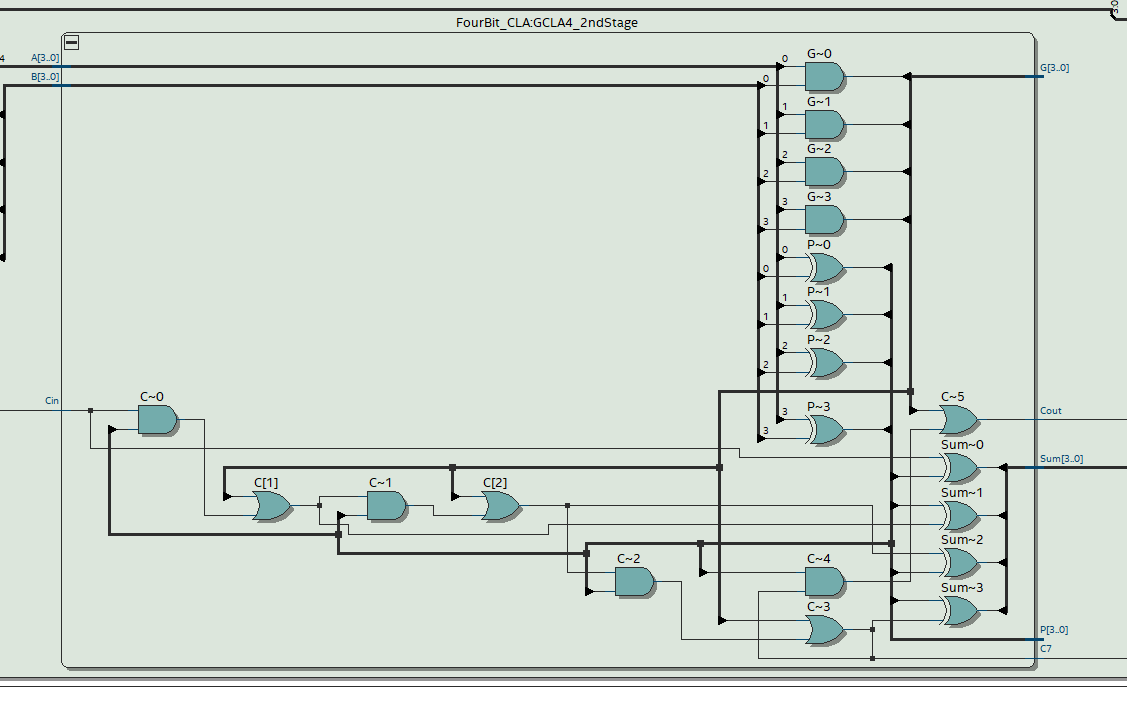
**Group Carry Look-Ahead Adder**

****

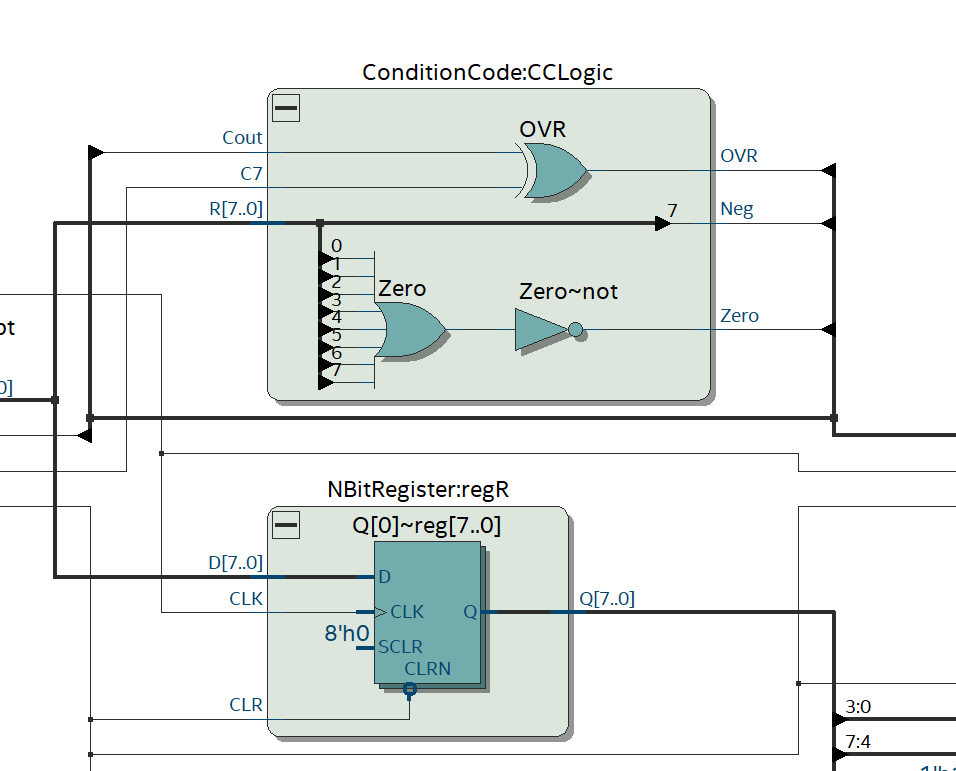
**4-Bit Carry Look-Ahead Adder Stage 1**

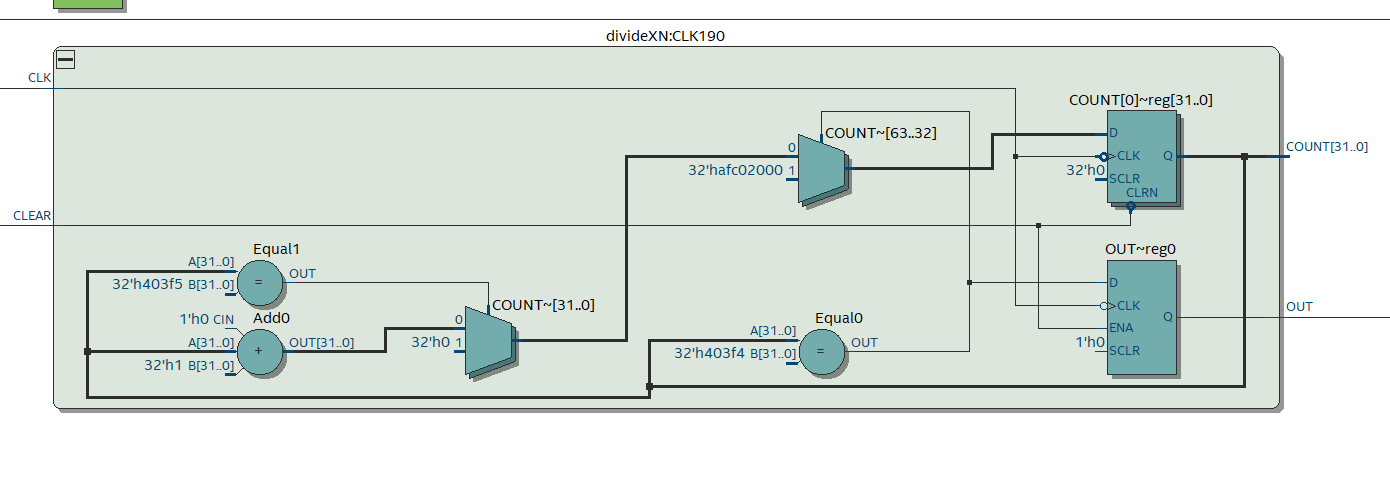
****

**4-Bit Carry Look-Ahead Adder Stage 2**

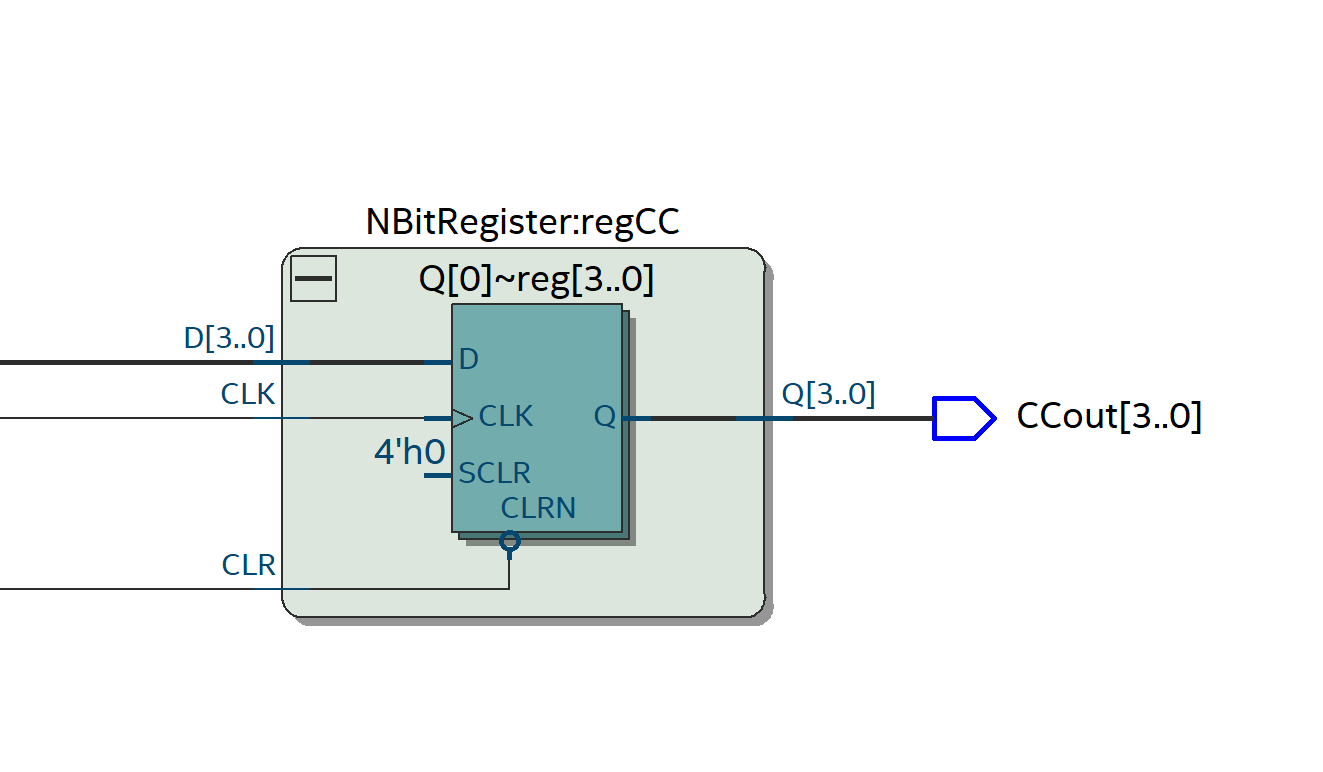
****

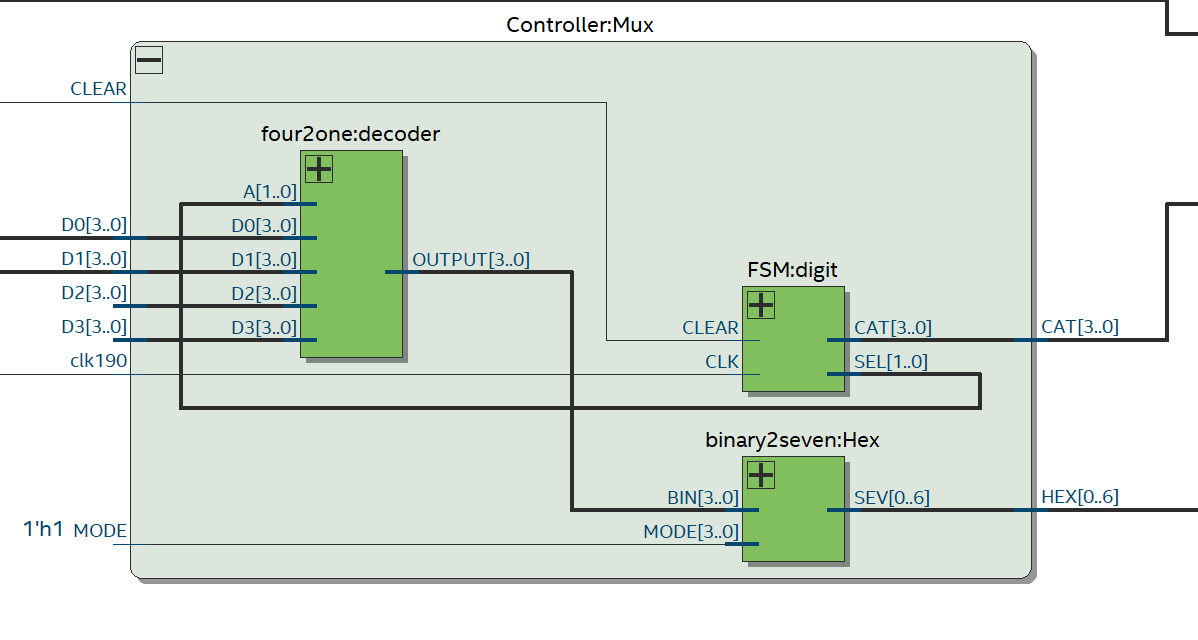
**Condition Code & Register R**

****

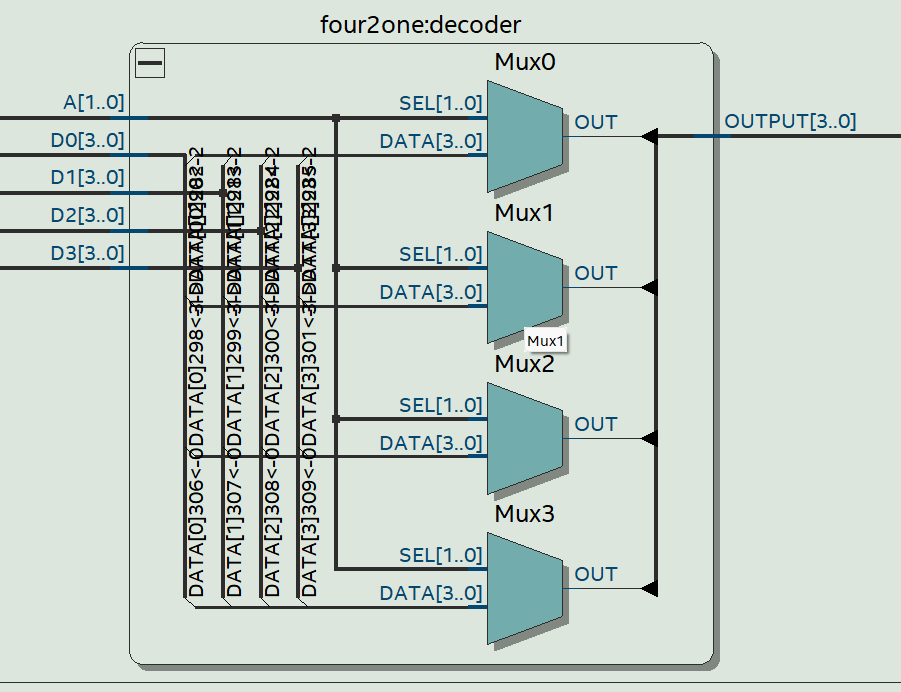
**Clock Divider **

**Register CC**

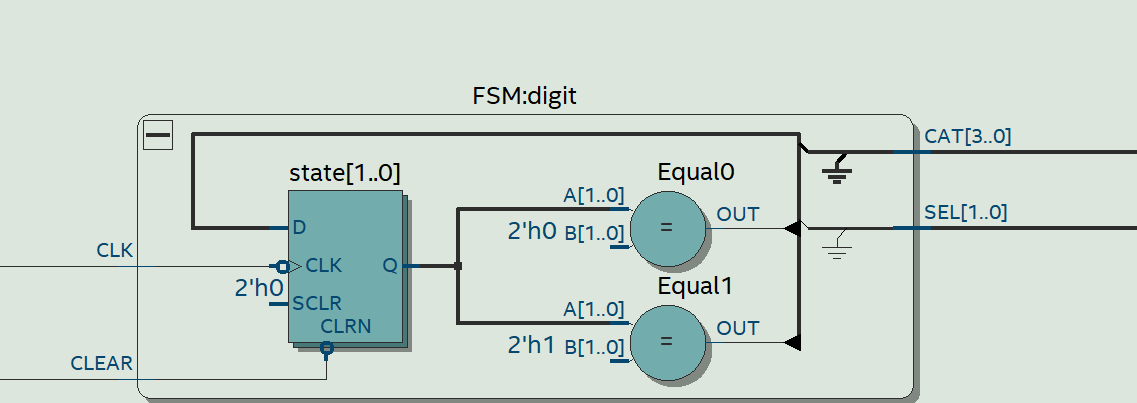
****

**Mux/Controller**

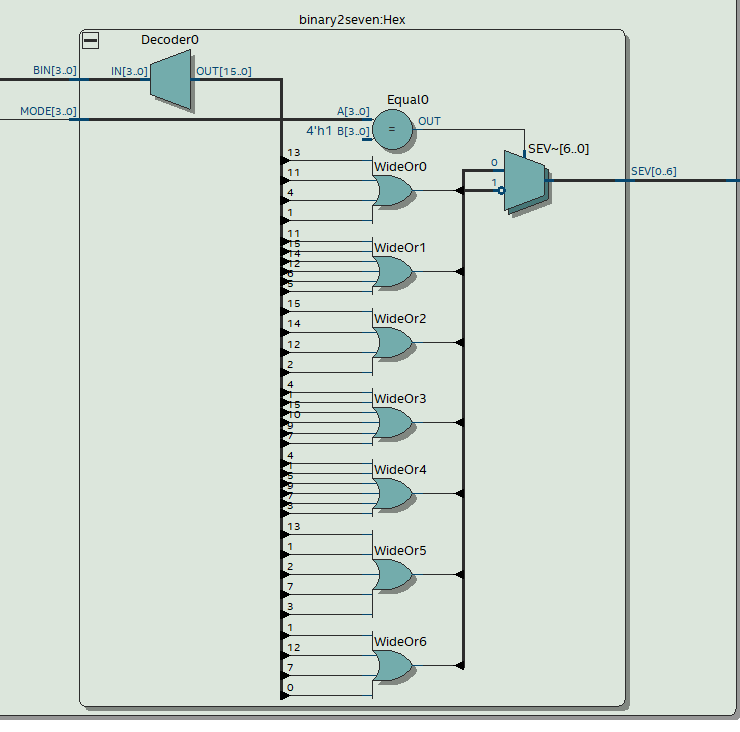
**Four to One Decoder**

****

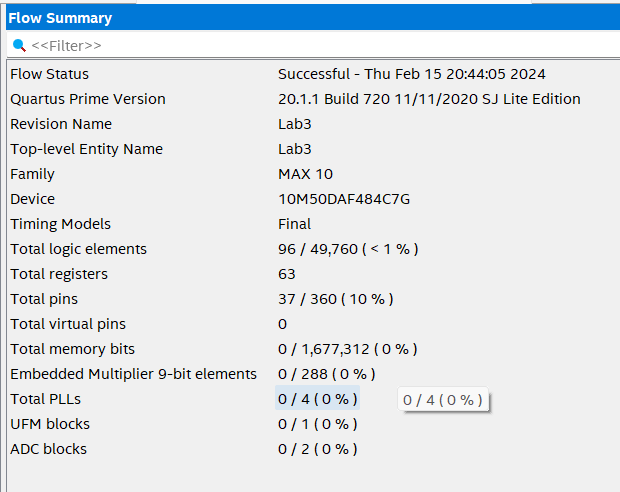
**Finite State Machine**

****

**Binary To Seven Hex Decoder**

****

**Compilation Summary**

****

**TIMING ANALYSIS DIAGRAMS**

**A screen shot of a computer

Description automatically generated**

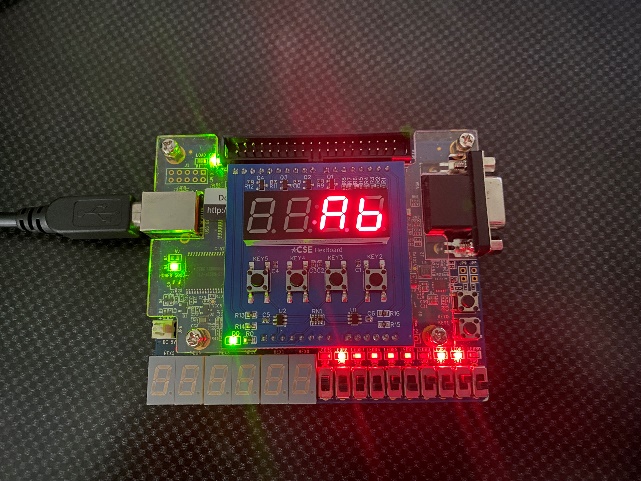
**TEST RESULTS**

**A table of numbers and symbols

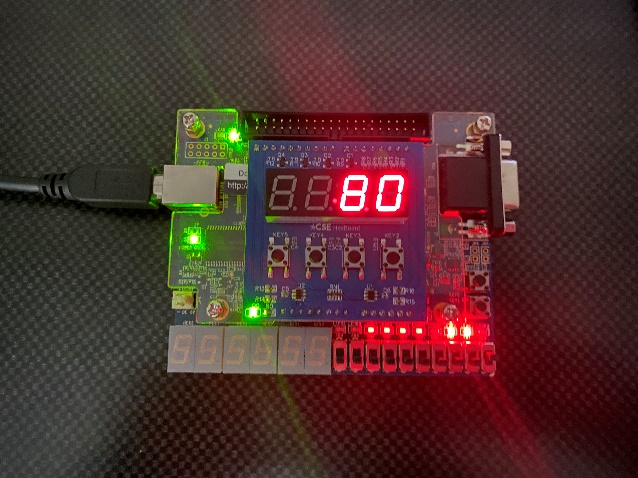
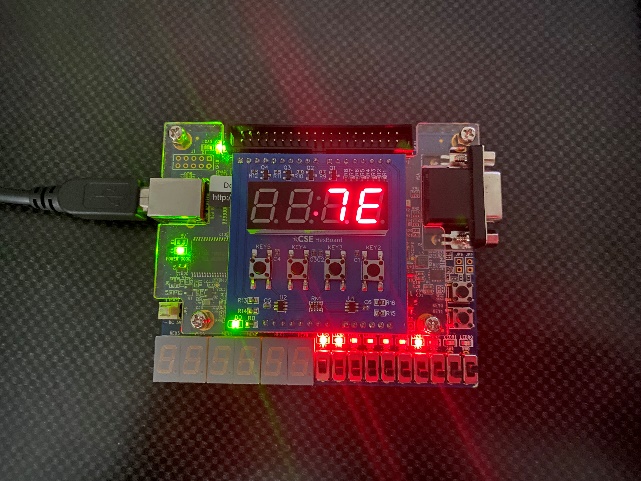
Description automatically generated**

**PHOTOS OF MARKED TEST RESULTS**

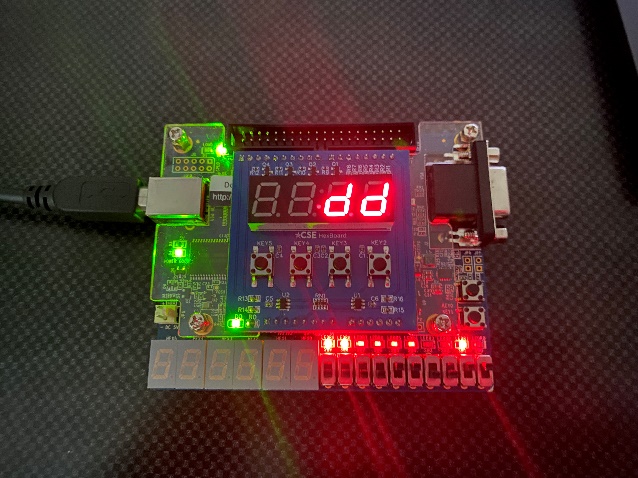
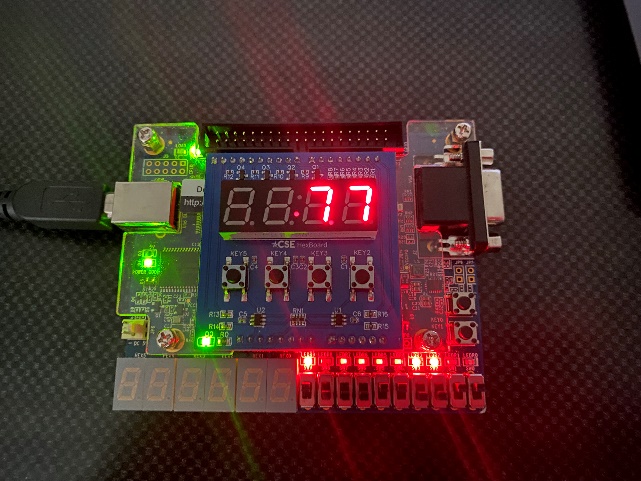
1. **(+) (-)**

** **

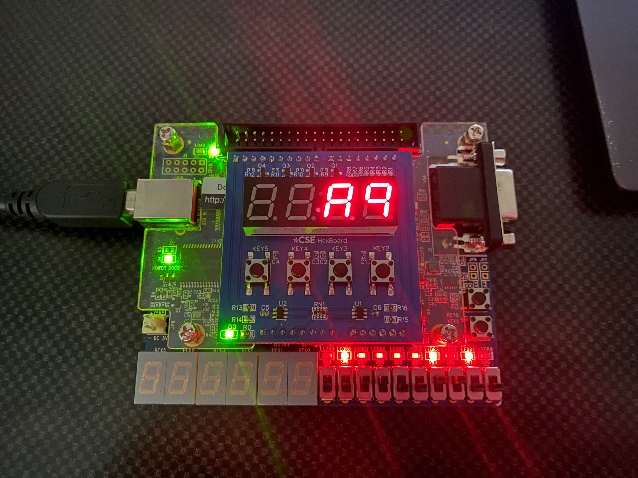
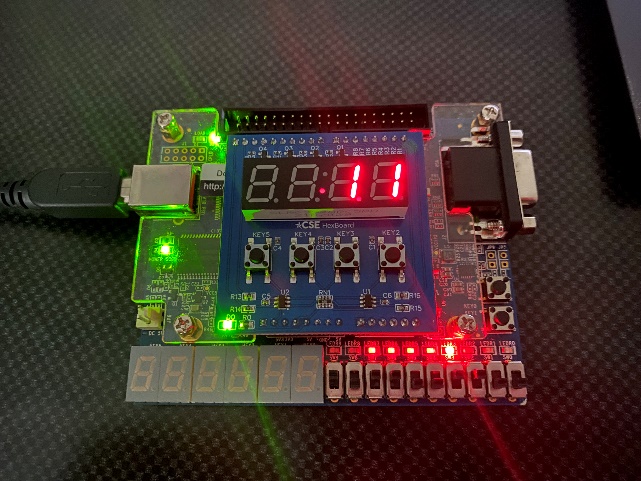
1. **(+) (-)**

** **

**e) (+) (-)**

** **

**h) (+) (-)**

** **