# EEL 3701 – Digital Logic and Computer Systems

# Report Design Problem 4

# Greg Bolling

# UF ID 64911960

## Problem Statement

The goal of the design was to create three circuits using a 64x8 ROM.

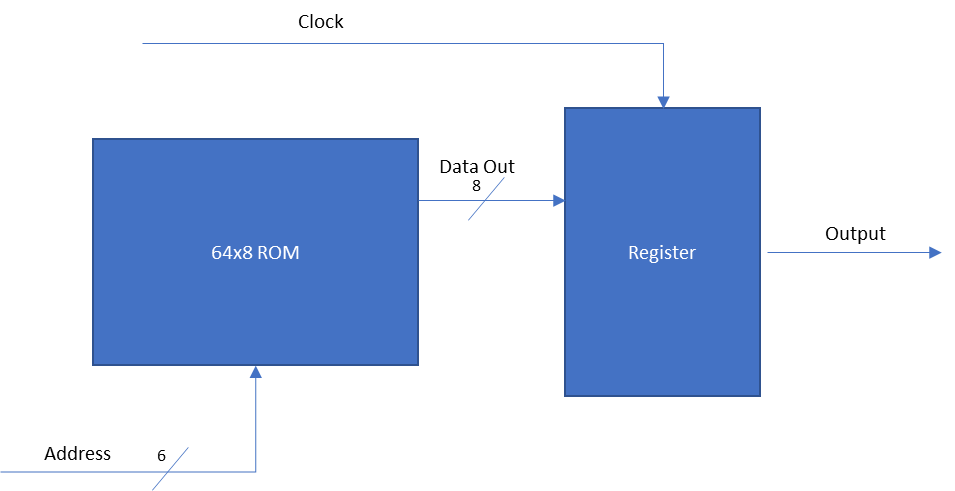
## Design

I implemented all three circuits using a 64x8 ROM. The code for the circuits is enclosed below.

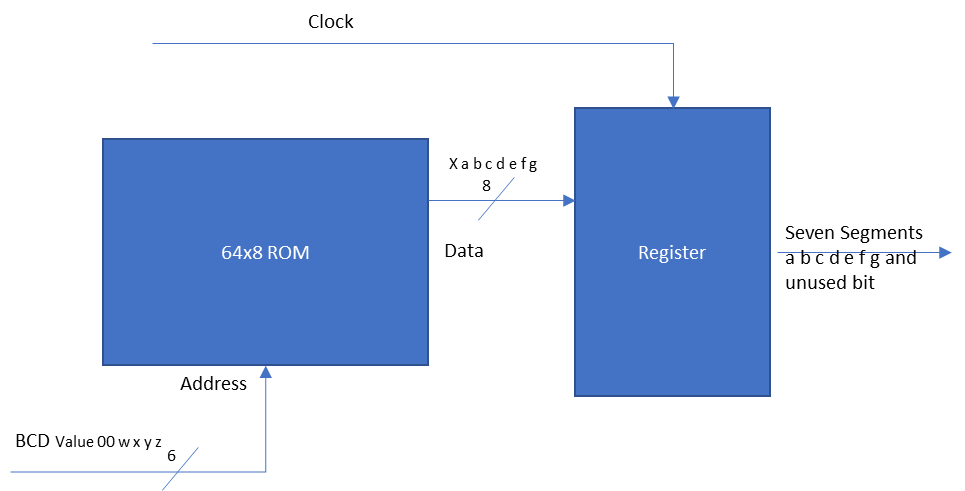
## Implementation

The ROM is a device that can take in any inputs and put out any outputs. The code uses this to create next state equations and similar to produce outputs.

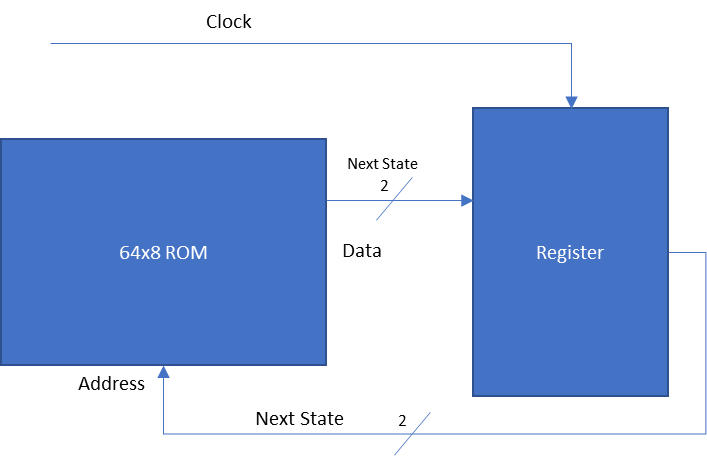
I implemented this design as a ROM using an array of std\_logic\_vector items. Each location had 8 bits and there were 64 locations.



In Part 2, the ROM became logic and it was used as an address to data translator.

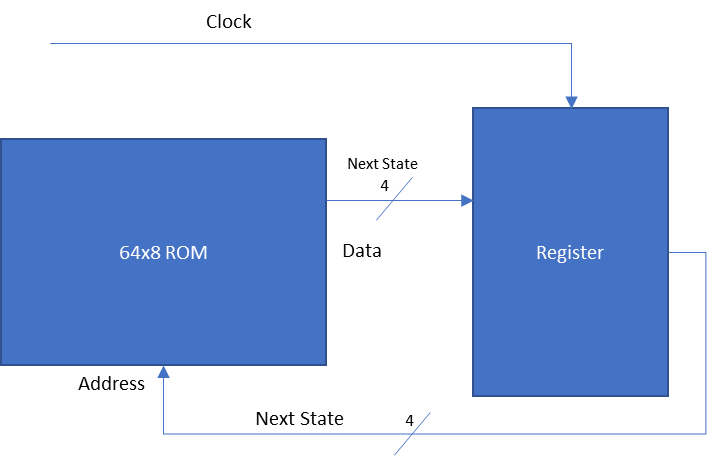
For the implementation of the A..J decoder, each address bit was just W,X,Y,Z as input to the address and the data was the output seven segments. Each bit location was a segment A..G.

For the state machines, the ROM was just a bit for the next state locations 00, 01, 10, 11 were the only bits used. So only four locations.



For the last one, the ROM was used as combinational logic to get the next state output equations for the counter. This counted 0, 3, 5, 7, 9, 0… The locations were each mapped to the next state values and that produced the count sequence.

This was the same as the above state machine only it used four bits.



## Part 1

The ROM code is enclosed below:

-------------------------------------------------------------------------------

-- Author: Greg Bolling

-- Date: March 12, 2020

-- Class; EEL 3701

-- Assignment: DesignProblem4 Top Level Entity

-- Class: 12368

-- Section 7441

-- PI Name: Savvas Ferekides

-------------------------------------------------------------------------------

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use ieee.std\_logic\_unsigned.all;

entity DesignProblem4\_Part1 is

port (

clock : in std\_logic;

address : in std\_logic\_vector(5 DOWNTO 0); -- six bits is 2^6 or 64 locations

outval : out std\_logic\_vector(7 downto 0)

);

end DesignProblem4\_Part1;

architecture AttachAll of DesignProblem4\_Part1 is

type ARam is array (0 to 63) of std\_logic\_vector(7 downto 0);

signal TheRam : ARam := (

"00000000", -- location 0

"00000000", -- location 1

"00000000", -- location 2

"00000000", -- location 3

"00000000", -- location 4

"00000000", -- location 5

"00000000", -- location 6

"00000000", -- location 7

"00000000", -- location 8

"00000000", -- location 9

"00000000", -- location 10

"00000000", -- location 11

"00000000", -- location 12

"00000000", -- location 13

"00000000", -- location 14

"00000000", -- location 15

"00000000", -- location 16

"00000000", -- location 17

"00000000", -- location 18

"00000000", -- location 19

"00000000", -- location 20

"00000000", -- location 21

"00000000", -- location 22

"00000000", -- location 23

"00000000", -- location 24

"00000000", -- location 25

"00000000", -- location 26

"00000000", -- location 27

"00000000", -- location 28

"00000000", -- location 29

"00000000", -- location 30

"00000000", -- location 31

"00000000", -- location 32

"00000000", -- location 33

"00000000", -- location 34

"00000000", -- location 35

"00000000", -- location 36

"00000000", -- location 37

"00000000", -- location 38

"00000000", -- location 39

"00000000", -- location 40

"00000000", -- location 41

"00000000", -- location 42

"00000000", -- location 43

"00000000", -- location 44

"00000000", -- location 45

"00000000", -- location 46

"00000000", -- location 47

"00000000", -- location 48

"00000000", -- location 49

"00000000", -- location 50

"00000000", -- location 51

"00000000", -- location 52

"00000000", -- location 53

"00000000", -- location 54

"00000000", -- location 55

"00000000", -- location 56

"00000000", -- location 57

"00000000", -- location 58

"00000000", -- location 59

"00000000", -- location 60

"00000000", -- location 61

"00000000", -- location 62

"00000000" -- location 63

);

begin

ARAMProc: process(clock)

begin

if (clock'event and clock = '1') then

outval <= TheRam(to\_integer(unsigned(address)));

end if;

end process;

end AttachAll;

## Part 2a

The code is enclosed below:

-------------------------------------------------------------------------------

-- Author: Greg Bolling

-- Date: March 12, 2020

-- Class; EEL 3701

-- Assignment: DesignProblem4 Top Level Entity

-- Class: 12368

-- Section 7441

-- PI Name: Savvas Ferekides

-------------------------------------------------------------------------------

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use ieee.std\_logic\_unsigned.all;

entity DesignProblem4Part2a is

port (

display : out std\_logic\_vector(3 downto 0);

clock : in std\_logic;

w : in std\_logic;

x : in std\_logic;

y : in std\_logic;

z : in std\_logic;

a : out std\_logic;

b : out std\_logic;

c : out std\_logic;

d : out std\_logic;

e : out std\_logic;

f : out std\_logic;

g : out std\_logic

);

end DesignProblem4Part2a;

architecture AttachAll of DesignProblem4Part2a is

signal address : std\_logic\_vector(5 DOWNTO 0); -- six bits is 2^6 or 64 locations

signal outval : std\_logic\_vector(7 downto 0);

type ARam is array (0 to 63) of std\_logic\_vector(7 downto 0);

signal TheRam : ARam := (-- X a b c d e f g and address is 00 w x y z

"00000000", -- location 0 minterm m0

"00001000", -- location 1 minterm m1

"01100000", -- location 2 minterm m2

"00110001", -- location 3 minterm m3

"01000010", -- location 4 minterm m4

"00110000", -- location 5 minterm m5

"00111000", -- location 6 minterm m6

"00000100", -- location 7 minterm m7

"01001000", -- location 8 minterm m8

"01111001", -- location 9 minterm m9

"01000011", -- location 10 minterm m10

"00000000", -- location 11 minterm m11

"00000000", -- location 12 minterm m12

"00000000", -- location 13 minterm m13

"00000000", -- location 14 minterm m14

"00000000", -- location 15 minterm m15

"00000000", -- location 16

"00000000", -- location 17

"00000000", -- location 18

"00000000", -- location 19

"00000000", -- location 20

"00000000", -- location 21

"00000000", -- location 22

"00000000", -- location 23

"00000000", -- location 24

"00000000", -- location 25

"00000000", -- location 26

"00000000", -- location 27

"00000000", -- location 28

"00000000", -- location 29

"00000000", -- location 30

"00000000", -- location 31

"00000000", -- location 32

"00000000", -- location 33

"00000000", -- location 34

"00000000", -- location 35

"00000000", -- location 36

"00000000", -- location 37

"00000000", -- location 38

"00000000", -- location 39

"00000000", -- location 40

"00000000", -- location 41

"00000000", -- location 42

"00000000", -- location 43

"00000000", -- location 44

"00000000", -- location 45

"00000000", -- location 46

"00000000", -- location 47

"00000000", -- location 48

"00000000", -- location 49

"00000000", -- location 50

"00000000", -- location 51

"00000000", -- location 52

"00000000", -- location 53

"00000000", -- location 54

"00000000", -- location 55

"00000000", -- location 56

"00000000", -- location 57

"00000000", -- location 58

"00000000", -- location 59

"00000000", -- location 60

"00000000", -- location 61

"00000000", -- location 62

"00000000" -- location 63

);

begin

display <= "0001";

address <= "00" & w & x & y & z; -- only need 16 locations for 2^4 input combinations

ARAMProc: process(clock)

begin

if (clock'event and clock = '1') then

outval <= TheRam(to\_integer(unsigned(address)));

end if;

end process;

a <= outval(0);

b <= outval(1);

c <= outval(2);

d <= outval(3);

e <= outval(4);

f <= outval(5);

g <= outval(6);

end AttachAll;

## Part 2b

The code is enclosed below:

-------------------------------------------------------------------------------

-- Author: Greg Bolling

-- Date: March 12, 2020

-- Class; EEL 3701

-- Assignment: DesignProblem4 Top Level Entity

-- Class: 12368

-- Section 7441

-- PI Name: Savvas Ferekides

-------------------------------------------------------------------------------

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use ieee.std\_logic\_unsigned.all;

--------------fastclock for sevensegment display-------------

entity DesignProblem4Part2b is

Port (clock, reset : in STD\_LOGIC;

display: out std\_logic\_vector(3 DOWNTO 0);

SevenSeg : out std\_logic\_vector(6 DOWNTO 0));

end DesignProblem4Part2b;

architecture Behavioral of DesignProblem4Part2b is

signal a\_g : std\_logic\_vector(6 downto 0); -- declare the (state-machine) enumerated type

signal present\_state : std\_logic\_vector(7 downto 0);

type ARam is array (0 to 63) of std\_logic\_vector(7 downto 0);

signal TheRam : ARam := (

"00000001", -- location 0

"00000010", -- location 1

"00000011", -- location 2

"00000000", -- location 3

"00000000", -- location 4

"00000000", -- location 5

"00000000", -- location 6

"00000000", -- location 7

"00000000", -- location 8

"00000000", -- location 9

"00000000", -- location 10

"00000000", -- location 11

"00000000", -- location 12

"00000000", -- location 13

"00000000", -- location 14

"00000000", -- location 15

"00000000", -- location 16

"00000000", -- location 17

"00000000", -- location 18

"00000000", -- location 19

"00000000", -- location 20

"00000000", -- location 21

"00000000", -- location 22

"00000000", -- location 23

"00000000", -- location 24

"00000000", -- location 25

"00000000", -- location 26

"00000000", -- location 27

"00000000", -- location 28

"00000000", -- location 29

"00000000", -- location 30

"00000000", -- location 31

"00000000", -- location 32

"00000000", -- location 33

"00000000", -- location 34

"00000000", -- location 35

"00000000", -- location 36

"00000000", -- location 37

"00000000", -- location 38

"00000000", -- location 39

"00000000", -- location 40

"00000000", -- location 41

"00000000", -- location 42

"00000000", -- location 43

"00000000", -- location 44

"00000000", -- location 45

"00000000", -- location 46

"00000000", -- location 47

"00000000", -- location 48

"00000000", -- location 49

"00000000", -- location 50

"00000000", -- location 51

"00000000", -- location 52

"00000000", -- location 53

"00000000", -- location 54

"00000000", -- location 55

"00000000", -- location 56

"00000000", -- location 57

"00000000", -- location 58

"00000000", -- location 59

"00000000", -- location 60

"00000000", -- location 61

"00000000", -- location 62

"00000000" -- location 63

);

begin

P0: process(clock, reset)

begin

if(reset = '1') then

present\_state <= (others =>'0');

elsif (clock='1' and clock'event) then

present\_state <= TheRam(to\_integer(unsigned(present\_state)));

end if;

end process;

P1: process(present\_state, a\_g)

begin

case present\_state(1 downto 0) is

when "00" =>

a\_g<= "0000001";

display <= "0001";

when "01" =>

a\_g<= "0100100";

display <= "0010";

when "10" =>

a\_g<= "0000000";

display <= "0100";

when "11" =>

a\_g<= "0001100";

display <= "1000";

when others =>

a\_g<= "0001100";

display <= "1000";

end case;

end process;

SevenSeg <= a\_g(0) & a\_g(1) & a\_g(2) &

a\_g(3) & a\_g(4) & a\_g(5) & a\_g(6);

end Behavioral;

## Part 2c

The code is enclosed below:

-------------------------------------------------------------------------------

-- Author: Greg Bolling

-- Date: March 12, 2020

-- Class; EEL 3701

-- Assignment: DesignProblem4 Top Level Entity

-- Class: 12368

-- Section 7441

-- PI Name: Savvas Ferekides

-------------------------------------------------------------------------------

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use ieee.std\_logic\_unsigned.all;

entity DesignProblem4Part2c is

Port (clock, reset : in STD\_LOGIC;

display: out std\_logic\_vector(3 DOWNTO 0);

Seven\_Seg : out std\_logic\_vector(6 DOWNTO 0));

end DesignProblem4Part2c;

architecture Behavioral of DesignProblem4Part2c is

signal a\_g : std\_logic\_vector(6 downto 0); -- declare the (state-machine) enumerated type

signal count : std\_logic\_vector(3 downto 0); -- counter

signal slow\_clk : std\_logic;

type ARam is array (0 to 63) of std\_logic\_vector(7 downto 0);

signal TheRam : ARam := (

"00000011", -- location 0

"00000000", -- location 1

"00000000", -- location 2

"00000101", -- location 3

"00000000", -- location 4

"00000111", -- location 5

"00000000", -- location 6

"00001001", -- location 7

"00000000", -- location 8

"00000000", -- location 9

"00000000", -- location 10

"00000000", -- location 11

"00000000", -- location 12

"00000000", -- location 13

"00000000", -- location 14

"00000000", -- location 15

"00000000", -- location 16

"00000000", -- location 17

"00000000", -- location 18

"00000000", -- location 19

"00000000", -- location 20

"00000000", -- location 21

"00000000", -- location 22

"00000000", -- location 23

"00000000", -- location 24

"00000000", -- location 25

"00000000", -- location 26

"00000000", -- location 27

"00000000", -- location 28

"00000000", -- location 29

"00000000", -- location 30

"00000000", -- location 31

"00000000", -- location 32

"00000000", -- location 33

"00000000", -- location 34

"00000000", -- location 35

"00000000", -- location 36

"00000000", -- location 37

"00000000", -- location 38

"00000000", -- location 39

"00000000", -- location 40

"00000000", -- location 41

"00000000", -- location 42

"00000000", -- location 43

"00000000", -- location 44

"00000000", -- location 45

"00000000", -- location 46

"00000000", -- location 47

"00000000", -- location 48

"00000000", -- location 49

"00000000", -- location 50

"00000000", -- location 51

"00000000", -- location 52

"00000000", -- location 53

"00000000", -- location 54

"00000000", -- location 55

"00000000", -- location 56

"00000000", -- location 57

"00000000", -- location 58

"00000000", -- location 59

"00000000", -- location 60

"00000000", -- location 61

"00000000", -- location 62

"00000000" -- location 63

);

component slow\_clock is

port(

clock, reset : in STD\_LOGIC;

slow\_clk : out STD\_LOGIC); end component;

component SevenSeg is

port(

binary : in std\_logic\_vector(3 downto 0);

a\_to\_g : out std\_logic\_vector(6 downto 0)); end component;

begin

U0: slow\_clock

port map(

slow\_clk=>slow\_clk,

reset=>reset,

clock=>clock);

U1: SevenSeg

port map(

binary => count,

a\_to\_g => a\_g);

P0: process(slow\_clk, reset)

begin

if(reset = '1') then

count <= "0000";

elsif (slow\_clk ='1' and slow\_clk'event) then

count(3) <= TheRam(to\_integer(unsigned(count(3 downto 0)))) (3);

count(2) <= TheRam(to\_integer(unsigned(count(3 downto 0)))) (2);

count(1) <= TheRam(to\_integer(unsigned(count(3 downto 0)))) (1);

count(0) <= TheRam(to\_integer(unsigned(count(3 downto 0)))) (0);

-- count(3) <= count(2) and count(1);

-- count(2) <= ((not count(3) and count(0)) and not count(2)) or

-- ((not count(3) and count(0)) and not count(1));

-- count(1) <= not count(3) and not count(1);

-- count(0) <= not count(3);

end if;

end process;

display <= "0001";

Seven\_Seg <= a\_g(0) & a\_g(1) & a\_g(2) &

a\_g(3) & a\_g(4) & a\_g(5) & a\_g(6);

end Behavioral;