CS/EE 120B

Homework #2 Given 2/15/01. Due 2/22/01

1. Use a custom datapath to design a circuit at the FSMD level for the following problem:

   Input an 8-bit number. Output a 1 if the number has the same number of 0's and 1's, otherwise, output a 0. e.g. the number 10111011 will produce a 0 output, whereas, the number 00110011 will produce a 1 output.

   Solve the problem by answering the following questions.

   a) Write the high-level pseudo code program to solve the problem.

   b) What is the minimum number of registers and functional units needed in the datapath? Specify what the functional units are and what variables are being assigned to the registers.

   c) Draw the datapath circuit.

   d) Derive the state action table for a Moore machine.

   e) Derive the next-state and output equations.

   f) Draw the FSMD circuit.

Answer

With an open-ended question like this, there are many other correct solutions. The following is just one possible answer.

The high-level pseudo code for solving the problem is as follows:

```
input n
count = 0                              // for counting the number of zeros

while n ≠ 0 {
   if LSB(n) = 1                       // least significant bit of n
      count = count + 1
   else
      count = count - 1
   n = n >> 1                          // shift n right one bit
   }

if count = 0 then
   output 1
else
   output 0

assert done
```

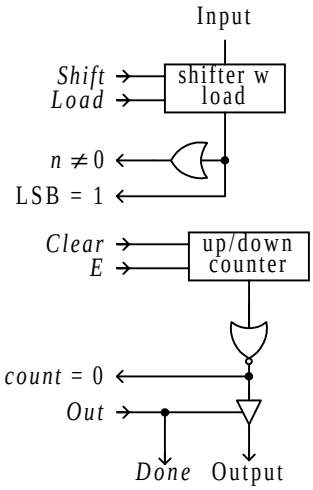
1 point for the pseudo code

The functional units required by the custom datapath are as follows:

- A shifter with parallel load register for storing *n*.
- One 3-bit up/down counter for *count*.
- A "not equal to 0" comparator and an "equal to 0" comparator.

1 point for the functional units and registers needed

Customize datapath



2 points for the datapath

State-action table for the Moore FSM

Current State $Q_3Q_2Q_1Q_0$ Name	Next State [Condition, State]	Unconditional Control and Datapath Actions
0 0 0 0 s_0	$\begin{bmatrix} start = 0, s_0 \\ start = 1, s_1 \end{bmatrix}$	*count* = 0 *done* = 0 *output* = Z
0 0 0 1 s_1	s_2	n = *input*
0 0 1 0 s_2	$\begin{bmatrix} n \neq 0, s_3 \\ n = 0, s_6 \end{bmatrix}$	
0 0 1 1 s_3	$\begin{bmatrix} \text{LSB}(n) = 1, s_4 \\ \text{LSB}(n) \neq 1, s_5 \end{bmatrix}$	
0 1 0 0 s_4	s_2	*count* = *count* + 1 $n = n >> 1$
0 1 0 1 s_5	s_2	*count* = *count* – 1 $n = n >> 1$
0 1 1 0 s_6	$\begin{bmatrix} count = 0, s_7 \\ count = 1, s_8 \end{bmatrix}$	
0 1 1 1 s_7	s_0	*output* = 1 *done* = 1
1 0 0 0 s_8	s_0	*output* = 0 *done* = 1

2 points for the state-action table

The following is done without simplifications of the state encodings.

Next-state equations using D flip-flops, $D_i = Q_{i(\text{next})}$ are:

For D_0, $Q_0 = 1$ in states s_1, s_3, s_5, and s_7, therefore, we look for these states in the next state column, i.e. what is the current state and may be an optional condition that will lead to these states. Hence, we get

$D_0 = s_0(\text{start}) + s_2(n \neq 0) + s_3(\text{LSB} = 1)' + s_6(count = 0)$
$= Q_3'Q_2'Q_1'Q_0'(\text{start}) + Q_3'Q_2'Q_1Q_0' (n \neq 0) + Q_3'Q_2'Q_1Q_0 (\text{LSB} = 1)' + Q_3'Q_2Q_1Q_0' (count = 0)$

$D_1 = s_1 + s_4 + s_5 + s_2(n \neq 0) + s_2(n \neq 0)' + s_6(count = 0)$
$= s_1 + s_2 + s_4 + s_5 + s_6(count = 0)$
$= Q_3'Q_2'Q_1'Q_0 + Q_3'Q_2'Q_1Q_0' + Q_3'Q_2Q_1'Q_0' + Q_3'Q_2Q_1'Q_0 + Q_3'Q_2Q_1Q_0' (count = 0)$

$D_2 = s_3(\text{LSB} = 1) + s_3(\text{LSB} = 1)' + s_2(n = 0) + s_6(count = 0)$
$= Q_3'Q_2'Q_1Q_0 + Q_3'Q_2'Q_1Q_0' (n = 0) + Q_3'Q_2Q_1Q_0' (count = 0)$

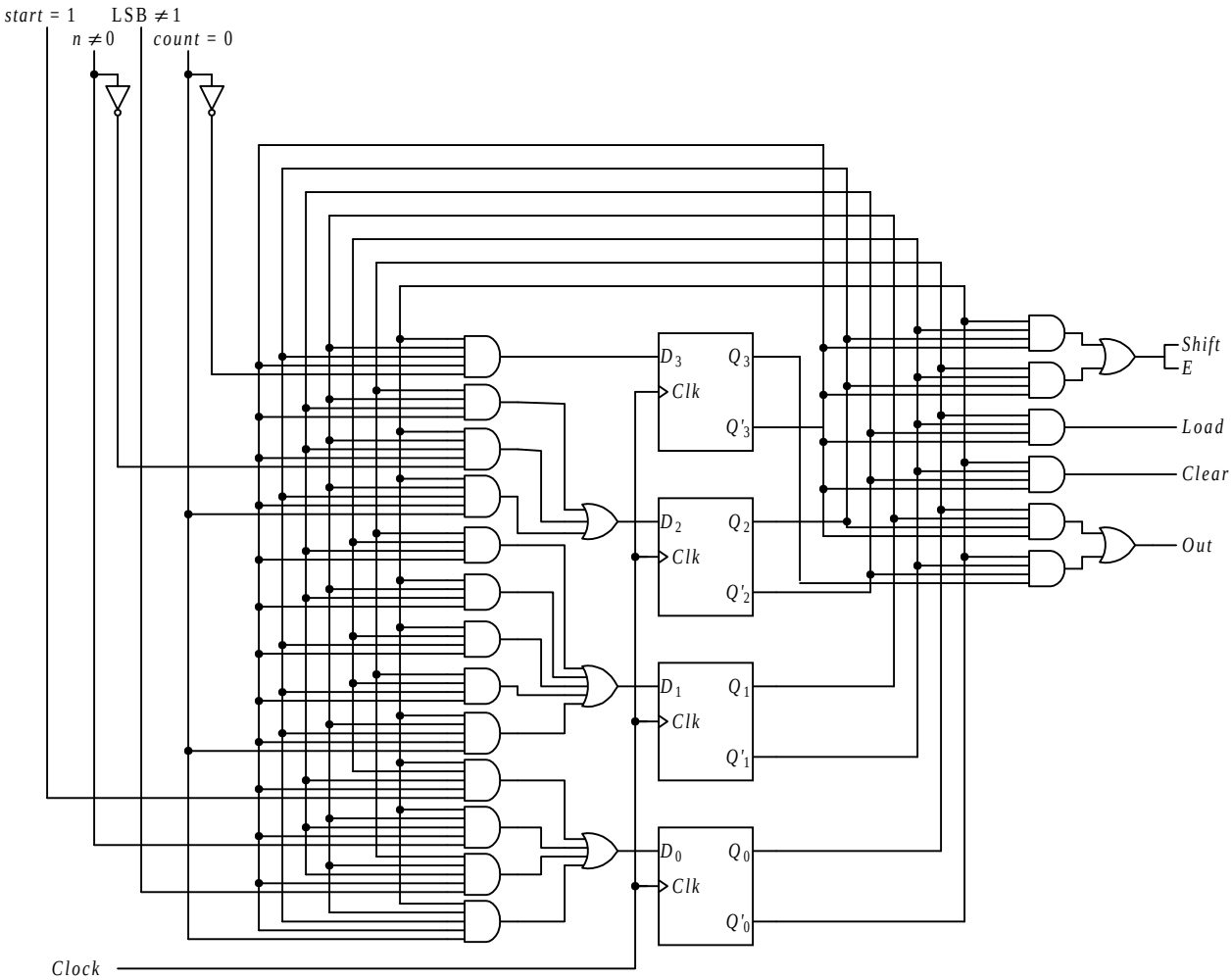
$D_3 = s_6(count = 0)'$
$= Q_3'Q_2Q_1Q_0' (count = 0)'$

2 points for the next-state and output equations

The output equations are:

$Shift = s_4 + s_5 = Q_3'Q_2Q_1'Q_0' + Q_3'Q_2Q_1'Q_0$
$Load = s_1 = Q_3'Q_2'Q_1'Q_0$
$\text{Clear} = s_0 = Q_3'Q_2'Q_1'Q_0'$
$E = s_4 + s_5 = Q_3'Q_2Q_1'Q_0' + Q_3'Q_2Q_1'Q_0$
$Out = s_7 + s_8 = Q_3'Q_2Q_1Q_0 + Q_3Q_2'Q_1'Q_0'$

The FSM circuit is as follows:



2 points for the FSM circuit

10 points total.