UNIVERSITY OF CALIFORNIA, RIVERSIDE

Department of Computer Science and Engineering

CS/EE120B – Introduction to Embedded Systems

Midterm 2

Summer 2 August 20, 2002

16

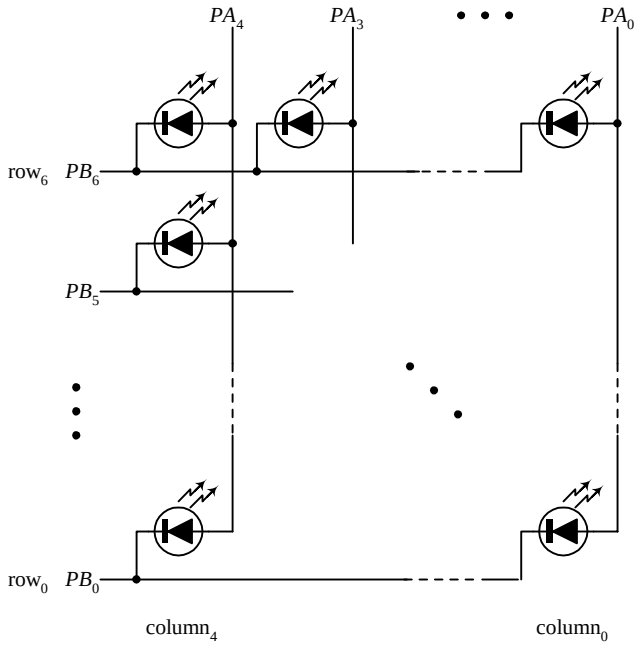
Name: Solution Key **Student ID#:**

Please print legibly

(Numbers in parenthesis denote total possible points for question.)

Do questions 1, 2 and 3, and either question 4 or 5.

1. A 5×7 array of LEDs is connected with the positive side of the LEDs connected in common for each of the 5 columns, and the negative side in common for each of the 7 rows. The columns are connected to port A and the rows to port B of the 8051 as shown below. Write a C style pseudo code that will turn on the LEDs in row 4 one at a time from left to right, i.e. turn on only the LED in column 4, then turn on only the LED in column 3, etc. When you get to column 0, then cycle back to column 4. (4)



Answer

Version 1. (using a rotate)

```
// 0 turns on a row
// 1 turns on a column
// LED turns on with a 0 on the row and a 1 on the column

PB = xEF  // select row 4 with a zero and all the other rows off
PA = x10; // select column 4 with a one and all the other columns off
while(1){
        for(I=0;I<10000;I++)
        PA = rotate_right(PA)
}
```

Version 2. (not using the rotate)

```
// 0 turns on a row
// 1 turns on a column
// LED turns on with a 0 on the row and a 1 on the column

PB = xEF         // select row 4 with a zero
while(1){
        PA = x10; // select column 4 with a one and all the others off
        for(I=0;I<10000;I++)
        PA = x08; // select column 3 with a one and all the others off
        for(I=0;I<10000;I++)
        PA = x04; // select column 2 with a one and all the others off
        for(I=0;I<10000;I++)
        PA = x02; // select column 1 with a one and all the others off
        for(I=0;I<10000;I++)
        PA = x01; // select column 0 with a one and all the others off
        for(I=0;I<10000;I++)
}
```

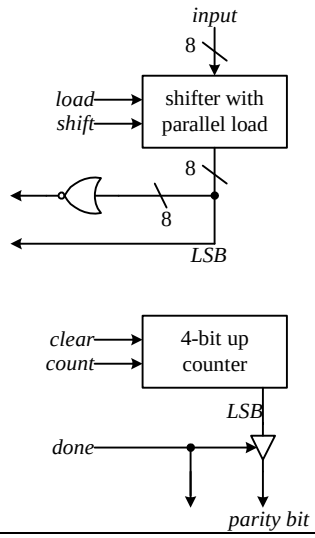
2. In serial communication, the even parity error checking option is sometimes used. The even parity protocol counts the number of 1's in the byte. If there is an odd number of 1's, then the parity bit is set to a 1 otherwise, it is set to a 0. Design a custom datapath that inputs a byte and outputs the correct parity bit. You can use any of the components that we have discussed. Make the datapath as small as possible and requires as few states as possible by the FSM. (4)

Answer

Version 1.

functional units needed:

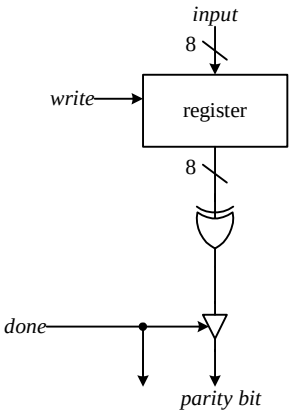
- shifter/register with parallel load
- 4-bit up counter



Version 2.

functional units needed:

- register
- 8-bit XOR
- tri-state buffer



The datapath must have at least:

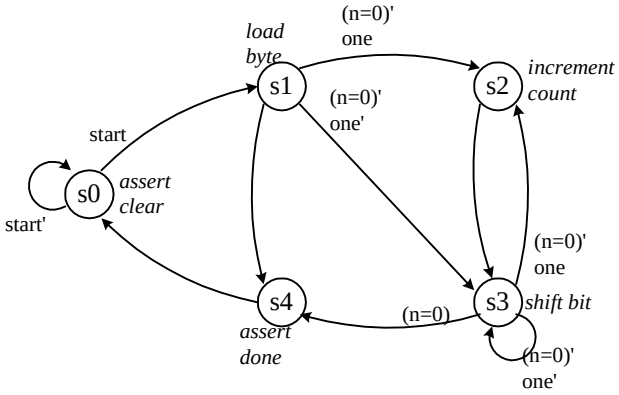
1) a register with load signal to load in the byte.
2) a done signal to output the parity bit.

No points if datapath looks similar to the general datapath with ALU, RF, mux, etc.

3. How many states are required by the FSM to control the datapath in question #2? Briefly comment on what each state is used for (like the comments or pseudo code that we put for each control word). You don't need to do the control words nor the FSM. (4)

Answer

State diagram for version 1 above.



4. The even parity circuit from question 2 can also be implemented with a combinational circuit. Draw this combinational circuit, which has 8 inputs and 1 output. (4)

Answer

An 8-input XOR gate.

5. Can we use tri-state buffers for output ports and latches for input ports? Give your reasons as to why yes or why not. (4)

Answer

- We CAN use tri-state buffers for output ports IF the device is fast enough to read the data. Usually, the device is not fast enough.

- We CANNOT use latches for input ports because a latch always outputs a 0 or a 1 and never the Z value. Without the Z value, the data from this input port will collide with the traffic on the data-bus.