LAB-4

Due to:
Part 2: Sunday, May 1, 2016, 23:55hrs

Submission: via COW

Part 2: Cafeteria Credit System (Teamwork)

(This part of the lab will be performed and submitted with your group partner. Each group should submit only a single copy.)

Problem Definition

In this part of the homework, you are required to develop a system which will be used by a charity. The aim of the charity is to choose some students who need help and to give them some credit to use in the cafeteria.

The system that you are going to develop should have the following specifications:

1) In this system, we should be able to *deposit credits* to student accounts, *withdraw money* from the accounts, *add new students* to the system and *list accounts*.

2) The system assists at most 7 students.

3) Each student has a 3 bit unique id and no studentID can be 000.

4) We store the sum of all student credits (**totalCredits**).

5) There are 4 modes in this system:

   a) Add New Student(00)

   A student is added to the system with the given id and zero credit. If a student is already in the list, then no action will be performed. Otherwise, student list is updated and the studentCount is increased by 1.

   b) Withdraw Credit(01)

   Students use their credits in the cafeteria for paying the meals. In this mode, if the price of the meal is less than the remaining credits, it is decreased from the student's account. Otherwise no action is performed.

   If a student uses more than **4** credits within an hour he will be added to the **suspended list**.

c) Deposit Credit(10)

In this mode, the credits of all student accounts except for the accounts that are in suspended list are increased by the given value. After transfering money to the accounts, suspended list will be cleared. In other words, the accounts that were in the suspended list will be allowed to deposit credit on the future transactions.

d) List Students(11)

The charity system periodically checks student accounts that are lower than a given treshold. In this mode, ids of students whose remaining credit is less than the given value are listed. You should list one id per CLK. After listing the ids that satisfies the condition, the system should give a warning(end_of_list) and stop listing.

6) Initially, the clock will be 0.
7) The hour format is in 24-hours clock; 0 up to 23.
8) Initially, the student id list will be empty.
9) After every deposit andwithdraw operation, **totalCredits** should be updated to reflect the sum of all student credits. You can assume that **totalCredits** will not be more than 7'b1111111 (127), i.e. we will not deposit credits that will cause overflow.

Input/Output Specifications

Name	**Type**	**Size**
studentID	Input	3 bits
credit	Input	3 bits
mode	Input	2 bits
incTime	input	1 bit
CLK	Input	1 bit
stime	Output	5 bits
idOutput	Output	4 bits
endOfListWar	Output	1 bit
studentCount	Output	4 bits

totalCredits	Output	7 bits

- **studentID** represents 3-bit Student id value.
- **credit** represents a 3-bit credit amount.
- **mode** is used for the selection of the following tasks:
  - *mode = 00* → Insert a student with the given studentID into the system.
  - *mode = 01* → The credits are decreased from the account of the given student.
  - *mode = 10* → The given amount of credit is transferred to all student accounts except for the accounts that are in the suspended list.
  - *mode = 11* → List ids of students whose remaining credit is less than the given value.
- **incTime** is the used to increment the hour by 1. There should be no deposit/withdraw/list operation when incTime equals to 1. It will be triggered by pressing Btn[1] and Btn[0] at the same time.
- **CLK** is the clock input for the module.
- **stime** shows the current time. After 23, it will continue with 0 and start over.
- **idOutput** shows the id of students sequentally in "List Students" mode.
- **endOfListWar** indicates that the listing is finished. It becomes "1" after a clock cycle following the last student id and turns to 0 after a clock cycle again.
- **studentCount** shows the count of the students in the list.
- **totalCredits** is the total amount of all student credits.

Sample Input/Output

Current State	**CLK (BTN0)**	**Next State**	**Explanation**
101 ,x ,00 ,0 ,0 ,0 ,0 ,0	↑	0 ,0 ,0 ,1 ,0	insert id=5 to the list
011 ,x ,00 ,0 ,0 ,0 ,1 ,0	↑	0 ,0 ,0 ,2 ,0	insert id=3 to the list
110 ,x ,00 ,0 ,0 ,0 ,2 ,0	↑	0 ,0 ,0 ,3 ,0	insert id=6 to the list

110 ,x ,00 ,0 ,0 ,0 ,3 ,0	↑	0 ,0 ,0 ,3 ,0	try to insert a previously added id(6) to the list
xxx ,7 ,10 ,0 ,0 ,0 ,3 ,0	↑	0 ,0 ,0 ,3 ,21(0010101)	7 credits are added to all of the accounts
110 ,3 ,01 ,0 ,0 ,0 ,3 ,21	↑	0 ,0 ,0 ,3 ,18(0010010)	3 credits withdrawn from the student 110(6)
xxx ,x ,xx ,0 ,0 ,0 ,3 ,18	BTN1+ ↑	1 ,0 ,0 ,3 ,18	increase time by 1 hour
101 ,3 ,01 ,1 ,0 ,0 ,3 ,18	↑	1 ,0 ,0 ,3 ,15(0001111)	3 credits withdrawn from the student 101(5)
101 ,2 ,01 ,1 ,0 ,0 ,3 ,15	↑	1 ,0 ,0 ,3 ,13(0001101)	2 credits withdrawn from the student 101(5). Student is added to suspended list
110 ,3 ,01 ,1 ,0 ,0 ,3,13	↑	1 ,0 ,0 ,3 ,10(0001010)	3 credits withdrawn from the student 110(6)
xxx ,4 ,11 ,1 ,0 ,0 ,3,10	↑	1 ,5 ,0 ,3 ,10	the first student(id=5) satisfying the condition(credit<4) is returned
xxx ,4 ,11 ,1 ,5 ,0 ,3,10	↑	1 ,6 ,0 ,3 ,10	the second student(id=6) satisfying the condition(credit<4) is returned
xxx ,x ,11 ,1 ,6 ,0 ,3,10	↑	1 ,0 ,1 ,3 ,10	the end of the list is reached
xxx ,x ,11 ,1 ,0 ,1 ,3,10	↑	1 ,0 ,1 ,3 ,10	endOfListWar stays in 1 in subsequent CLK operations until mode is changed to another value.
xxx ,2 ,10 ,1 ,0 ,1 ,3,10	↑	1 ,0 ,0 ,3 ,14(0001110)	student in the suspended list(id=5) does not gain 2 credit money in the first deposit

xxx ,2 ,10 ,1 ,0 ,0 ,3 ,14	↑	1 ,0 ,0 ,3 ,20(0010100)	2 credits are added to all of the accounts

- The values in ***Current State*** column, which are seperated by "," are defined as: **studentID**, **credit**, **mode**, **stime**, **idOutput**, **endOfListWar**, **studentCount**, **totalCredits**, respectively.
- The values in ***Next State*** column, which are seperated by "," are defined as: **time**, **idOutput**, **endOfListWar**, **studentCount**, **totalCredits**, respectively.
- "x" denotes "don't care" value.

FPGA Implementation

Name	Type	Size
studentID	SW7, SW6, SW5	Left-most 3 switches (A)
credit	SW4, SW3, SW2	Middle 3 switches (B)
mode	SW1, SW0	Right-most 2 switches (C)
incTime	BTN1 + BTN0	Right-most two buttons (N + O)
Clock (CLK)	BTN0	Right-most button (O)
stime	7-segment display	Left most 2, 7 segment display (K)
idOutput	7-segment display	Middle 7 segment display (L)
endOfListWar	LD0	Right-most LED (E)
studentCount	7-segment display	Right most 7-segment display (M)
totalCredits	LD7-LD1	Left most 7 LEDs (D)

Deliverables

- Implement your module in a single Verilog file: lab4_2.v. Do NOT submit your testbenches, or project file (.xise). You may share your testbenches on the newsgroup.
- Submit the file through the COW system before the given deadline. May 1, 2016, 23:55hrs
- This part is supposed to be done with your group partner. Make sure both of you take roles in implementation of the project. Any kind of inter-group cheating is not allowed.
- Use the newsgroup metu.ceng.course.232 for any questions regarding the homework.
- You will make a demo with the FPGA board the next week after the submissions (in your lab session hours). The exact dates and place will be announced later.