

Student ID: \_\_\_\_\_

# CS 151 Final

Name : \_\_\_\_\_ , \_\_\_\_\_  
(Last Name) (First Name)

Student ID : \_\_\_\_\_

Signature : \_\_\_\_\_

## **Instructions:**

1. Please verify that your paper contains **15 pages** including this cover.
2. Write down your Student-Id on the top of each page of this final.
3. This exam is **closed book**. No notes or other materials are permitted.
4. Total credits of this final are **100 + 10 EXTRA CREDIT**.
5. To receive credit you must show your work clearly.
6. **No re-grades will be entertained if you use a pencil.**
7. Calculators are **NOT** allowed.

	Q1	Q2	Q3	Q4	Q5	Total
Credit	25	15	30	15	25	110
Score						

Student ID: \_\_\_\_\_

**Q1: Design of a Moore Machine**

**[25 points]**

We want to design a non-resetting sequence detector using a finite state machine with one input X and one output Y. The FSM asserts its output Y when it recognizes the following input bit sequence: "**1101**". The machine will keep checking for the proper bit sequence and does not reset to the initial state after it has recognized the string.

[Note: As an example the input string X= "..1101101.." will cause the output to go high twice: Y = "..0001001.."]

(a) Capture the Moore FSM.

(10points)

Student ID: \_\_\_\_\_

(b) Create the architecture

(3 points)

Student ID: \_\_\_\_\_

(c) Encode the states (use a simple binary encoding)

(2 points)

(d) Create the state table

(10 points)

Student ID: \_\_\_\_\_

**Q2: [RTL Design]**

**[15 points]**

We want to design a timer with 2 modes: **1Min**, **30Sec**. There is a **START** button and a **STOP** button for this timer. The timer is initially in **standby** mode. Then, we select a mode by pressing **1Min** or **30Sec**. After pressing the button **START**, the timer starts counting until it reaches the time (if it is in **1Min** mode it counts for 1 minute and if it is in **30Sec** mode, it counts for 30 seconds). Then it beeps and goes to **standby** mode. If during the time that the timer is working we press the **STOP** button the timer beeps and it goes to **standby** mode. If the timer is working in **30Sec** mode and the user presses **1Min** button, then after completion of 30 seconds the timer works for another 30 seconds. This can happen just once and then the timer beeps and goes to **standby** mode.

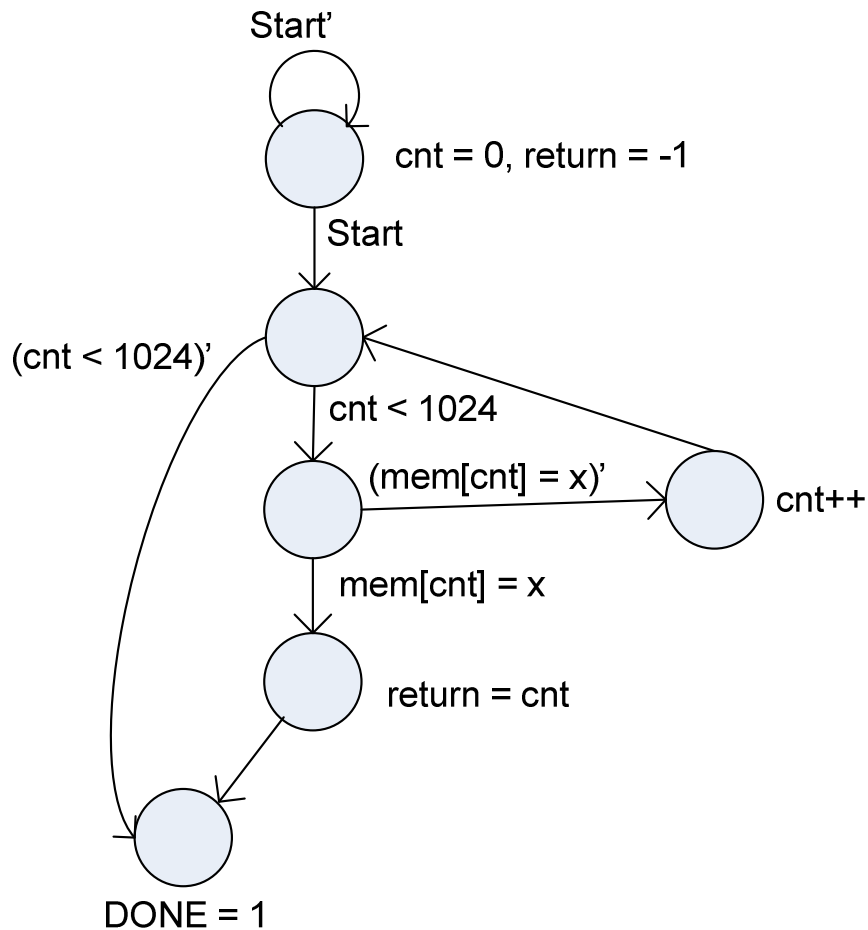
Assuming that we have necessary components to design the circuit for this timer, design the high-level state machine for this timer.

Student ID: \_\_\_\_\_

**Q3: [Data Path Design]**

**[30 points]**

The following high-level state machine is designed to find an input  $X$  in a register file of size 1024.  $X$  is the input that we are searching for in the register file,  $START$  is a one bit input and  $RETURN$  and  $DONE$  are the outputs.  $Mem[t]$  is the value stored in the register file at address  $t$ .



Student ID: \_\_\_\_\_

1. Design the data path for this high-level state machine. [15 points]

Student ID: \_\_\_\_\_

2. Design the interface of the controller and the data path as well as the interface of the system. [5 points]

Student ID: \_\_\_\_\_

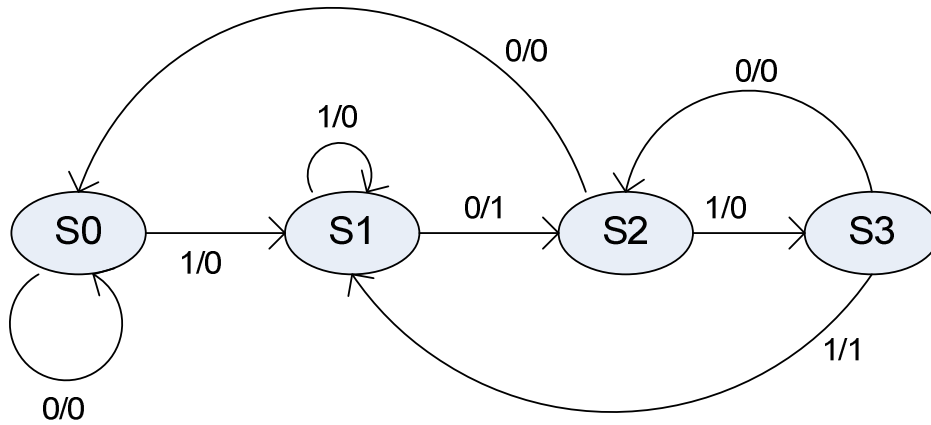
3. Design the controller's FSM. [10 points]

Student ID: \_\_\_\_\_

**Q4: [FSM Timing]**

**[15 points]**

The following FSM has one bit input I and one bit output O. (I/O is presented on each edge of the FSM).

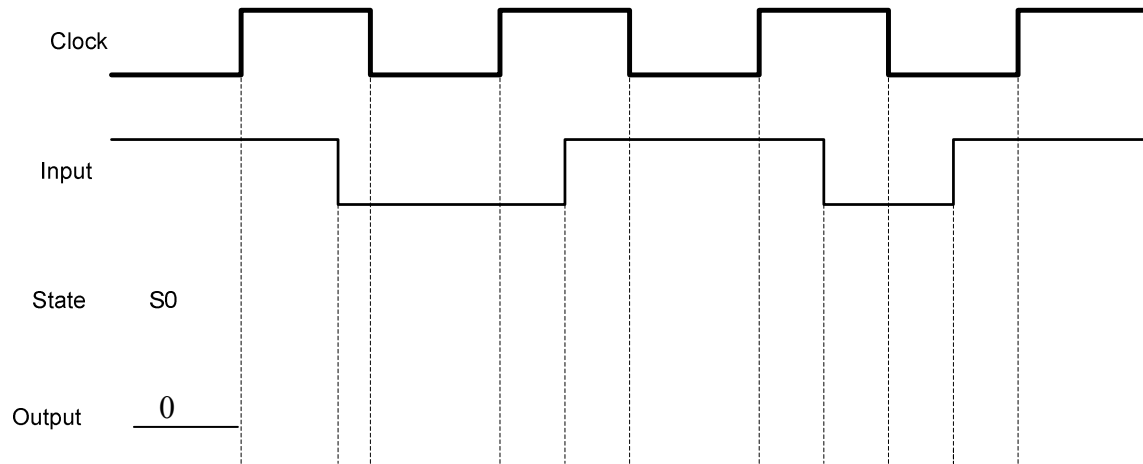


1. Determine if it is a Mealy machine or a Moore machine. [2 points]

Student ID: \_\_\_\_\_

2. In the following timing diagram show how the state and the output change with input. Assume that we initially start from state S0. [13 points]

NOTE: Assume that delay of the gates and the register is negligible.



Student ID: \_\_\_\_\_

**Q5: [Component Design] \_\_\_\_\_ [25 points]**

1. We want to design a **frequency divider** circuit that divides a given frequency by 3 or 5. If the input MODE=0 it divides the frequency by 3 and if MODE= 1 it divides the frequency by 5. [15 points]

Use as many of the following as needed in your design:

- a- 3-bit counter with clear
- b- 2-to-1 multiplexer
- c- Logic gates (AND, OR, NOT)

Student ID: \_\_\_\_\_

2. Design a 16-to-1 multiplexer using ONLY 4-to-1 multiplexers. [10 points]

Student ID: \_\_\_\_\_