DigitLabwork#6

Operation of An Elevator-in-Lab

[1] Training goals

[2] Preparations

[3] Lab-task instruction

[4] Observations/Evaluation

= =

[1] Training goals

Through the lab-work, class members are expected to get acquainted with the following matters.

a)) Issues concerning proper control over elevator operations and how the control should be done via the use of sequential logic or FSM.

** operation of one single elevator with primitive controls;

b)) the use of Verilog for logic circuit description and simulation, including

** circuit module building up in Verilog,

** setting up of the test data;

c)) the interactive commands required in operating the Verilog-code development system:

** compilation for a syntax error-free Verilog description and test data set;

** simulation of the Verilog-coded circuit module;

[** synthesis of the Verilog-coded circuit module].

[2] Preparation

Every class member should get prepared with the following knowledge prior to attending the lab-work sessions.

a)) operation of counters.

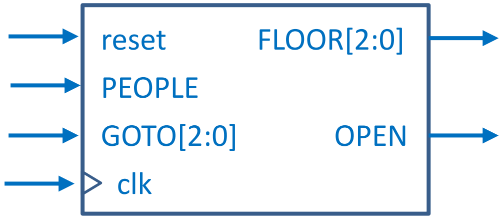
b)) skills of generating required outputs during state transitions in the operation of a sequential circuit.

[3] Lab-task instruction

In digitlabwork#5, 1 task is assigned to every class members as given below.

[TASK1]

(1) Write a Verilog code for elevator operation under simplified controls in compliance with the requirements given below.



(2) Put the codes under simulation; observe and interpret the waveforms of output signals

(3) Try drawing the state transition diagram (or FSM) for the elevator operation of your own design,

(4) Try drawing the circuit diagram of your design using counters, logic gates and components one deems as necessary.

a)) circuit operational specifications

** All signals are high-active

** All inputs arrive at the falling edge of the clock

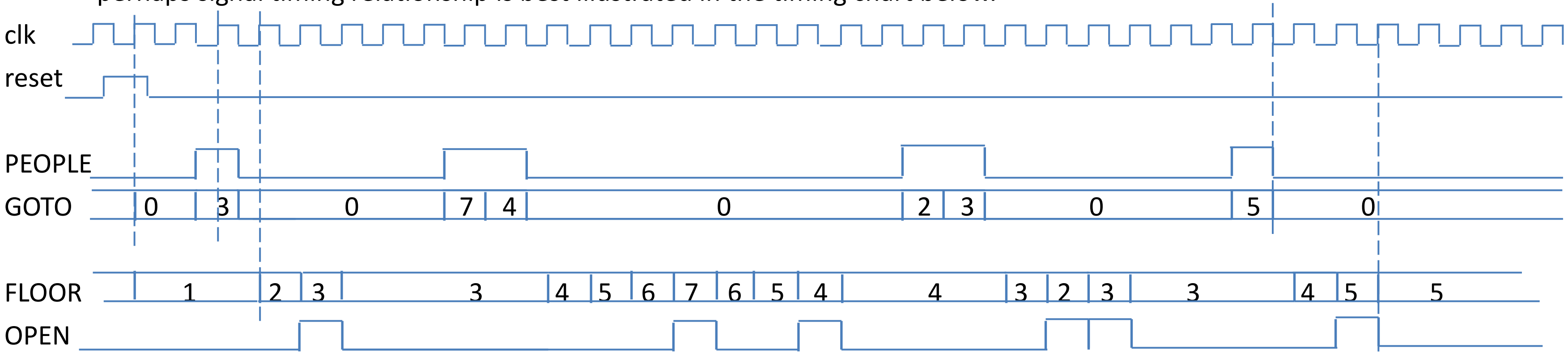
** All outputs appear at the rising edge of the clock

** I/O layout:	inputs	reset, clk, PEOPLE, GOGO[2:0];
	outputs	OPEN, FLOOR[2:0];

** signal implications:

signal	implications
reset	** initiation of the system at clock-synchronization; ** appears only once for every power-up operating cycle
PEOPLE	** indicating the situation when the user depresses destination-floor button after entering the car; ** may last for 1 clock or two when becoming active; ** at the rising edge of the clock following the high-to-low transition of PEOPLE, the elevator starts moving up or down at the rate of 1 floor level per clock;
GOTO[2:0]	** indicating that a destination-floor button, ranging from 1 to 7, is depressed by the user in the car; ** valid only when PEOPLE is active, 0 otherwise; ** in the case that PEOPLE remains active for two clocks, two distinct or same GOTO values are allowed as destination-floor requests, first-come-first-served;
clk	** all inputs supplied at falling edge of the clock; ** the system samples inputs at rising of the clock and reacts accordingly;
FLOOR[2:0]	** indicating the floor level which the car is currently standing on or passing through; ** starts changing at the 1st clock following the falling edge of PEOPLE, either increasing or decreasing by 1 per clock;
OPEN	** indicating the open/close of the doors of the elevator; ** remains active for 1 clock when the elevator reaches the floor level designated by GOTO value;

** perhaps signal timing relationship is best illustrated in the timing chart below.



b)) coding specifications:

** write behavior-level descriptions for the elevator operations of your own design, fulfilling the operational protocol given below.

** the module has reset, clk, PEOPLE and GOTO[2:0] as inputs, and FLOOR[2:0] as outputs.

** elevator operation protocol

Event	reactions
at receiving reset (reset (clk^))	entering initial state: 1) FLOOR: 1 2) OPEN: 0 3) initiation for internal use
at receiving PEOPLE (PEOPLE (clk^))	1) recording GOTO value 2) recording one more GOTO value if PEOPLE remains valid for the 2nd clock 3) in the case of receiving two GOTO values, first-come-first served policy is undertaken 4) FLOOR starts updating at the rising edge of the clock following the high-to-low transition of

	PEOPLE, either increasing or decreasing by 1 per clock until one or both designated-floor level(s) reached in order. 5) when reaching the designated-floor level, for the case of receiving 1 GOTO value *)) OPEN becomes active for 1 clock, *)) the elevator then stands still on the very floor level , waiting for further PEOPLE input otherwise, *)) OPEN becomes active for 1 clock, *)) the elevator then, in the following clock, starts moving toward the floor level designated by 2nd GOTO value, *)) OPEN becomes active for 1 clock when reaching the floor level designated by the 2nd GOTO value, *)) the elevator then stands still on the very floor level .

** Timing: no delay(s) to be considered in this task

c)) testing data:

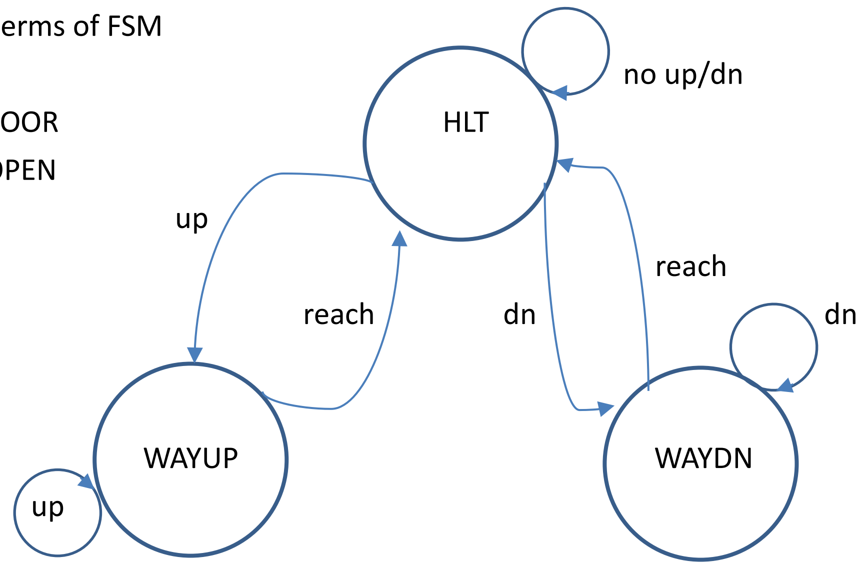
to be supplied by TAs.

d)) some tips

** thinking in terms of FSM

outputs: FLOOR

OPEN



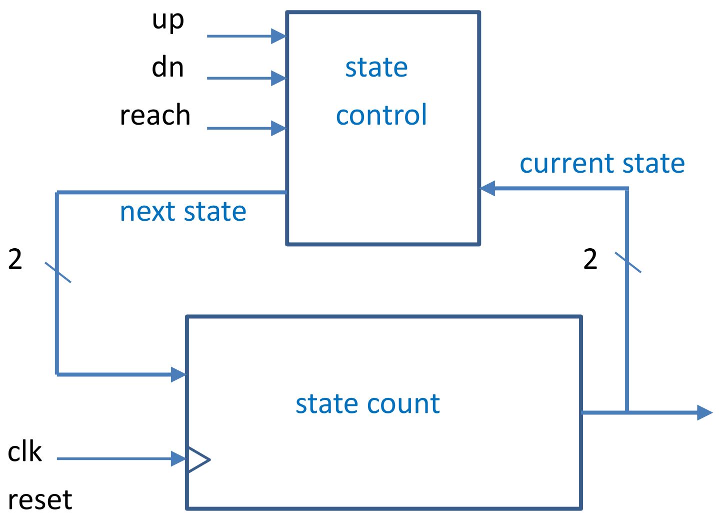
** other FSM forms?

[4] Observations

** Imaging how messy things can go when two elevators running in conjunctive operation in a building 7-level above the ground and 1-level under.

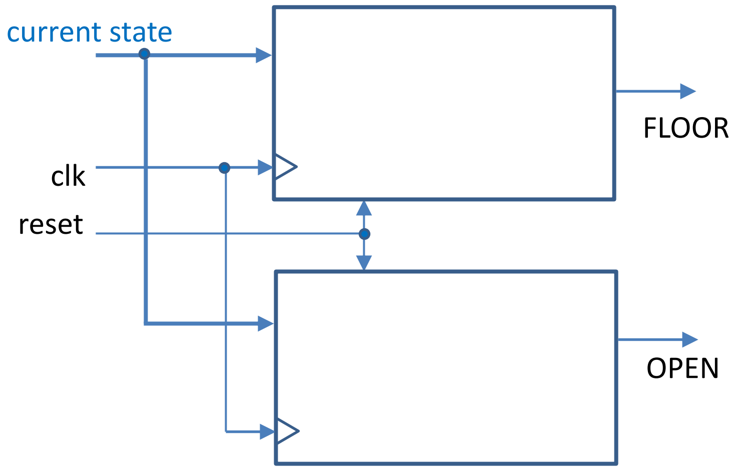
A prototype from scratches

1)) the state-transition module

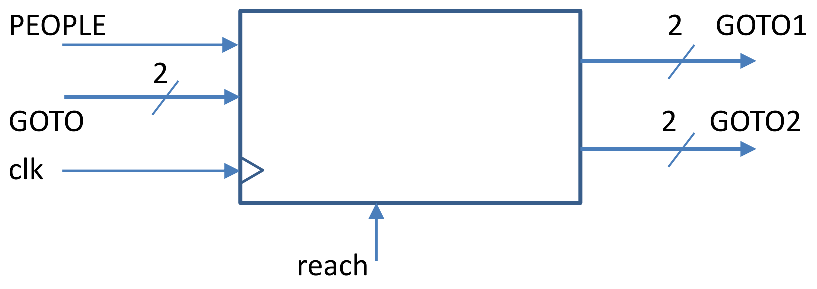


state	
00	HLT
01	WAYUP
10	WAYDN
11	x

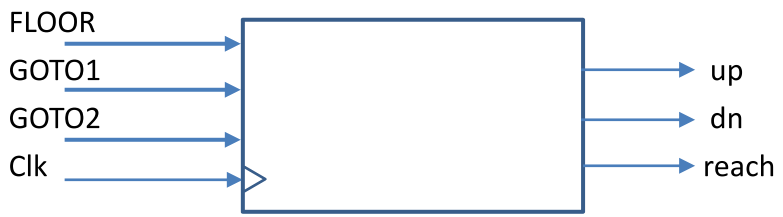
2)) output module FLOOR and OPEN



3)) input module



4)) up/down/halt lifting control


FLOOR
GOTO1
GOTO2
Clk
up
dn
reach