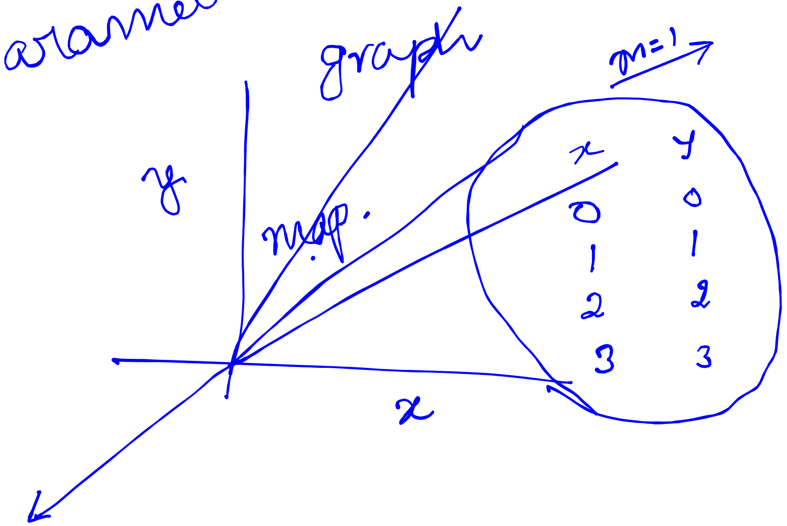
parametric plot

mx



x	y
0	0
1	1
2	2
3	3

Tabulated

2 variables

3 dimensional

Karnaugh map

Reduction and Simplification

Now I need to reduce this logic representation to a logic statement that can lead to logic gates.

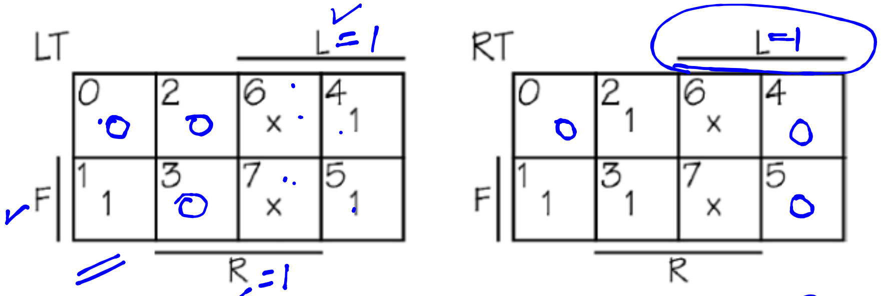
- To reduce and simplify the logic statements, the *Karnaugh Map*, which is a graphical way of doing the job, is used.
- This is just one way that we could do the job. Figure below shows the Karnaugh maps.

don't care

Table

N0.	L	R	F	LT	RT
0 ✓	0	0	0	0	0
1 ✓	0	0	1	1	1
2 ✓	0	1	0	0	1
3	0	1	1	0	1
4	1	0	0	1 ✓	0
5	1	0	1	1 ✓	0
6	1	1	0	X	X
7	1	1	1	X	X

Map



4 dimn function → $LT = L + F.\bar{R}$ Karnaugh maps

$L=1 \quad LT=1$

$F=1$ & $R=0$

$F.\bar{R} \to 1$

$RT = R + F.\bar{L}$

→ easier reading of

We will do this without explanation --- We will take it up later.

Table

Reduction and Simplification

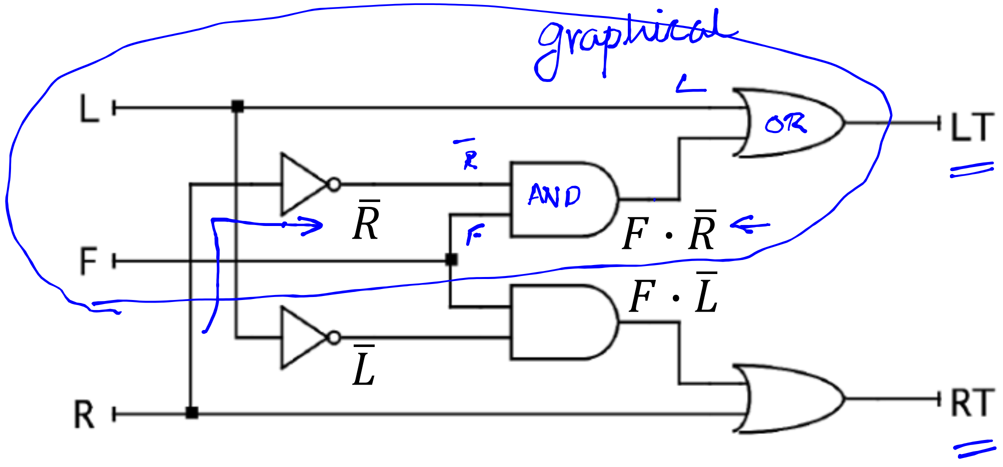
The less obvious part is how we get the following logic equations from them:

$$\begin{cases} LT = L + F \bullet \bar{R} \\ RT = R + F \bullet \bar{L} \end{cases}$$

- Actually, the equations are quite readable if you know that the + means *or*, the · means *AND*, and the bar over a symbol means to take the logical complement i.e., if L = 1, L with a bar over it is 0, and vice versa).
- Read the equation for LT: the left-turn bulb is to light if the left-turn switch (L) is actuated, or if the emergency-flasher switch (F) is actuated and the right-turn switch (R) is not actuated.
- Try this for the other equation.
- But the real test of all this is to implement it.

Circuit Implementation

Figure below is a drawing of the entire circuit.



$LT = \boxed{L + \boxed{F \bullet \bar{R}}}$

$RT = R + F \bullet \bar{L}$

Combinational circuit

The circuit is organized in exactly the same manner as the equations. There are only six gates and just three different kinds:

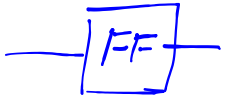
- Two Inverter, Two *AND* gates and two OR gates.

control

hold

0

1



1 or 0

no change

0 (0→0 ; 1→0)

1 (0→1 ; 1-1)

WHAT'S A SEQUENTIAL CIRCUIT?

- The basic sequential device is a flip-flop, which is the simplest form of memory.
  - ➢ The flip-flop *stores* a 1 or a 0. Applying the proper signals to its inputs causes the stored value to change—if it is a 1, it becomes a 0, for example.
- We can combine flip-flops into fancier sequential circuits such as the counter. counter time watch
  - ➢ This device counts the number of binary 1s presented to it over time on a particular input. Its "contents" are a binary number composed of several bits (binary digits).

SEQUENTIAL EXAMPLE

rt. turn   RT → RB blinks

R = 1
ON
OFF
ON
OFF
} rate

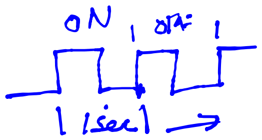
Now let's make the lamps flash – as in real practice. We want them to flash at a reasonable rate when activated.

Specifications

Either lamp is to flash at an appropriate rate when the signal to activate the lamp is received.

- The LT and RT signals from the previous design are to be the inputs to the new circuit.
- The lamps are to flash at the rate of one flash every two seconds. A clock signal (Clk) will be available that "ticks" every second.   1 c/s

  Keeps Time
- The outputs to the light bulbs are to be LB and RB; bulbs light when the signal is 1.



Notice that the clock is an external signal that provides the timing. There are circuits that generate clock ticks but I'm not going to worry about them now.

State Diagram

We will develop the design for the left-turn bulb; the design for the right-turn bulb will be identical except for signal names.

The truth table was one way to describe the combinational circuit in complete detail.

The *state diagram* is a way to do this for a sequential circuit. This diagram shows the transitions from one state to another.

Bulb ON → 1st State
Bulb OFF → 2nd state

2-states ON & OFF

In my design, one state is with the bulb lit and the other state is with the bulb dark. These two states will appear in my state diagram.

LT & RT

if LT=1 LB=1 → combinational

periodically turn ON &OFF

State Diagram

The state diagram for the left-turn lamp is show below.

Writing on the arc describes the condition

movement is every clock tick

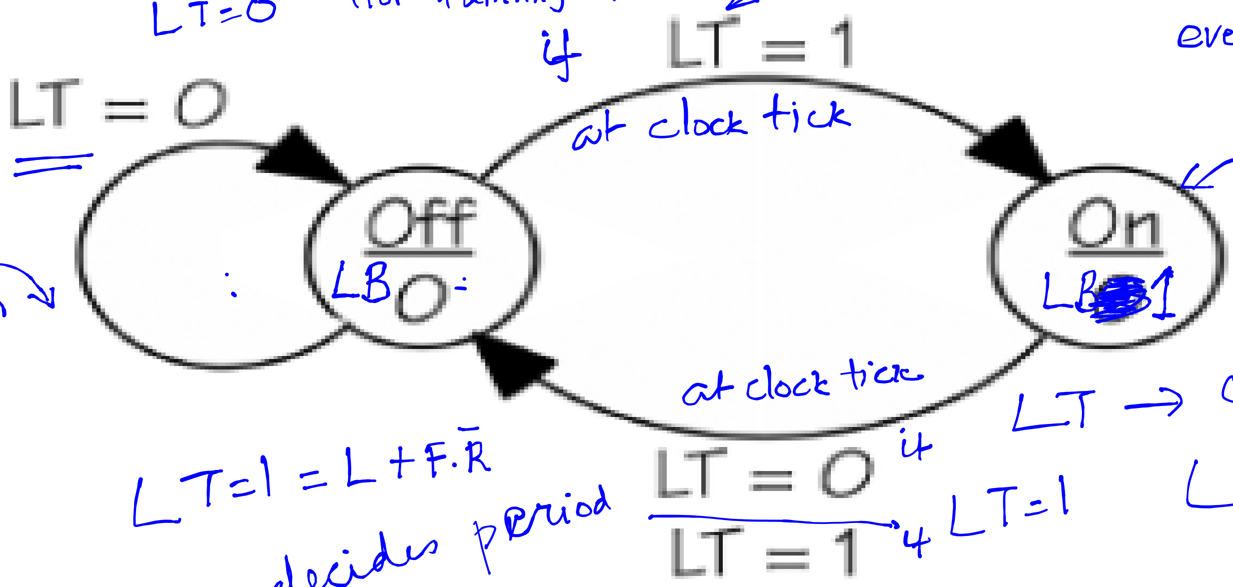
LT=1 turning left
LT=0 not turning left

LB switch

Variable LT, ~~RT~~ L, R, F → 3 LB

LT → if
RT → 1
F → 1

if



flashing

ON|OFF|ON/OFF/...

State diagram

LT → control for signal

LB light RB

LT=1 = L+F.R̄

it

4 LT=1

IIIT

clock decides period

timing → clocking

LT=0
LT=1

LB is always OFF
LB is periodically ON

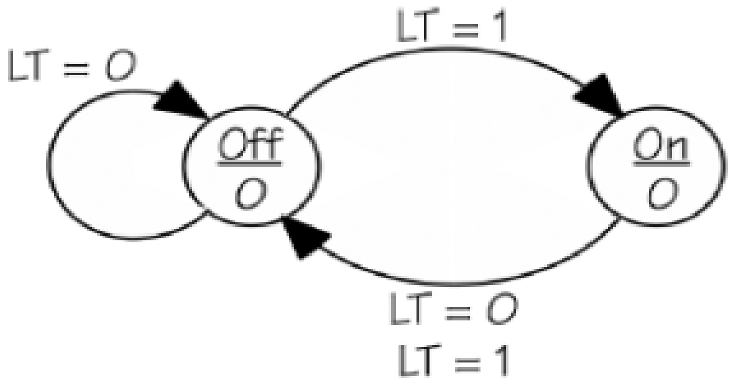
Flashing

State Diagram

There are three things to notice:

- The input to this diagram is the left-turn signal from the combinational circuit we just designed. It always appears as LT written along the lines that show the transitions between states.
- The circles are the states: the name of the state is above the line and the binary output is below the line. The "Off" state has an output of LB = 0, the "On" state has an output of LB = 1.
- A transition along one of the paths is made *only* when the clock tick comes along.

State Diagram



This diagram is not hard to read.

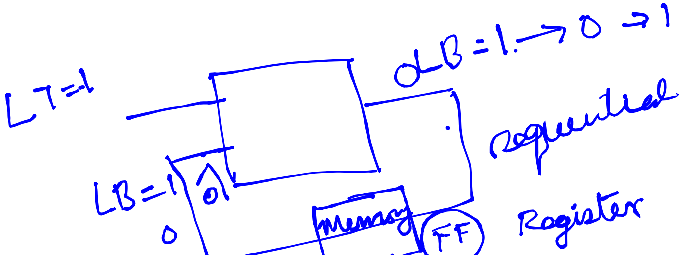
Suppose, for example, that the circuit is in the "Off" state, which means that the most recent tick of the clock left it there.

Suppose that the input LT = 0. At the next tick of the clock, the state will not change because the transition arrow for LT = 0 ends on the "Off" state.

Suppose that the circuit is still in the "Off" state and the input is now LT = 1. At the next clock tick, the state will change to the "On" state.

Follow the rest of the transitions to see it has been done right!

Transition/Excitation Table



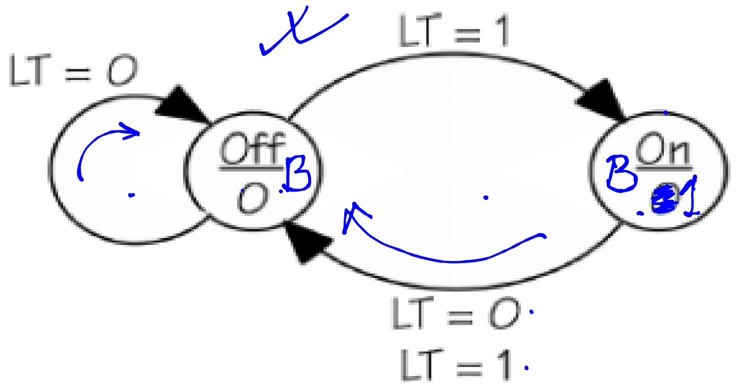
This information can also be expressed via a State Table, as shown below

clock

$LF(L,F,R) = L + F\cdot\bar{R} = LT$

L, Excitation Table

Current State	Input LT	Next State
OFF	0	OFF
OFF	1	ON
ON	0	OFF
ON	1	OFF



Some designers use diagrams, some use tables, but most use a combination of these.

Now we need to reduce this to a more "binary" design.

Curr. State	LT	Next State
OFF	1	ON
OFF	0	OFF
ON	1	OFF
ON	0	OFF