MICROPROCESSOR LABORATORY

African Regional Course on Advanced VLSI Design Techniques

24 November - 12 December 2003

Kumasi - Ghana

Project

Design of a Programmable Traffic Signal Controller

PROGRAMMABLE TRAFFIC SIGNAL CONTROLLER

INTRODUCTION

In any city, the streets constitute a complex urban network and there are many "traffic signal" nodes in this network, in such a way that they put some order to the traffic increasing the safety and the "efficiency". The concept of "efficiency" is not very well defined, but everybody has an intuitive idea about what "traffic efficiency" means. To define accurately what "Vehicular Traffic efficiency" is, it is important to establish what the objective parameters are that permit to us to evaluate the quality of the vehicular traffic. These are parameters that we should be able to measure. The problem doesn't finish here because every individual interested in "efficiency" and "optimality" of the network expects a different thing. For example if we take as parameter of quality of the traffic like the average velocity of the cars in the urban network, the drivers would like it to be high, but the pedestrians will like it to be low for security reasons. This example shows that the problem is not only technical but also political, in the sense that a city administration may decide the definition of optimum.

However, once a criterion is fixed to evaluate the quality of the vehicular traffic, it is important to have the means to bring the traffic towards an optimal condition. Among the means to reach that situation, we have the "traffic signal" (TS), the experiences of which indicate its extreme importance. Then the quality of the vehicular traffic is sensitive to how the "traffic signals" are configured. We will assume that configuration of the "traffic signal lights", as the set of parameters that characterises completely the state of the traffic signals of the network.

There are normally two modes in which the light traffic works. They are the intermittent yellow and the cyclic mode that alternates between yellow, red and green. In the last case, the colour is a periodic function of the time. We need four parameters to characterise this function: the duration of each colour (3 parameters), and the phase of the signal, e.g. the instant in which the yellow, for a determined street, starts.

We assume a simple traffic light (in the sense that it regulates only two crossing streets) and of equal duration for yellow on the two streets. With this assumption, we propose a "programmable light traffic controller". This device is capable of receiving information containing the working mode and the colour's duration, which is updated when a synchronisation signal arrives. This controller is capable of avoiding dangerous and traumatic situations of discontinuity in the traffic. The instant at which **syncro** (synchronisation signal) arrives, fixes the phase of the traffic light. With the signal **syncro**, also arrive the working mode (**mode**) and the colour's duration: **tyel**, **tred** and **tgre** (duration of yellow, red and green respectively).

We are looking for a device with the external ports as shown in Fig. 1. Table 1. shows how the internal registers can be set-up for operation. Table 2. essentially gives the output associated with each of the states of the state machine. The state machine flow diagrams are described in the following pages.

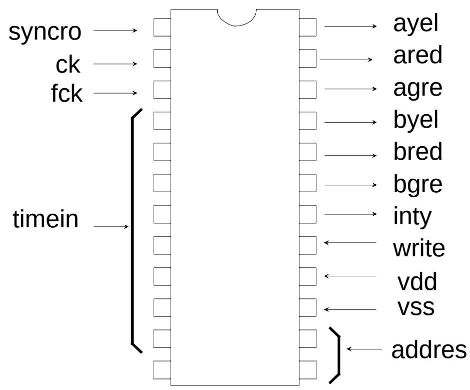


Fig. 1. Possible pinout of the Programmable Traffic Controller Chip

FCK	write	address	tyel(7-0)	tred(7-0)	tgre(7-0)	mode(1-0)
L to H	L	X	X	X	X	X
L to H	H	'00'	timein(7-0)	X	X	X
L to H	H	'01'	X	timein(7-0)	X	X
L to H	H	'10'	X	X	timein(7-0)	X
L to H	H	'11'	X	X	X	timein(1-0)

Table 1. Setting up of the internal registers.

CK	MODE	STATE	duration (In number of periods of 'ck')	ayel	ared	agre	byel	bred	bgre	inty
L to H	'01'	INTYEL	permanent	L	L	L	L	L	L	H
L to H	'00'	YELLOW	tyel	H	L	L	L	H	L	L
L to H	'00'	RED	tred	L	H	L	L	L	H	L
L to H	'00'	REDINT	tyel	L	H	L	L	L	H	L
L to H	'00'	REDYEL	tyel	L	H	L	H	L	L	L
L to H	'00'	GREEN	tgre	L	L	H	L	H	L	L
L to H	'10'	RED	permanent	L	H	L	L	L	H	L
L to H	'11'	GREEN	permanent	l	L	H	L	H	L	L

Table 2. The outputs that are associated with each state

We will divide the complete architecture in three parts: **(1)** a synchronous variable mod counter, **(2)** the registers of the data (times plus mode) controlled by an address (**address**) and an external fast clock (**fck**), and **(3)** the logic part to control the state of the lamps.

The Algorithmic Finite State Machine

We start with some definitions:

- The device has clock (**ck**) as input, its period will be taken as the unit of time in which the duration will be expressed, and a fast clock (**fck**) used to introduce the data into the internal registers.

- The state of the TS is defined by the state of each colour (**ayel**, **ared**, **byel**, **bgre**, etc.) and its duration (**timef**).

- **mode**:(1 downto 0), indicates the working mode.

  mode <= '00' normal cyclic mode
  mode <= '01' intermittent yellow
  mode <= '10' permanent red (in a determined direction)
  mode <= '11' permanent green (in the same determined direction)

- **timef** : (7 downto 0) , indicates the normal duration of the state. In some cases that the FSM (finite state machine) goes into a new state, the counter is reset to zero putting the signal **cntreset = '1'** . When the counter reaches the value of **timef** the flag **cntflag** is raised (**cntflag = '1'**) and this fact will be used to decide about the change of the state. The values that **timef** can assume are: **tyel**, **tred** or **tgre**.

- For the street A
  ayel <= '1' means "on" , '0' means "off"
  ared <= '1' means "on" , '0' means "off"
  agre <= '1' means "on" , '0' means "off"

  The same for the street B with **byel, bred** and **bgre**
  and for both streets
  inty <= '1' means "on" , '0' means "off"

- For us the lights **ayel** , **ared** and **agre** are mutually exclusive (in Trieste the style is **ayel** and **agre** at the same time slightly before the end of **agre,** and remains till the end when the signal changes to red). The same for the other street. The signal **intyel** is incompatible with the other lights.

- The situation : **agre** <= '1' & **bgre** <= '1' is absolutely forbidden.

- The change from green to red must be done by means of an intermediate yellow

- Each time that the configuration is updated, the new parameters must be simultaneously provided to the chip along with a **syncro** signal with duration of one clock period (**ck** single pulse).

There are six States of the Finite State Machine (SFSM) called: INTYEL, YELLOW, RED, REDYELL, GREEN and REDINT. In each SFSM the colours have a specified constant state ('on' or 'off'), then they change only if the state changes.

In the Fig. 2 below, we represent the colours on each street, as a function of time, with the different cases in which the synchronisation signal can arrive.

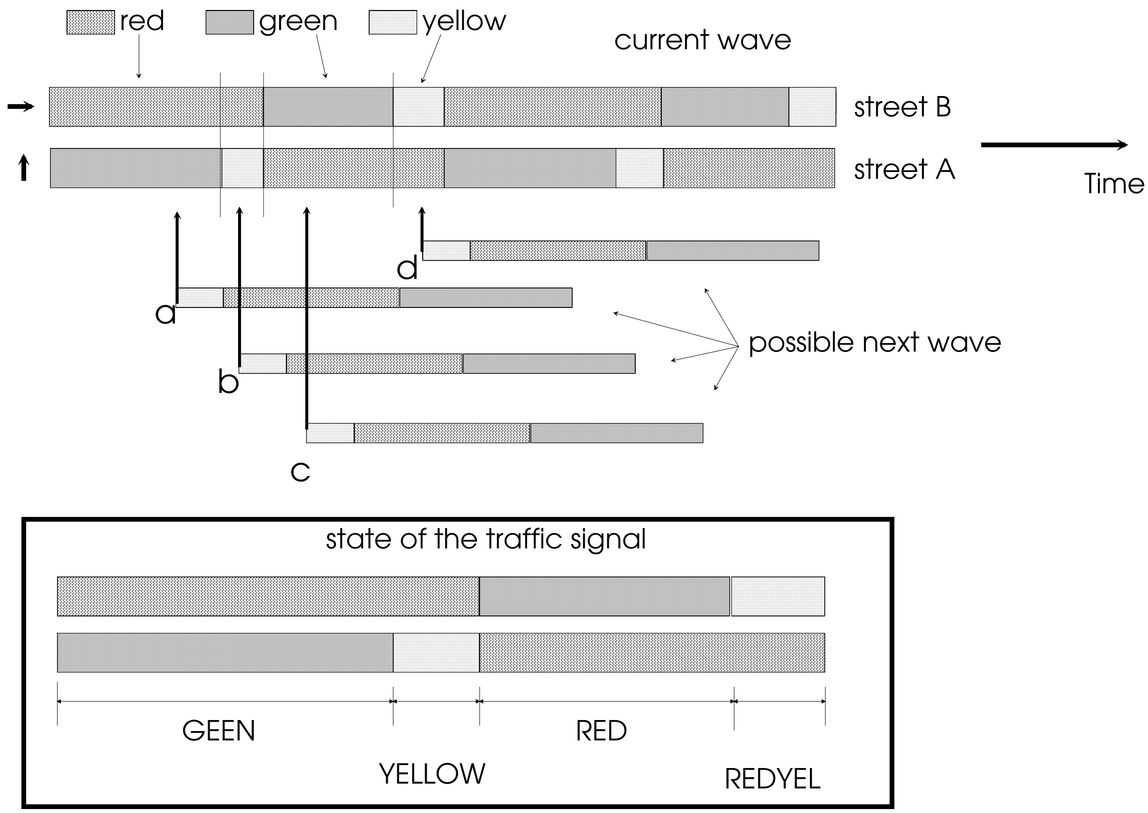
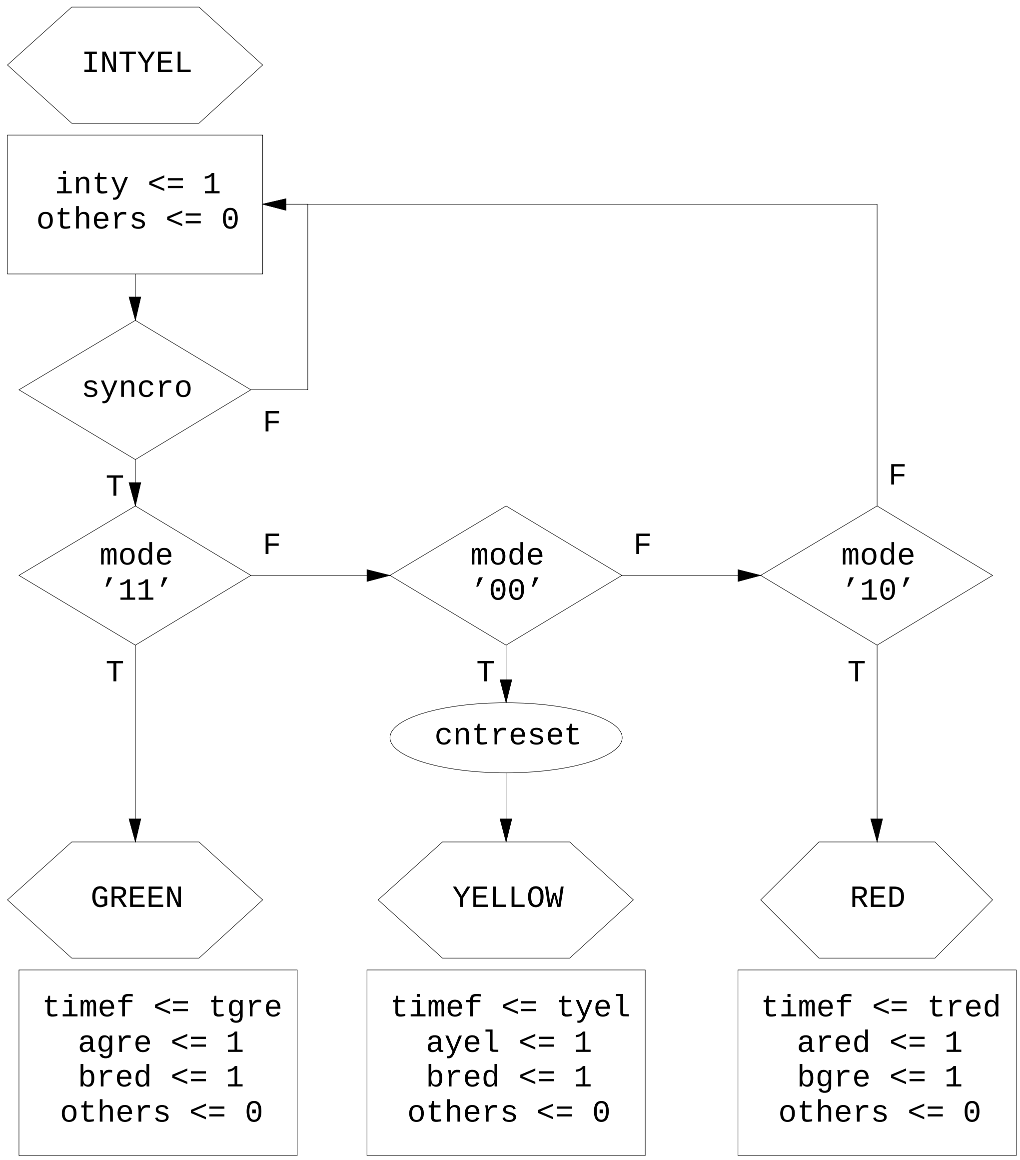


Fig. 2. Timing Relation between the Traffic signals

In the following flow diagrams we represent with circles the name of each state and within the lower rectangle the inherent constant values associated with each SFSM, with diamond the conditionals and with T the actual direction when the condition is truth, the same for F when the condition is false. The symbol:

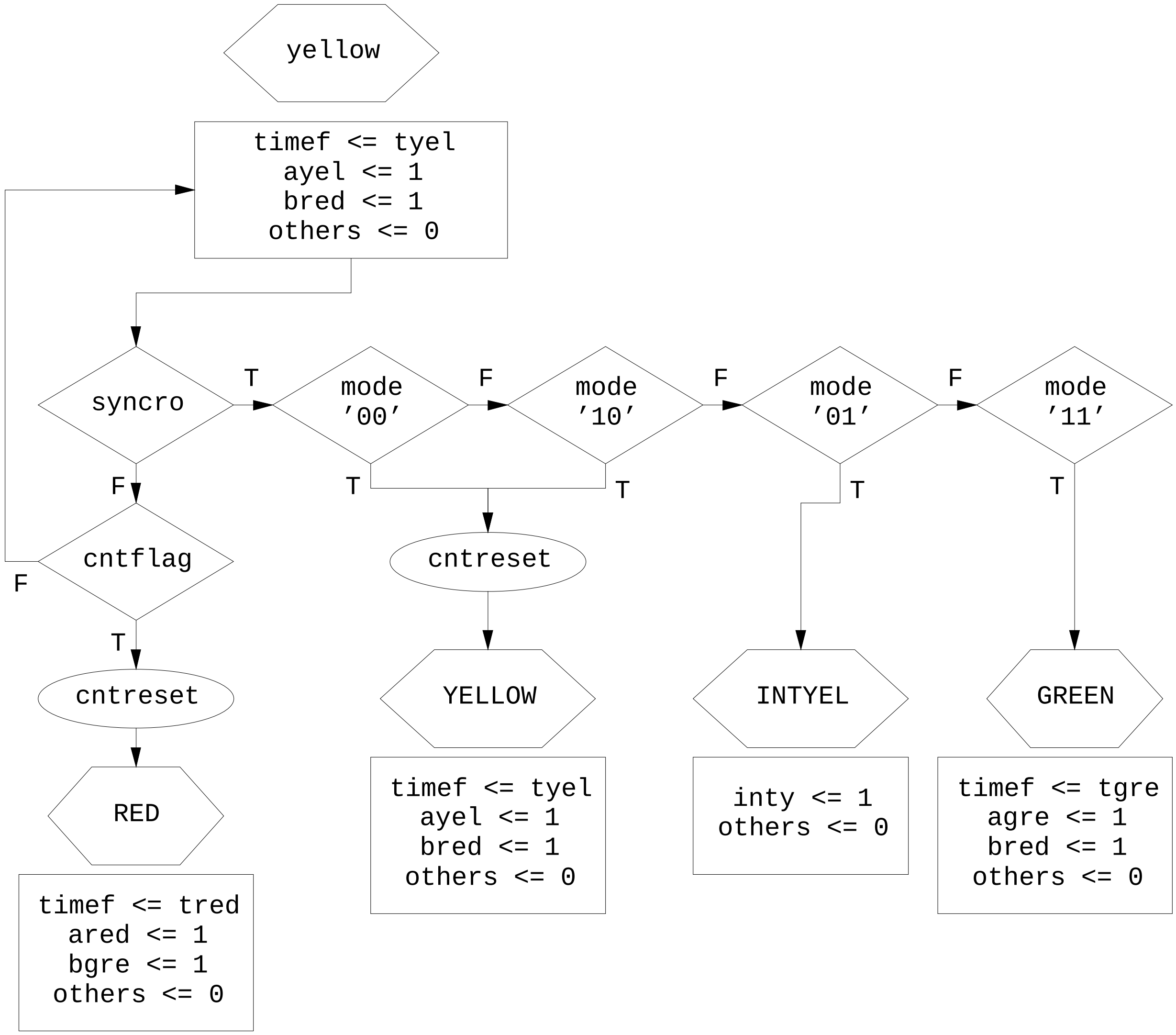
means that the internal signal **cntreset** takes instantaneously the value '1' when the condition is reached, resetting the synchronous counter when the state changes. The following flow diagrams show the transition conditions from each state:

State INTYEL:



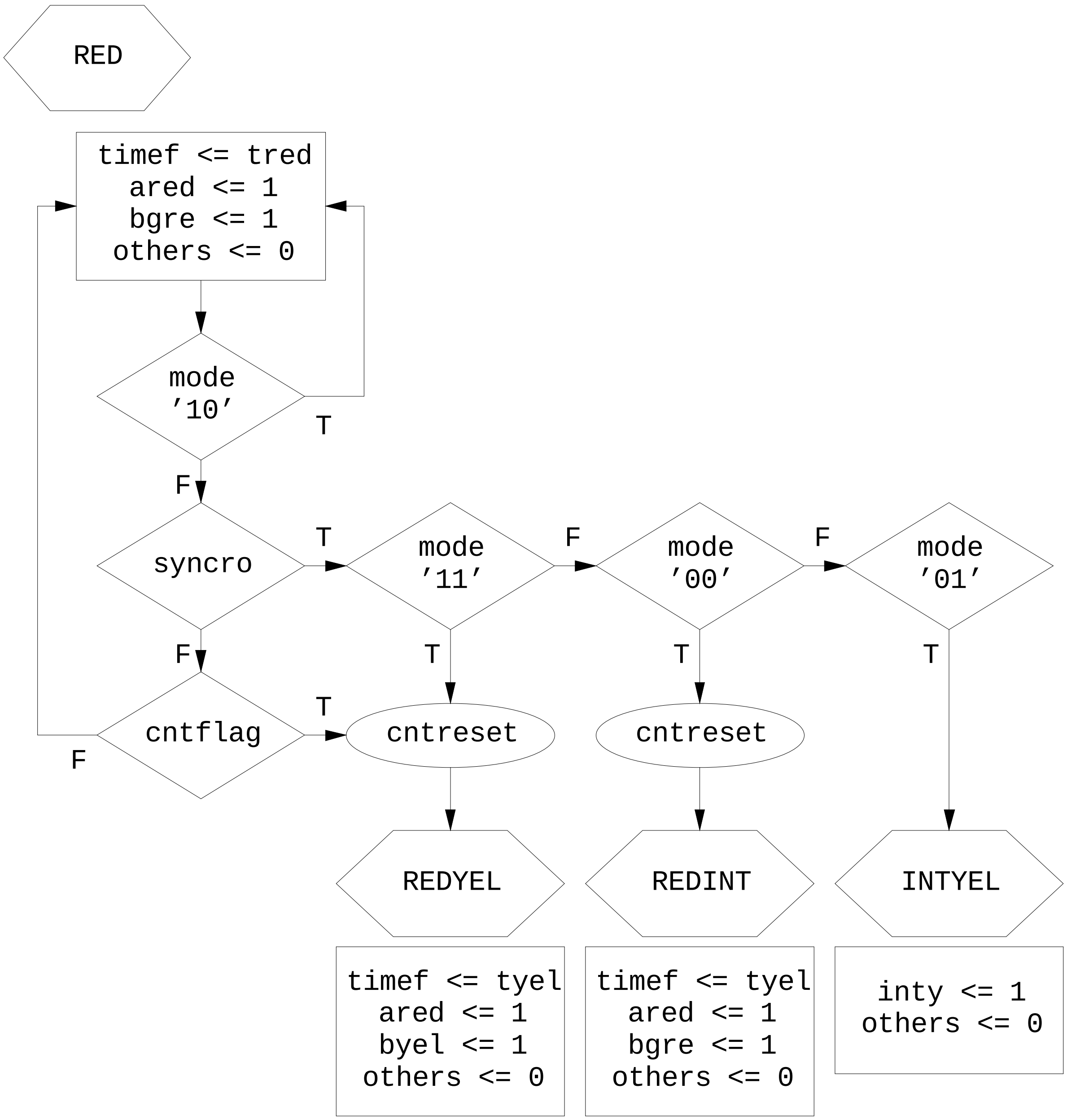
The state INTYEL corresponds to the intermittent yellow for both directions and will remain in such state until the signal **syncro** arrives together with the new working mode. If the new mode is permanent green or permanent red then the state of the counter is irrelevant. If the new mode is the normal cyclic mode then the new state will be yellow and immediately before this change the counter must be reset to zero in order to control the exact duration of the yellow state.

State YELLOW :



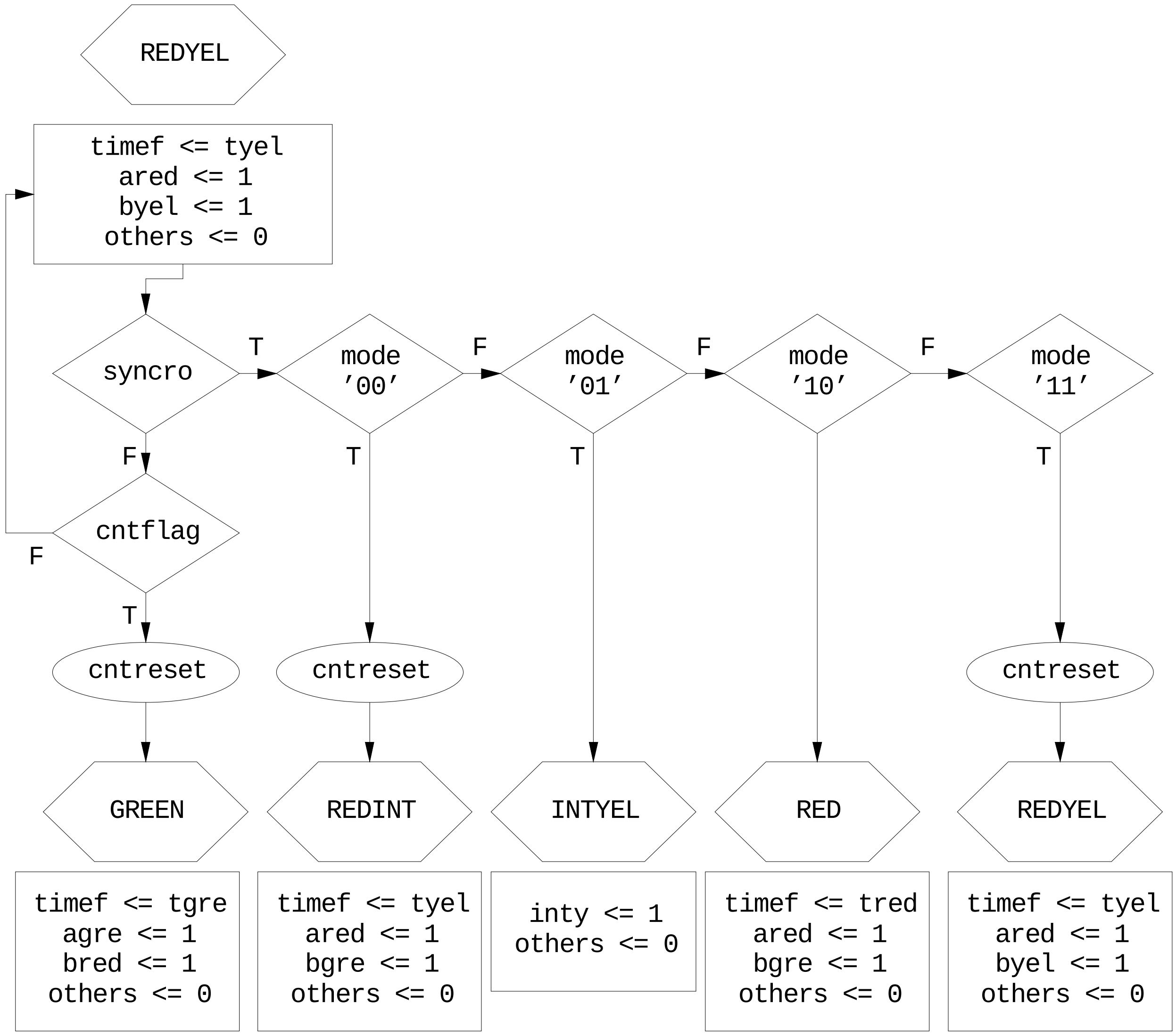
If the machine is in the yellow state and the **syncro** signal arrives, with the normal cyclic mode of operation, then the counter is reset to zero and the machine waits until the **cntflag** arrives, so as to change to the red state. If the new mode is permanent red then, as before, we put the counter to zero so as to be sure that at least the prescribed yellow duration be granted.

State RED:



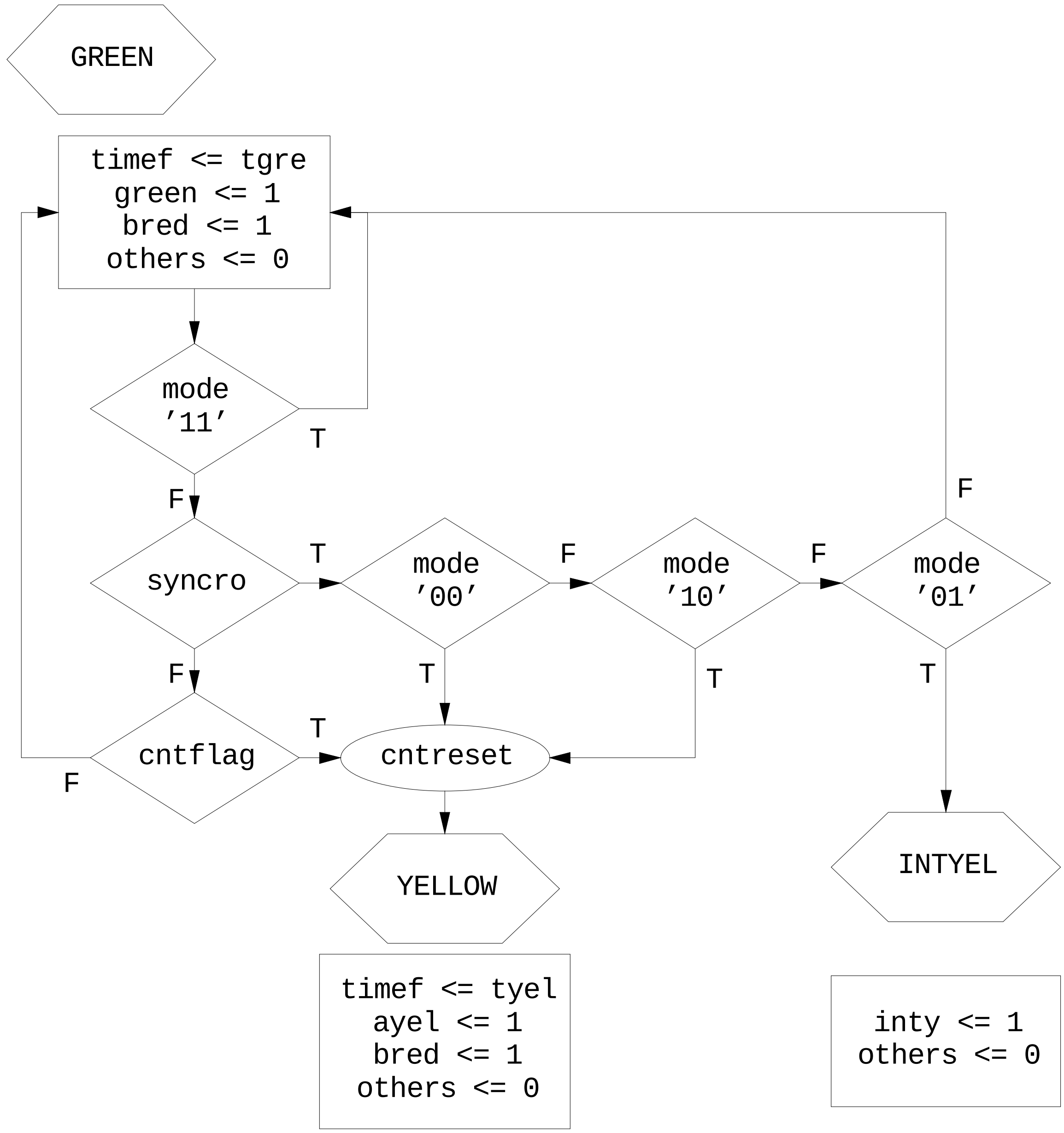
In this state, if the first conditional is true, the machine cycles permanently in this state. If the mode is different from "10" and the synchronisation signal (**syncro**) is not present, then the machine waits for the **cntflag** so as to change to the state REDYEL, while resetting the counter. If with the **syncro**, the normal cyclic mode arrives, then the next state will be REDINT, i.e. a red state but with a duration of the yellow, in order to fit the new "colour wave" without discontinuity, prolonging the actual state of the colours (see Fig. 2. a).

State REDYEL:



In this state if the **syncro** signal is not present, then the machine waits for the **cntflag,** changing to the green state after the prescribed period. If with the **syncro**, the mode is '00' (cyclic) then the next state will be REDINT (fitting the new wave) (See Fig. 2. b). If the new mode is permanent green then the state will be again REDYEL but with the counter starting from zero (to grant the duration prescribed for the state REDYEL) before the change to permanent green.

State GREEN:


GREEN
timef <= tgre
green <= 1
bred <= 1
others <= 0
mode
'11'
T
F
syncro
T
F
mode
'00'
F
mode
'10'
F
mode
'01'
F
T
T
T
cntflag
T
F
cntreset
YELLOW
INTYEL
timef <= tyel
ayel <= 1
bred <= 1
others <= 0
inty <= 1
others <= 0

State REDINT: