

Exercises



Alfredo BENSO
Politecnico di Torino (Italy)
UCSD, CA

Alfredo.benso@polito.it
www.testgroup.polito.it

Histogram Generator

2.4

2

Histogram generator

A ROM memory (1K x 8) stores 1024 data of 8 bits each, representing the samples of an arithmetic function (they are stored in the “2’s complement” notation).

Design a system that, when the START signal is raised, stores into a RAM memory the histogram of the samples stored in the ROM.

At the end of the computation, the RAM stores the statistical frequency of occurrence of each sample; in particular:

- cell 0 will store the number of times the value -128 appears in the ROM
- cell 1 will store the number of times the value -127 appears in the ROM
- ...
- cell 255 will store the number of times the value +127 appears in the ROM

2.4

3

reset

Wait for the START signal

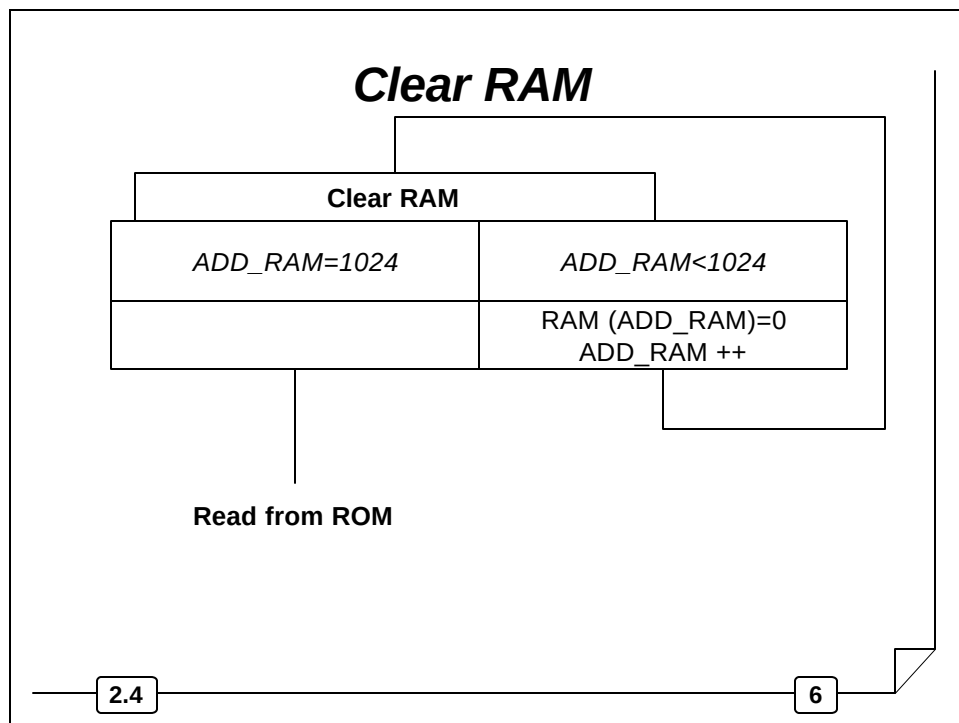
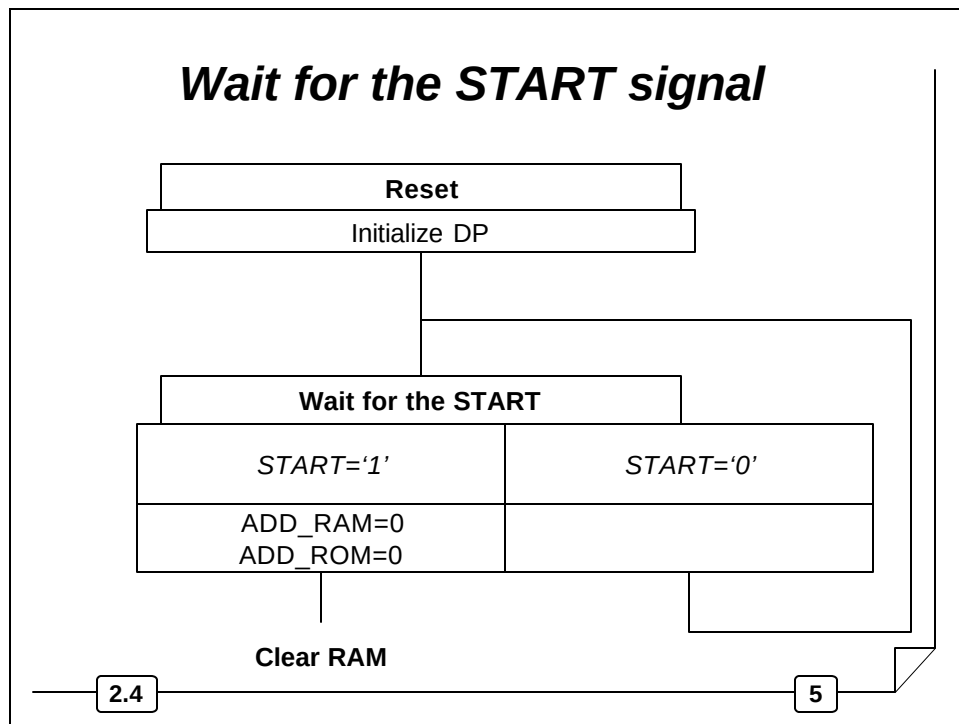
Clear RAM

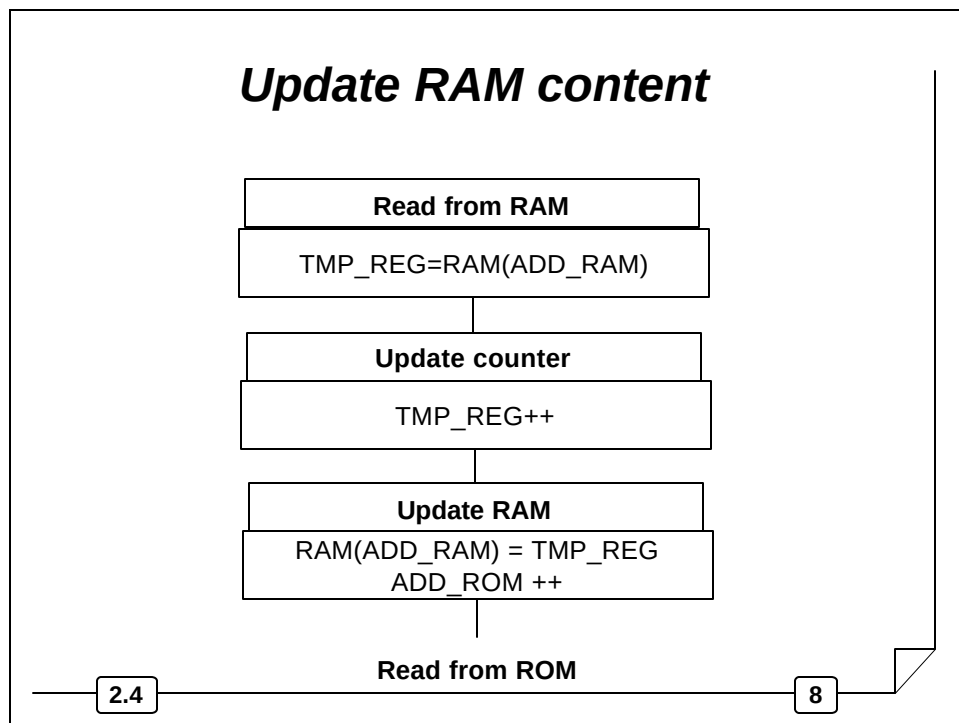
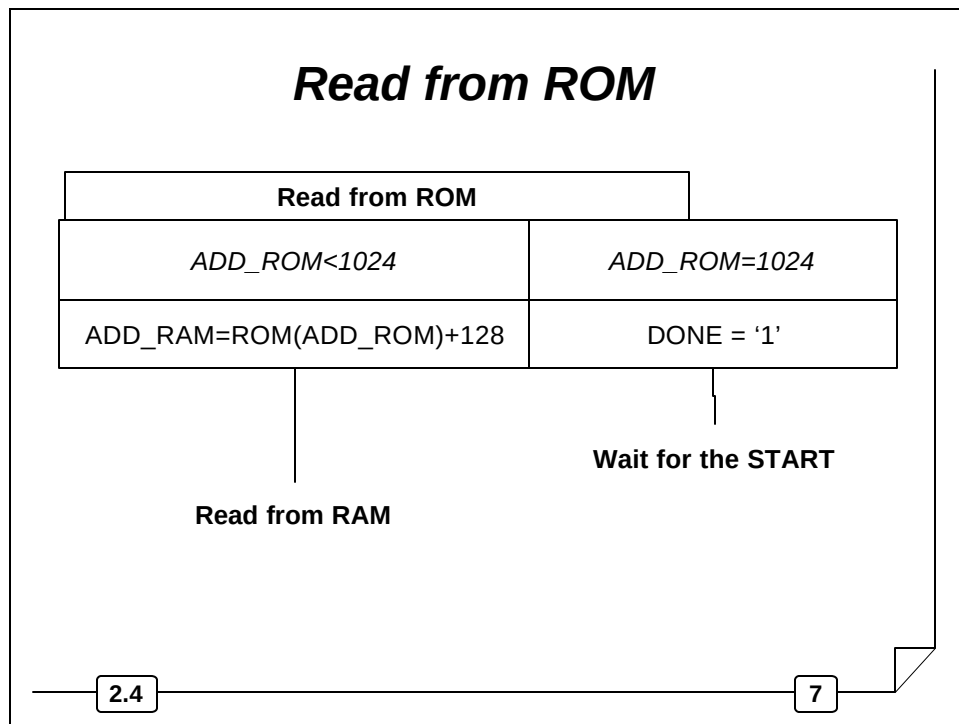
Read from ROM

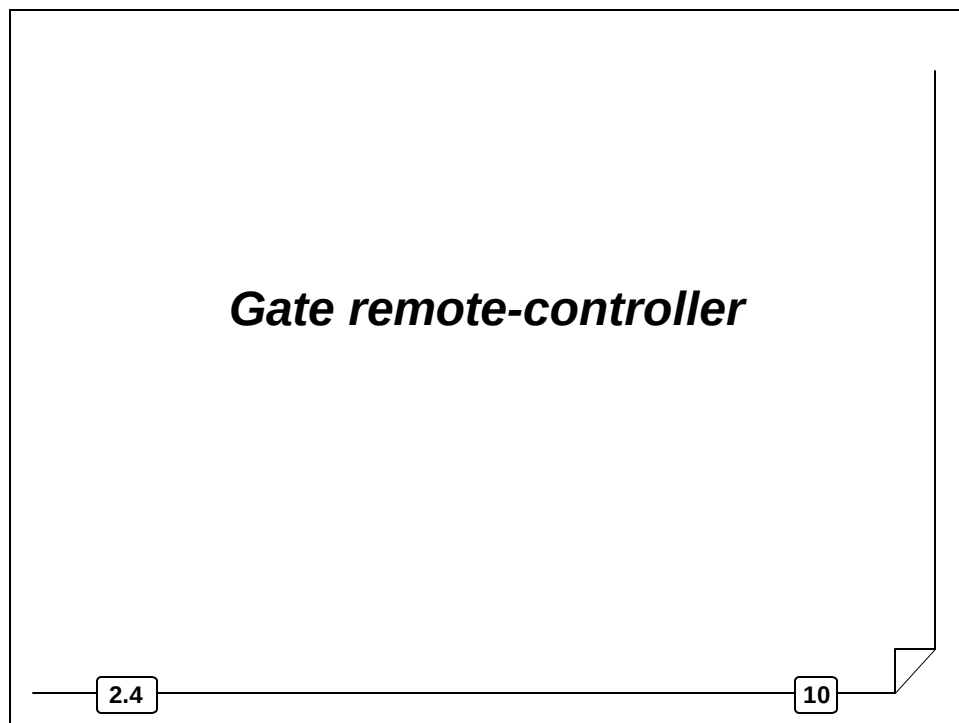
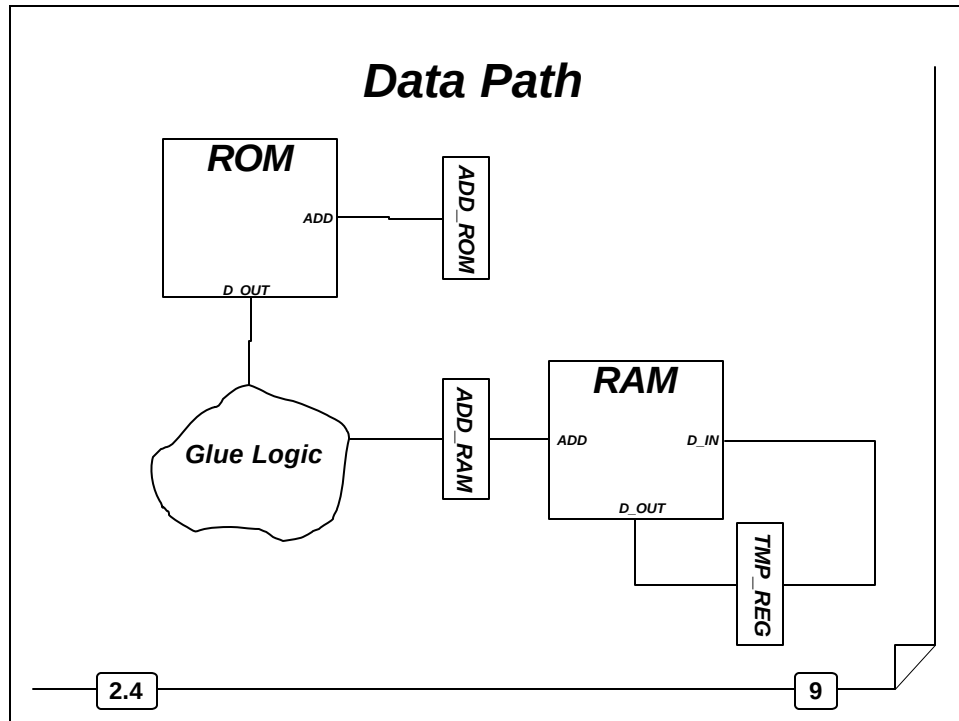
Update RAM content

2.4

4







Gate remote-controller

A gate remote controller is to be designed.

It must fulfill the following requirements:

The open and close commands arrive through a *new_com* signal, positive edge triggered. When a new command arrives, the gate has to:

- open, if it is closed
- close, if it is open
- invert the moving direction, if it is moving.
 - During normal functioning, the gate must close after a user defined amount of time (from 0 to 5 min, with a precision of half a second).
 - The gate opens inward.
 - A light must blink while the gate is moving, and it must blink for a user defined amount of seconds before the movement (from 0 sec. to 10sec., with a precision of half a second)

2.4

11

Gate remote-controller

– For security sake there are:

- Two photocells, one inside the court (to detect objects that may hamper the complete opening of the doors) and an external one on the gate jambs (to detect objects that may hamper the complete closure of the doors). If the inner photocell light is crossed during the opening or if one of the two is crossed during the closure, the gate must invert its moving direction.
- Two strength sensors able to notice whether during the movement the gate encounters an obstacle out of the photocells' range: in this case the gate must invert its moving direction.

2.4

12

Gate remote-controller

In case of reaction to an obstacle, if the gate ends in 'close' position, it stays so, while if it ends in 'open' position, it will retry to close after a user defined amount of time.

The working clock rate is 10MHz.

All signals coming from sensors are level triggered (active on high value) and must stay high at least 1ms to be recognized.

A position sensor, able to communicate if the gate is open, closed or in a middle position, is available. In case of strange situations (i.e. impossibility to close or to open at the same time) an acoustic alarm must be turned on, until it is possible to return to a normal condition.

2.4

13

Gate remote-controller

The controller has the following interface signals:

System signals

CLK – 10 MHz clock

A_RES – asynchronous reset

NEW_CMD – command signal

BUS_LIGHT – used to access the time between light turning on and gate opening

BUS_OPEN – used to access the time before closure

Sensors signals (level triggered)

IS_OPEN – high if the gate is completely open

IS_CLOSED – high if the gate is completely close

If both previous signals are high the gate is in a middle position

INT_CELL – high if the inner photocell is crossed

EXT_CELL – high if the extern photocell is crossed

STRNG_1 – high if the gate encounters an obstacle during the opening

STRNG_2 – high if the gate encounters an obstacle during the closing

Control signals (level triggered)

LIGH_ON – if it is high, the light is on

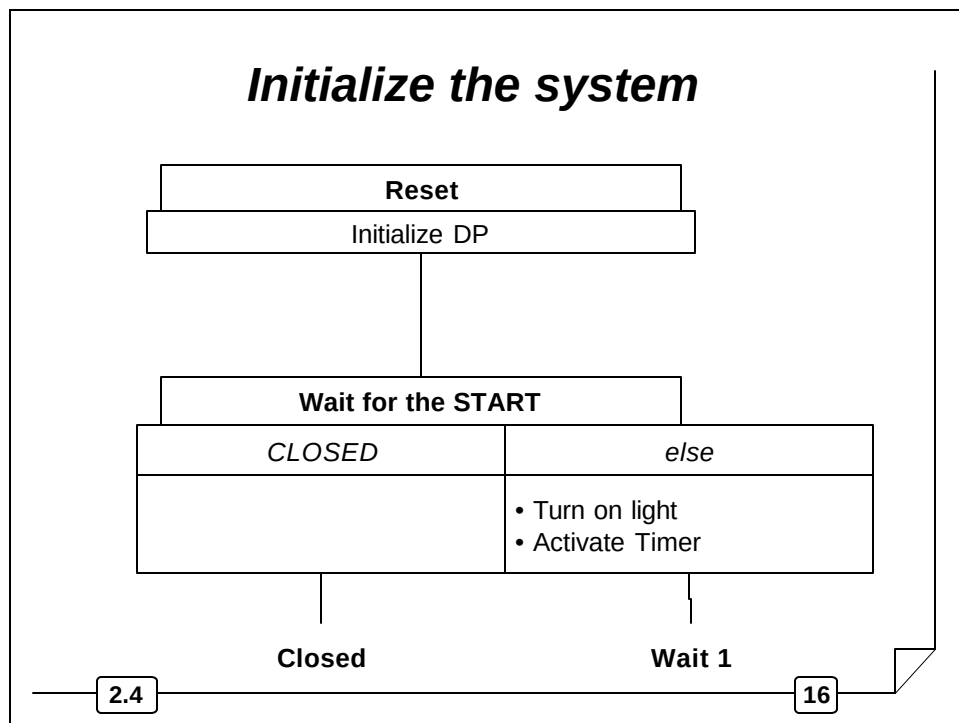
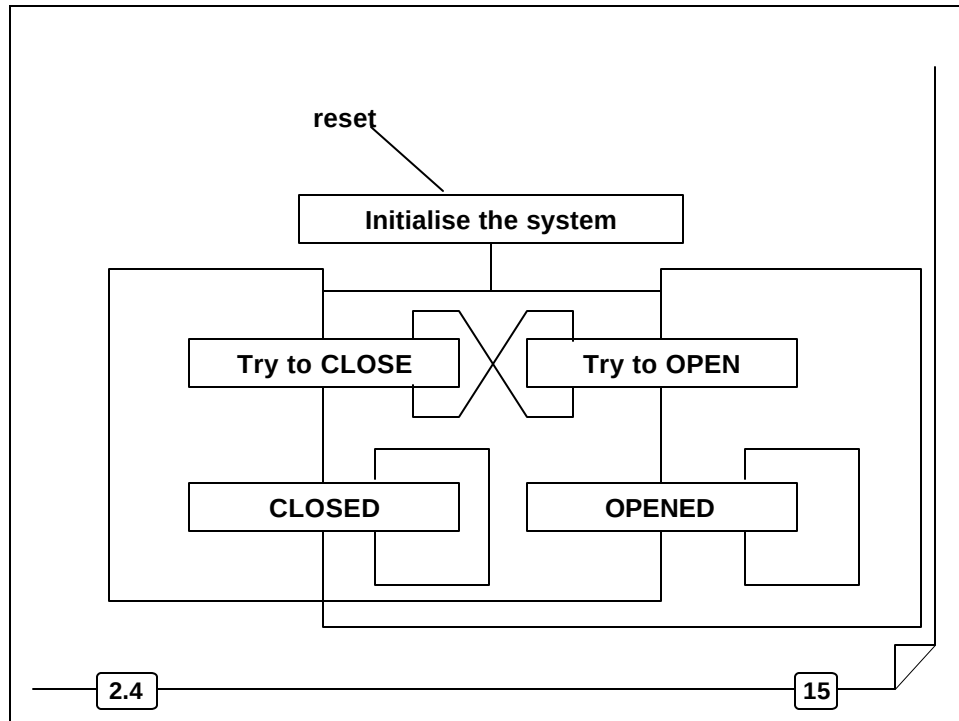
SOUND_ON – if it is high, the alarm is on

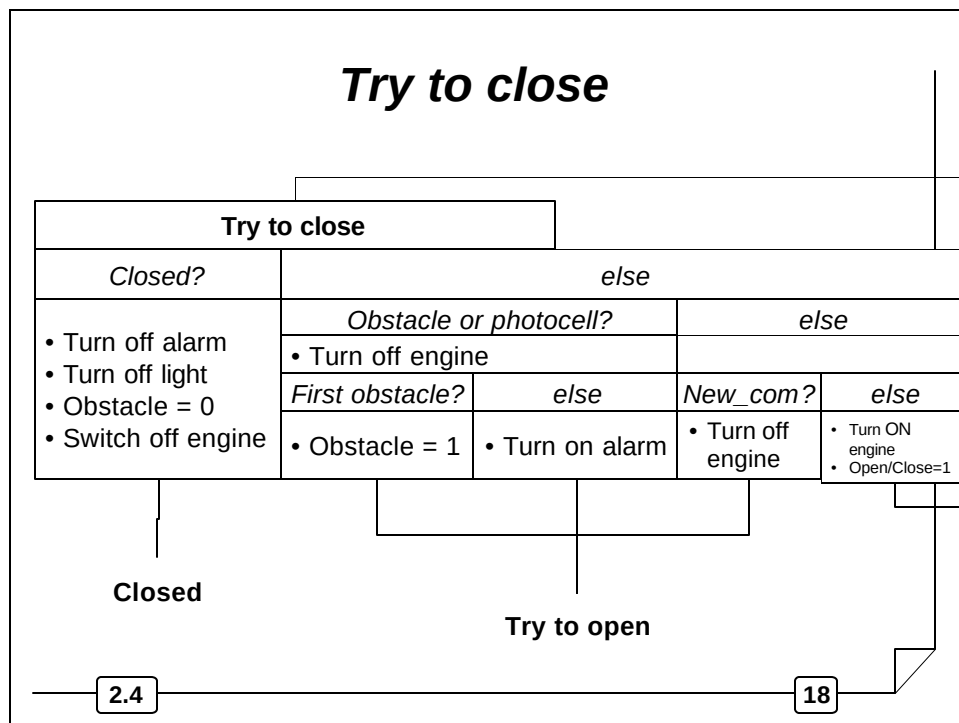
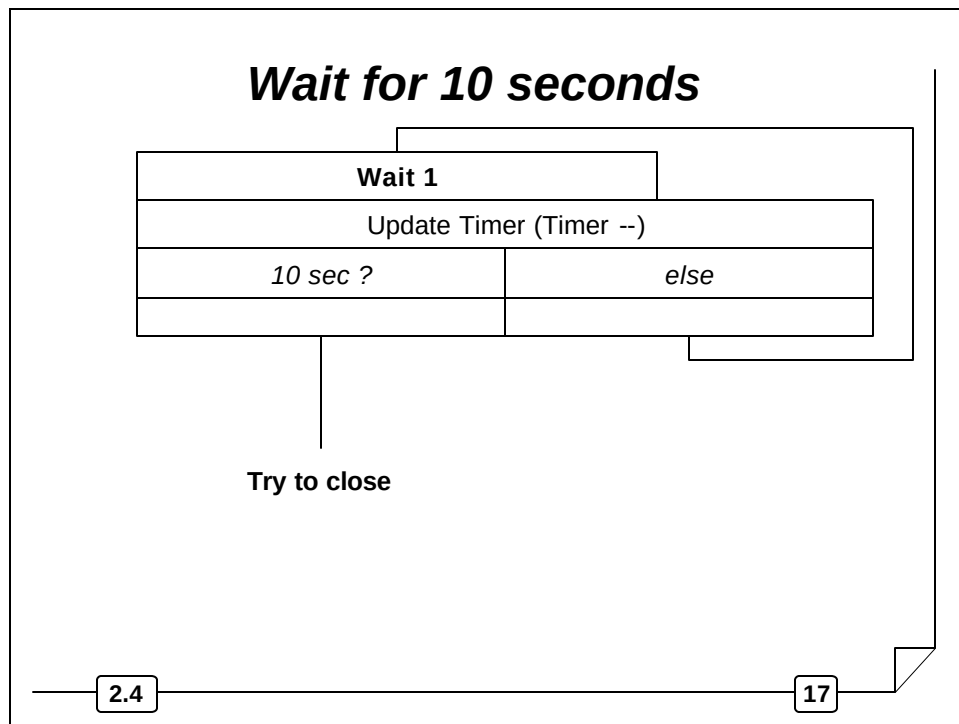
ENGINE_ON – if it is high, the engine is on

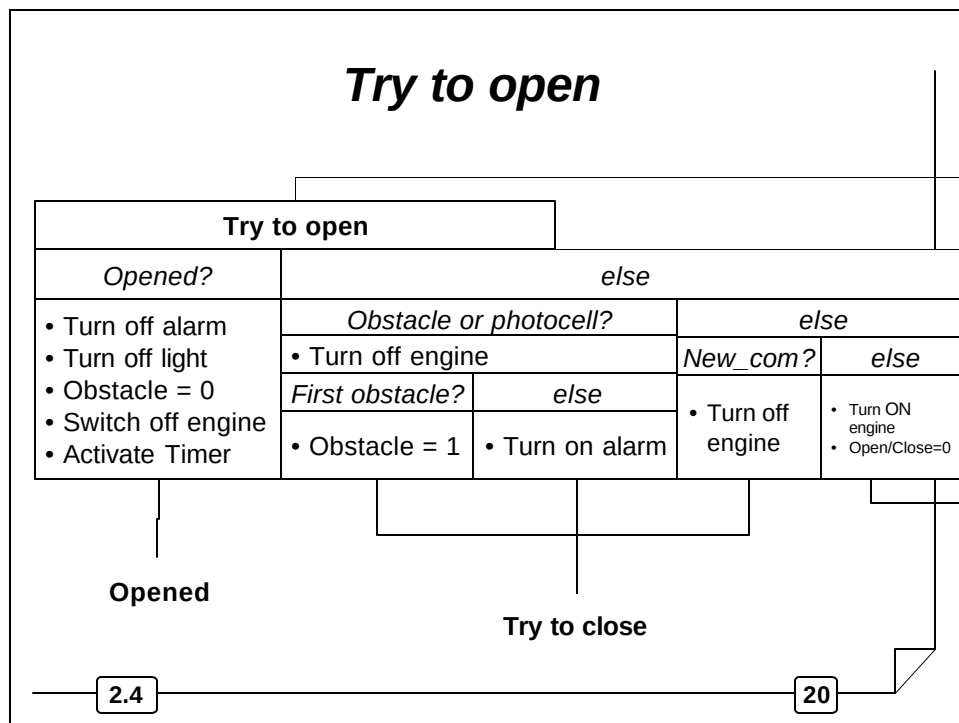
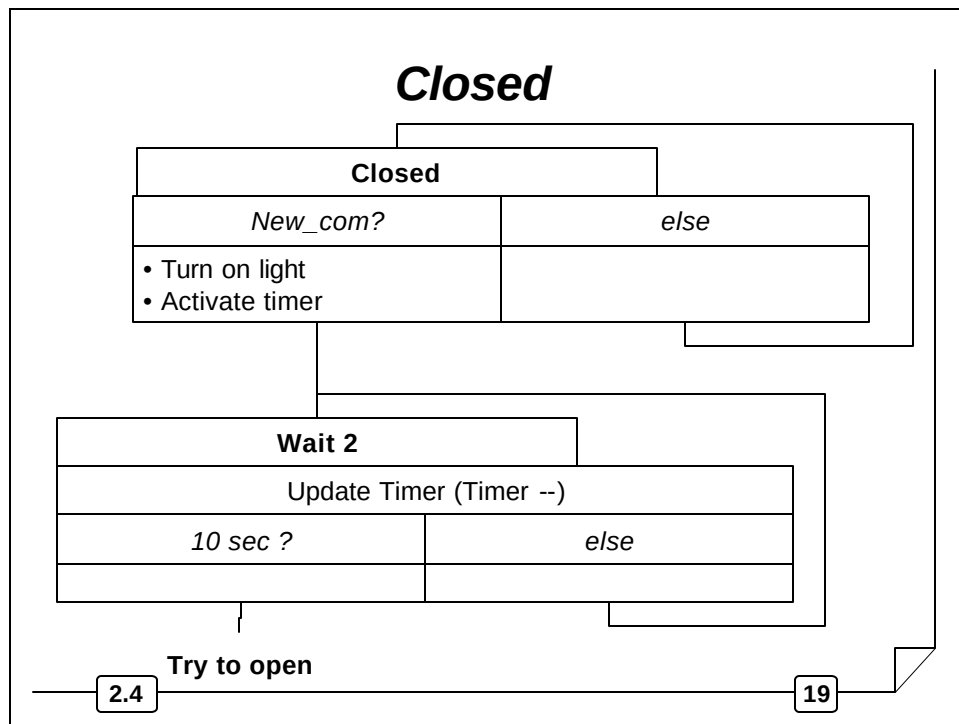
OPEN / ~CLOSE – if it is high, the engine opens the gate, otherwise it closes it

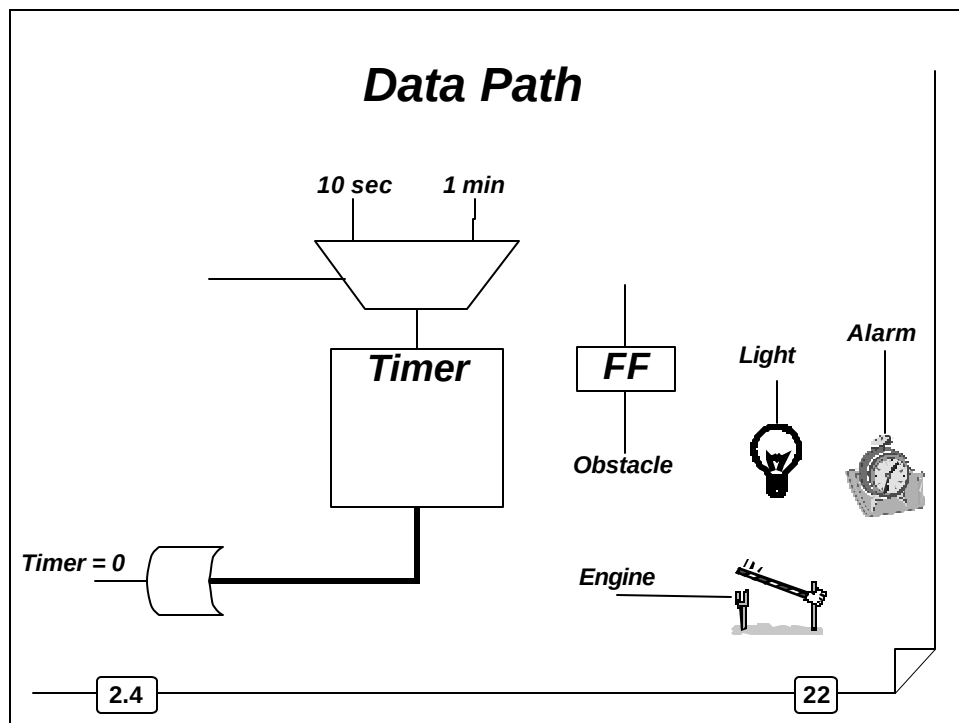
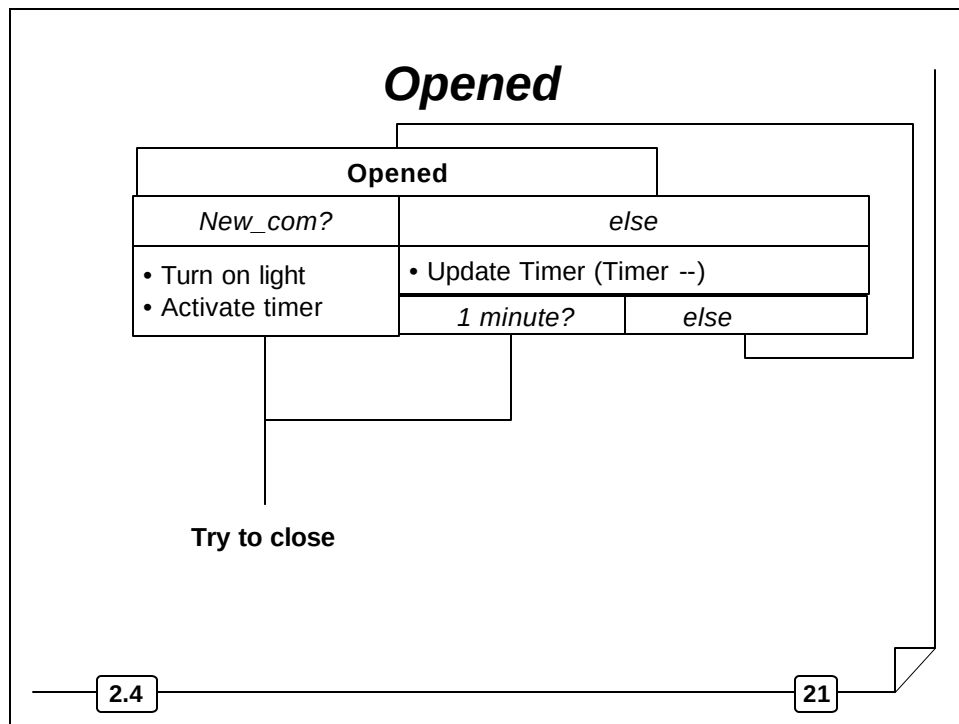
2.4

14









TV remote-controller

2.4

23

TV remote-controller

Design a TV remote controller.

The remote controller receives the following input signals:

- Change channel (P+ e P-)
- Change volume (V+ e V-)
- On/Off Switch: each pressing on the ON/OFF signal forces a change in the TV state (it switches the TV off if the TV was on, and vice versa).

The communication between remote controller and tv is implemented on a serial line; the possible commands sent by the remote to the TV are:

Command	Code
---------	------

- | | |
|---------|-----|
| • P+ | 001 |
| • P- | 000 |
| • V+ | 011 |
| • V- | 010 |
| • ON | 101 |
| • OFF | 100 |
| • ALIVE | 111 |

2.4

24

TV remote-controller

When the serial line is inactive, it sends '0's.

It is required to define the protocol (with start and stop bits) to be used over the serial line to send and receive signals.

The V+ and V- signals can be sent continuously to the TV, and they are valid until the corresponding key is pressed; all the other signals have to be sent only once, following the pressing of the corresponding key.

The remote controller has to send to the TV at least one command every second. If there are no commands to send, it has to transmit the ALIVE command.

2.4

25

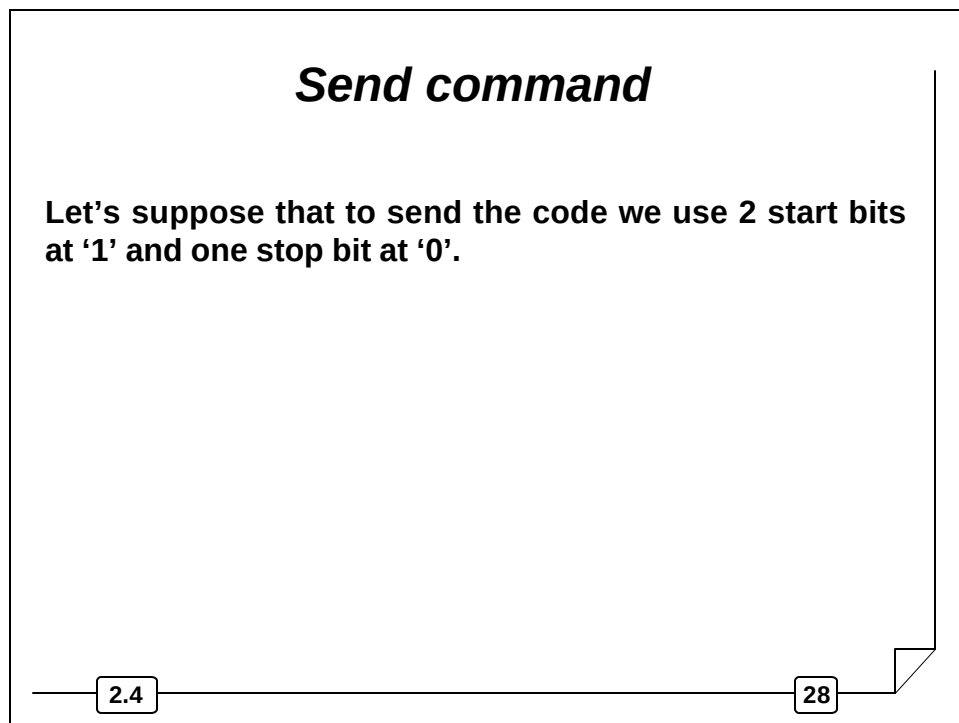
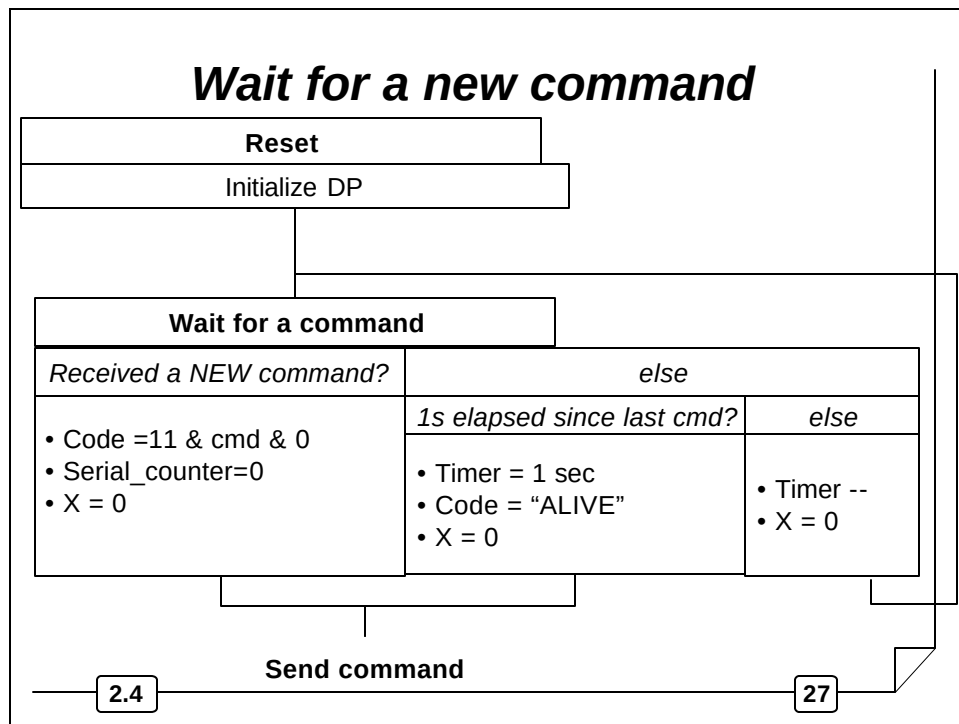
reset

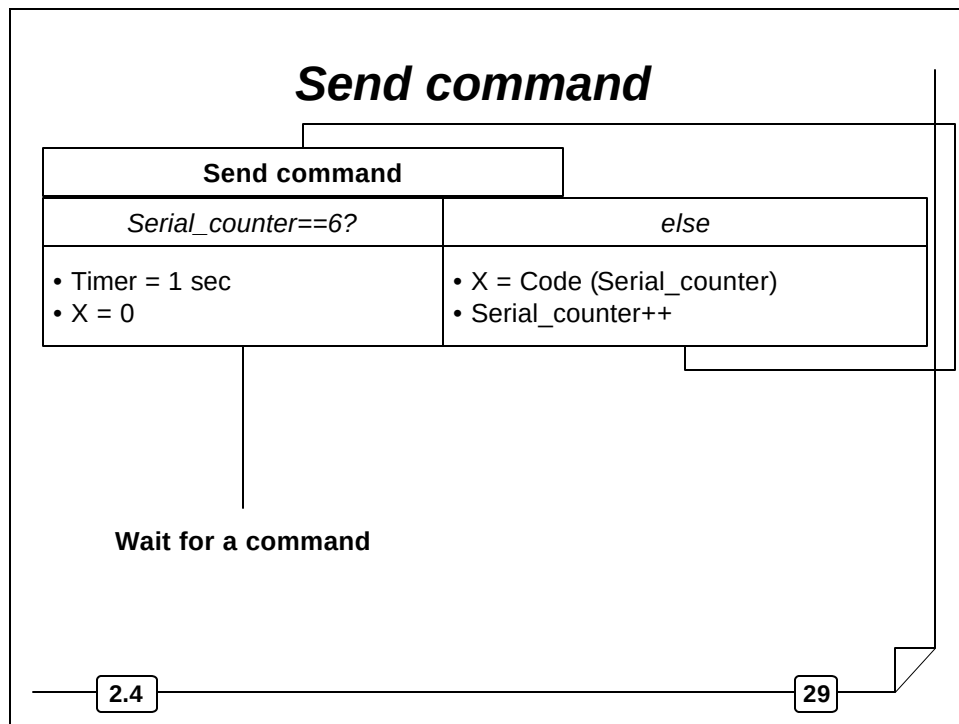
Wait for a new command

Send command

2.4

26





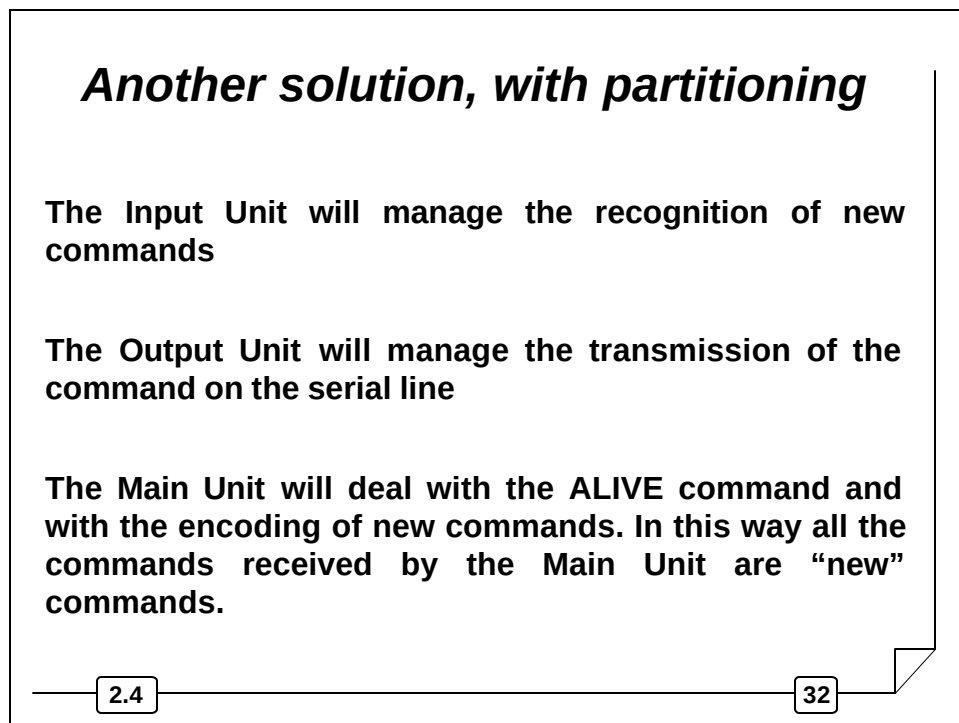
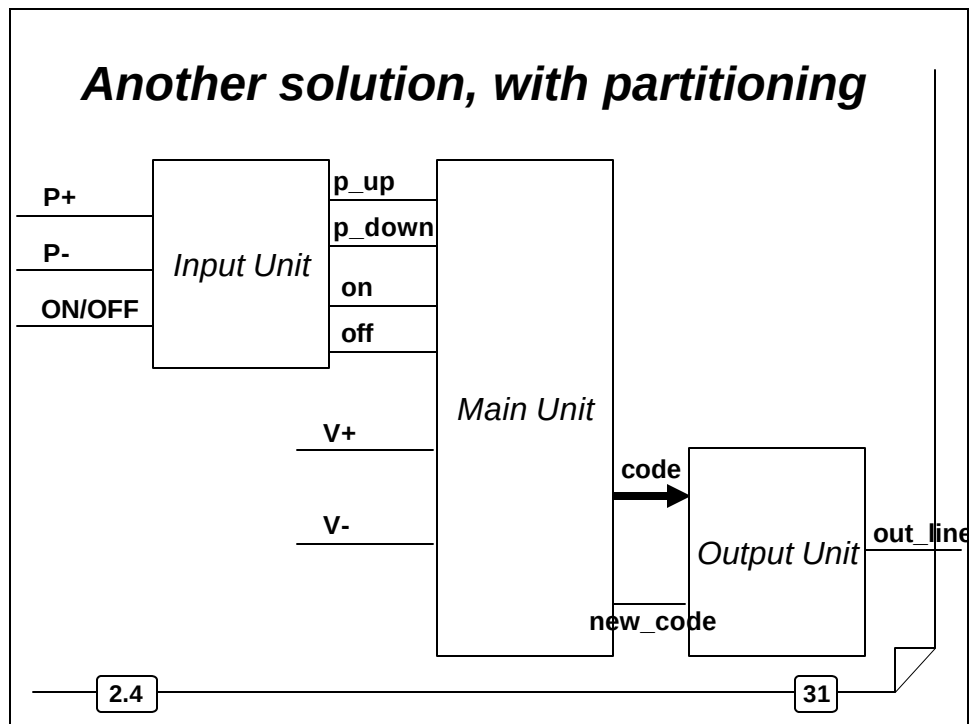
DP

In the state “Wait for a command” the condition “Received a NEW command” might be difficult to implement. A NEW command is received if V+ or V- are ‘1’, or one of the other buttons have been pressed again (not held down).

I can solve the problem either by complicating the CU or by designing an ad-hoc circuit in the DP. Both solutions will lead to complex designs.

This is a good example where partitioning might help!

2.4 30



Badge access control system

2.4

33

Badge access control system

A badge access control system is to be designed.

The system has to manage the access to 8 different gates. Each gate has:

- A magnetic badge reader, which is connected to the controller through a serial line. On the serial line, each data is transmitted using one start bit, 8 data bits, and 1 stop bit; the line is at logic '1' if nothing is transmitted.
- An automatic lock, which is open when the corresponding OPEN signal is 1. The lock is closed if OPEN=0.

When a badge is swiped in the reader, the reader sends to the control system, on the serial line, the ID (16 bits) stored in the badge. The controller decides if to grant or not the access to the gate browsing the content of a RAM memory. The memory stores the list of gates and times each ID has access to. The organization of the data into the RAM has to be defined.

2.4

34

Badge access control system

If the access is granted, the opening of the lock is enabled for 10 seconds.

Discuss the dimension of the RAM memory, depending on the max number N_USERS of possible badges.

In a second RAM it is also requested to store the last 1024 accesses granted by the system.

2.4

35

Different solutions

Before starting drawing an STG and a DP, it is better to start thinking about the general structure of the system we want to design.

We need to partition the system in order to obtain good performances and low complexity.

2.4

36

Solution #1: general idea

We can design 8 different independent systems, one for each gate.



2.4

37

Solution #1: comment

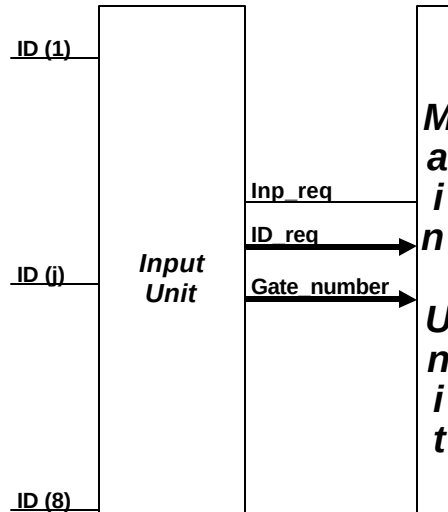
This solution reduces the complexity of the problem to the design of only one access controller, but it is probably not the best solution in terms of area and RAM memory management; changing the access policy for one ID means to act on each of the 8 RAMs.

2.4

38

Solution #2: general idea

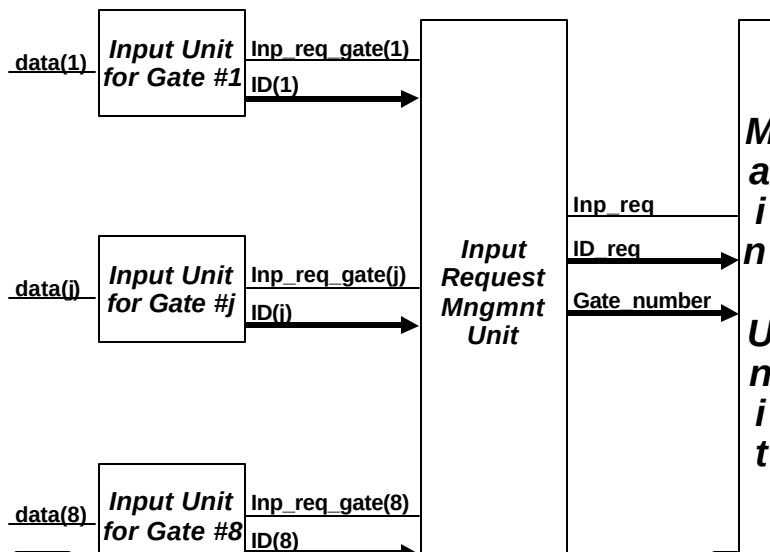
This is a good solution, but the Input Unit might be a bit complex since it has to manage (concurrently) 8 serial lines from the 8 different gates.



2.4

39

Solution #2: general idea



2.4

40

Solution #2: Input Units

This solution divides the input unit in one input unit for each gate plus an Input Request Management Unit that will not have to deal with the serial protocol, but only with the scheduling of the different requests coming from the gates.

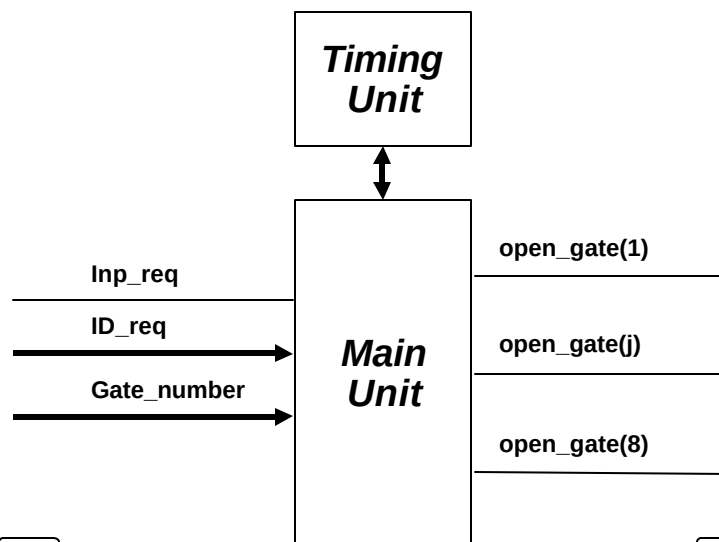
In particular the Input Request Mngmnt Unit is in charge of:

- Detecting input requests
- Selecting the one to be served, according to its service policy (round-robin, priority- based, etc)
- Providing the Main Unit the necessary info (ID, Gate #)

2.4

41

Solution #2: Main and Timing Unit



2.4

42

Solution #2: Main and Timing Unit

The Timing Unit allows the Main Unit to keep a gate open for 10 seconds and it is activated by the Main Unit when an access has to be granted.

2.4

43

Solution #2: Memory Architecture

The last decision that needs to be taken is the architecture of the RAM memory that will store the gates and times each badge has access to.

There are different solutions, each of them possibly affecting performances, area, CU and DP complexity.

In the following we will use the following notations:

- N_{max} = max number of users
- N_{usr} = actual number of users

2.4

44

Memory #1

- 8 RAMs, one for each gate
- Each RAM has the following structure:



Word: 48 bit, one for each half hour

of words: one for each Badge ID – 2^{16} words

Total size: $8 * 2^{16} * 48 = 3 * 2^{23}$

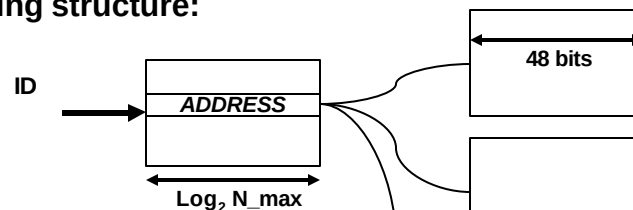
Access time: $O(1)$

2.4

45

Memory #2.1

- We can use an indirect addressing with the following structure:



Word: $\log_2 N_{\max}$ and 48

of words: $2^{16} + 8 * N_{\max}$

Total size: $2^{16} * \log_2 N_{\max} + 8 * 48 * N_{\max}$

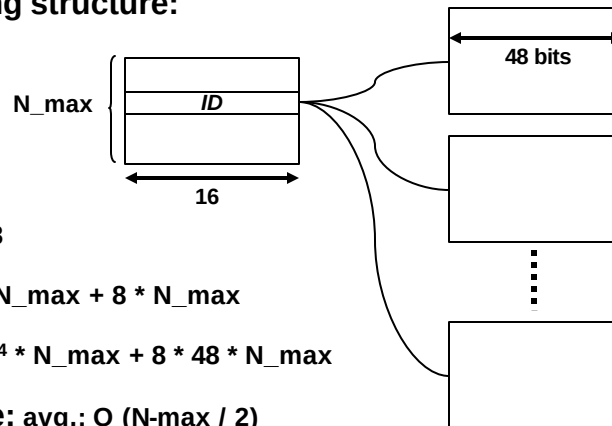
Access time: $O(2)$

2.4

46

Memory #2.2

- We can use an indirect addressing with the following structure:



Word: 16 + 48

of words: $N_{\max} + 8 * N_{\max}$

Total size: $2^4 * N_{\max} + 8 * 48 * N_{\max}$

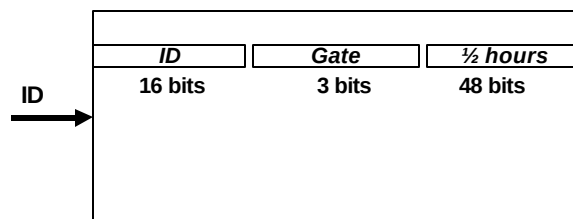
Access time: avg.: $O(N_{\max} / 2)$

2.4

47

Memory #3

Only one RAM with the following structure:



Word: 67

of words: $8 * N_{\max} * 67 = 536 N_{\max}$

Total size: $2^{16} * \log_2 N_{\max} + 8 * 48 * N_{\max}$

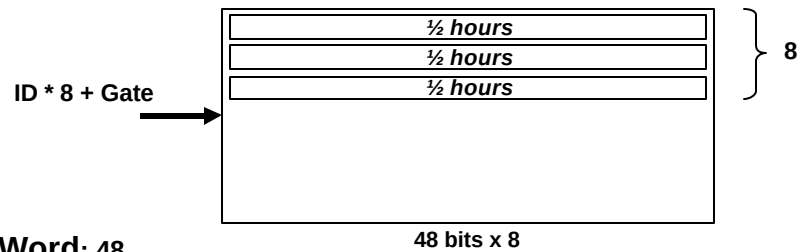
Access time: $O(2)$

2.4

48

Memory #4

Only one RAM with the following structure:



Word: 48

of words: $8 * N_{max}$

Total size: $8 * 48 * N_{max}$

Access time: $O(1)$

2.4

49

Comparison

Memory Arch.	Area	Access Time
1	$3 * 2^{23}$	$O(1)$
2.1	$2^{16} * \log_2 N_{max} + 3 * 2^7 * N_{max}$	$O(2)$
2.2	$400 * N_{max}$	$O(N_{max})$
3	$536 * N_{max}$	$O(8 * N_{max})$

2.4

50

