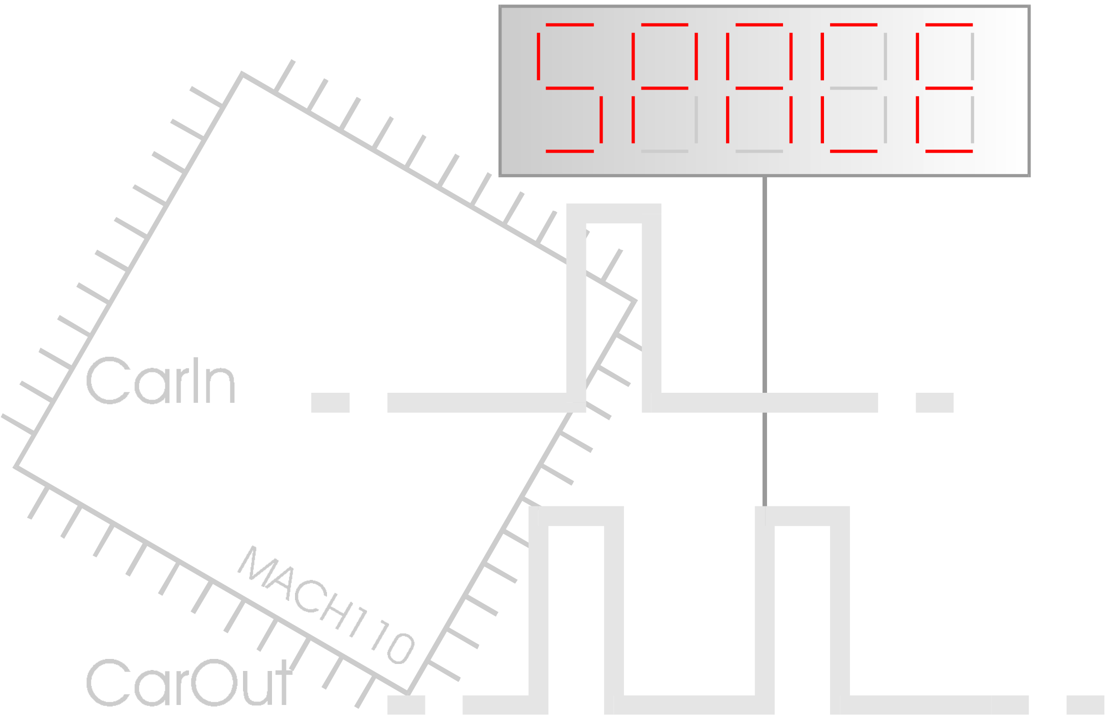
Lab-Report
Digital Electronics

Car Park Monitor

implemented in a MACH110



Name:	Dirk Becker
Course:	BEng 2
Group:	A
Student No.:	9801351
Date:	08/12/1998

UEL

UNIVERSITY of
EAST LONDON

1. Contents

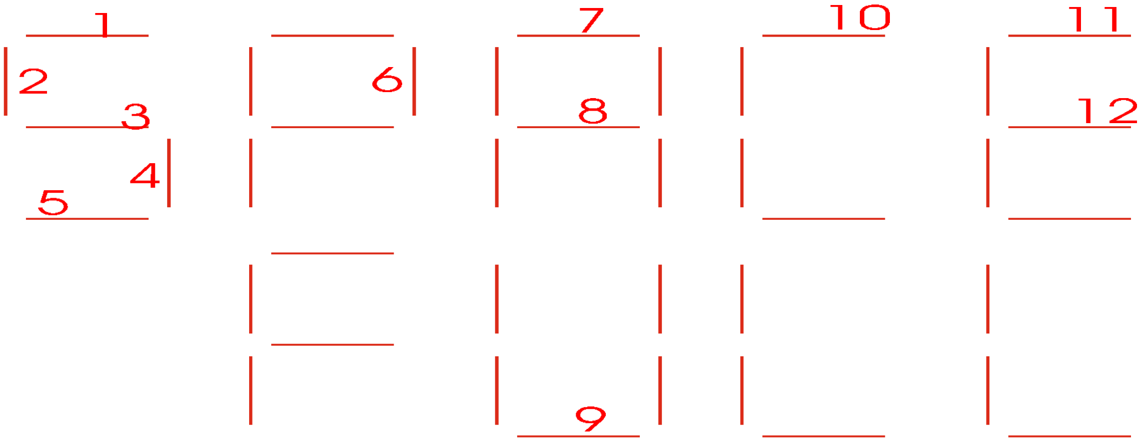
2. Introduction

A Car Park Monitor was to implement in one MACH 110 Chip. The MACH 110 contains two PAL 22V16 structures with internal facilities to connect them together.
The used Software MACHXL 2.0 is very similar to PALASM4.
The Car Park Monitor which was to design should be able to recognise incoming and outgoing cars and it should display “SPACE” or “FULL” depending on the counter status from 0 to 100.
The IN and OUT signals were generated by pulse generaters with a duration of 1ms.

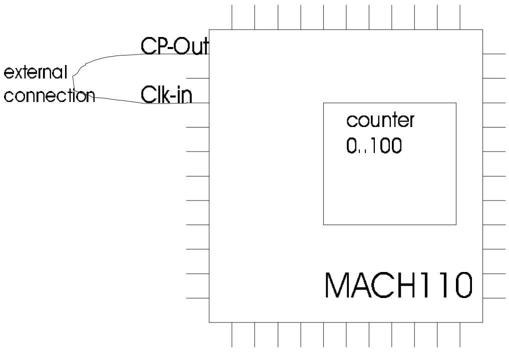
3. Implementation

a) Requirements

The Car Park Monitor must be able to recognise the INcoming and OUTgoing signals and it must count up or down from 0 to 100 depending of the input signals.
The counter to implement should count from 0 to $(100)_{10}$=$(01100100)_{2}$., which leads to a 7-Bit counter with a limited value (no reset) of $(100)_{10}$.
The following figure shows the Inputs and Outputs of the required circuit.



The outputs should be a LED-Display which should either display “SPACE”, when the counter status is between 0 and 100 or “FULL”, when the counter reaches 100.
The output LED-Display is shown on the following figure:



Segments, which are not numbered lit always, segment N^{o} 9 lits when “FULL” and all other numbered segments lit when “SPACE”.

The outputs should be all realised by MACH110 outputs (here D[0..11]).

b) Signal Conditions

At first a circuit, i.e. an equation for the converting of the incoming signals to counter compatible signals was to design.
The following truth-table shows the different conditions of the input and output to counter signals.

CarIn	CarOut	Up_Down	CP
0	**0**	**X**	**0**
0	**1**	**0**	**1**
1	**0**	**1**	**1**
1	**1**	**X**	**0**

Up_Down changes the counter between counting up or counting down and CP is the signal for a counting action. It is directly connected to the clock input of the MACH110 chip. Therefore an external connection between CP and clk is necessary.

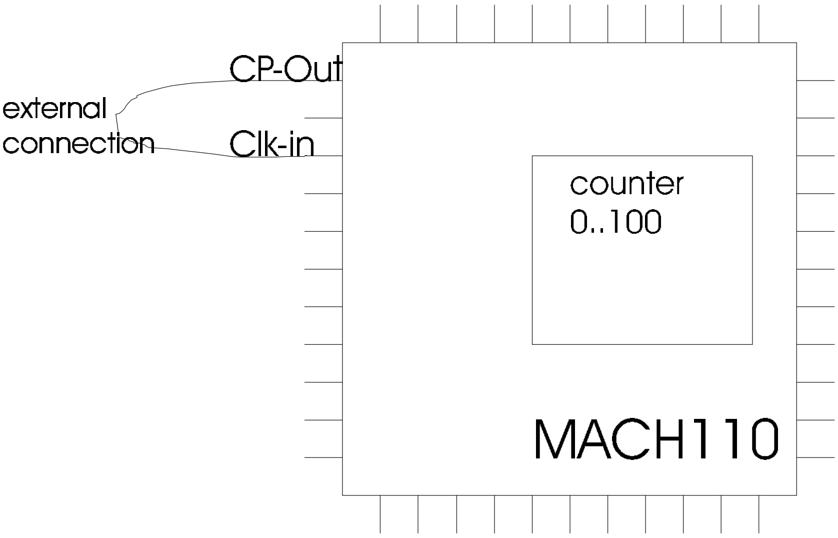


figure 1 - external connection for clk

From the truth table the following Boolean equations can be derived:

$$Up\_down = CarIn$$
$$CP = CarIn \oplus CarOut$$
$$CP = \overline{CarIn} \cdot CarOut + CarIn \cdot \overline{CarOut}$$

So the counter should increase, when a CarIn signal appears and it should decrease, when a CarOut signal appears.
The signal conditioner (the Boolean equations) change this signals into the signals Up_down and CP (i.e. clk), which are suitable to the counter.

c) Counter Design

The counter for the Car Park Monitor must have an up/down input and two limits, which are at 0 and 100. The counter has to set the output signals “FULL” and “SPACE”, which are called FULL and MIN in the MACHXL listing.
The counter itself is a usual synchronus up/down counter. The counter counts up or down when the signal CP occurs due to the external connection at the clk input. Up or down counting depends on the CAR_IN signal.
Up-counting is only performed, when no full signal is present and down-counting is only performed, when the actual counter state is greater than 0. In the listing this requirement is fulfilled by the MIN equation.
The counter outputs need’nt to occur as PINS at the MACH110, because they are only needed for internal use.
The only outputs are the outputs for the common anode LED-Display. They are directly derived from the full signal via simple Boolean inverter function.
The LED-Display connectors can be seen at a) Requirements

The complete MACHXL listing is following:

;MACHXL Design Description

```
;---------------------------------- Declaration Segment ------------
TITLE    CARCNT.PDS    ;An up/down counter - Range 0 to 100 - Car Counter
PATTERN
REVISION BETA
AUTHOR   Dirk Becker, 9801351
COMPANY  UEL
DATE     18/11/98

CHIP  _CNT  MACH110

;---------------------------------- PIN Declarations ---------------
;
PIN 3 CAR_IN
PIN 2 CAR_OUT
PIN 35 CLK
PIN 37,30,19,41,38,16,31,28,25,43 D[11..2]
PIN 40,9 D[1..0]
NODE 17,14,11,8,5,2,18 Q[6..0] REG

NODE 3 FULL
NODE 6 MIN
PIN 27 CP ; count
```

```
;----------------------------------- Boolean Equation Segment ------
EQUATIONS

FULL=  Q[6]*Q[5]*/Q[4]*/Q[3]*Q[2]*/Q[1]*/Q[0]  ; The Car Park is full
                                               ; >= 100
MIN=/Q[0]*/Q[1]*/Q[2]*/Q[3]*/Q[4]*/Q[5]*/Q[6]  ; if counter at 0 stop
CP=/CAR_IN*CAR_OUT+CAR_IN*/CAR_OUT             ; counting output

D[0]=FULL    ; The outputs for the LEDs - D[0] is LED connector 1
D[1]=FULL    ;                            D[1] is LED connector 2
D[2]=FULL    ;                            ...
D[3]=FULL
D[4]=FULL
D[5]=FULL
D[6]=FULL
D[7]=FULL
D[8]=/FULL
D[9]=FULL
D[10]=FULL
D[11]=FULL

Q[6..0].CLKF = CLK       ;States which bistables are to be clocked.
IF CAR_IN = 1 THEN      ; To count up
  BEGIN
    Q[0].T = /full ;Toggles on every count if full is low
    Q[1].T = Q[0]
    Q[2].T = Q[1]*Q[0]
    Q[3].T = Q[2]*Q[1]*Q[0]
    Q[4].T = Q[3]*Q[2]*Q[1]*Q[0]
    Q[5].T = Q[4]*Q[3]*Q[2]*Q[1]*Q[0]
    Q[6].T = Q[5]*Q[4]*Q[3]*Q[2]*Q[1]*Q[0]
  END
ELSE                 ; To count down until 0
  BEGIN
    Q[0].T = /min
    Q[1].T = /Q[0]*/min
    Q[2].T = /Q[1]*/Q[0]*/min
    Q[3].T = /Q[2]*/Q[1]*/Q[0]*/min
    Q[4].T = /Q[3]*/Q[2]*/Q[1]*/Q[0]*/min
    Q[5].T = /Q[4]*/Q[3]*/Q[2]*/Q[1]*/Q[0]*/min
    Q[6].T = /Q[5]*/Q[4]*/Q[3]*/Q[2]*/Q[1]*/Q[0]*/min
  END

    ;----------------------------------- Simulation Segment ------------
SIMULATION
TRACE_ON CP CAR_IN CAR_OUT FULL MIN CLK Q[6..0] D[0..11]
SETF /CLK CAR_IN
FOR I := 1 TO 105 DO
  BEGIN
    CLOCKF CLK
  END
SETF CAR_OUT /CAR_IN
FOR I := 1 TO 110 DO
  BEGIN
   CLOCKF CLK
  END
TRACE_OFF
;-------------------------------------------------------------------
```

•

4. Pinout

The pinout design from MACHXL is different from the one in PALASM. The pins are usually not declared directly by the user. The user decides only between input, output and internal used pins. The programme assigns the pins later by itself using a function called “fitter”.
It’s done this way because the program can be more flexible with the large number of internal connections and the internal connection to the external pins can be handled more flexible.
The automatically generated pinout shows the following printout.

```
TITLE:     CARCNT.PDS
PATTERN:
REVISION:  BETA
AUTHOR:    Dirk Becker, 9801351
COMPANY:   UEL
DATE:      18/11/98
MACRO:     _CNT

                      GND     «-------------------+
   REG  Q[1]          CAR_OUT «----------------+  ¦  +-----------------»   VCC
        FULL          CAR_IN  «-------------+  ¦  ¦  ¦  +--------------»   D[2]          NC
        NC            NC      «----------+  ¦  ¦  ¦  ¦  ¦  +-----------»   NC            NC
   REG  Q[2]          NC      «-------+  ¦  ¦  ¦  ¦  ¦  ¦  ¦  +--------»   D[8]          NC
        MIN           NC      «----+  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  +-----»   D[1]          NC
                                   +  +  +  +  +  +  +  +  +  +  +
                                +------------------------------------+
                                ¦  6  5  4  3  2  1  44 43 42 41 40  ¦
        NC            NC      +-¦ 7                               39 ¦-+   NC            NC
   REG  Q[3]          NC      +-¦ 8                               38 ¦-+   D[7]          NC
        NC            D[0]    +-¦ 9                               37 ¦-+   D[11]         NC
                      NC      +-¦ 10                              36 ¦-+   NC            NC
                      NC      +-¦ 11                              35 ¦-+   CLK
                      GND     +-¦ 12           MACH110            34 ¦-+   GND
                      NC      +-¦ 13                              33 ¦-+   NC
        NC            NC      +-¦ 14                              32 ¦-+   NC
   REG  Q[4]          NC      +-¦ 15                              31 ¦-+   D[5]           NC
        NC            D[6]    +-¦ 16                              30 ¦-+   D[10]         NC
        NC            NC      +-¦ 17                              29 ¦-+   NC            NC
                                ¦  18 19 20 21 22 23 24 25 26 27 28  ¦
                                +------------------------------------+
                                   +  +  +  +  +  +  +  +  +  +  +
   REG  Q[5]          NC      «----+  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  +-----»   D[4]          NC
        NC            D[9]    «-------+  ¦  ¦  ¦  ¦  ¦  ¦  ¦  +--------»   CP            NC
        NC            NC      «----------+  ¦  ¦  ¦  ¦  ¦  +-----------»   NC            NC
   REG  Q[6]          NC      «-------------+  ¦  ¦  ¦  +--------------»   D[3]          NC
                      VCC     «----------------+  ¦  +-----------------»   NC            Q[0]    REG
                      GND     «-------------------+
```

5. Testing

A very important part of the lab was to test the designed circuit carefully, because of its complexity errors can occur very easily.
For testing a complete run was performed, where all possible actions occurred.
In the simulation segment an 105 cycles are counted up and then 110 cycles are counted down. So all the possible states should be tested.

a) Starting counting

The counter has to start at 0 and the counter has to increase at each clk impulse, which is later external connected to the CP output.
The following table shows the expected states of the outputs D[0..11]:

D[]	***0***	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***HEX***	***State***
	0	0	0	0	0	0	0	0	1	0	0	0	008	SPACE
	1	1	1	1	1	1	1	1	0	1	1	1	FF7	FULL

A printout of the counting start is following:

The counter starts counting and the outputs take their expected states, because a low signal means, that the segment lits (active low).

b) Reaching “FULL” Status and counting down

When (100) $(64)_{16}$ cycles are reached the counter stops counting and a FULL signal appears, the outputs D[0..11] change to $(FF7)_{16}$ and the status is hold, until the counter starts through the CarOut signal counting down.
The waveforms of the inputs and outputs at these states are shown at he following two printouts.
The outputs D[0..11] change of course to 008 when counting down.

c) Reaching the zero state

The counter counts down, until the outputs of f the counter are all 0. To prevent not allowed states, the counter stops counting at 0. A continued counting is prevented by the circuit design. Also the internal MIN signal occurs (which prevents counting more down than to 0). The printout of the counting down sequence is following:

6. Conclusion

PALs can be used to implement relative complex electronic circuits in one chip. A discrete implementation of the Car Park Monitor would result in using very much discrete flipflops and devices to connect them together.
By the use of a PAL development suite like MACHXL the design can be reduced to write a short programme, and the software generates automatically the code for programming the PAL chip.
Also errors can be avoided by carefully testing the programme via the software own simulation segment.