JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Technische Informatik: Abgabe 10

Michael Mardaus   Andrey Tyukin

16. Januar 2014

Exercise 10.1 (7-Segment display PLA)

Sorry, but I switched the order to most-significant-bit-first for the inputs.
So in order to be Question-compatible you have to exchange w and z; and x and y here.

Truthtable for LED-display

i	w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
10	1	0	1	0	-	-	-	-	-	-	-
11	1	0	1	1	-	-	-	-	-	-	-
12	1	1	0	0	-	-	-	-	-	-	-
13	1	1	0	1	-	-	-	-	-	-	-
14	1	1	1	0	-	-	-	-	-	-	-
15	1	1	1	1	-	-	-	-	-	-	-

This leads to these K-maps:

a	yz			
wx	00	01	11	10
00	1	0	1	1
01	0	1	1	0
11	d	d	d	d
10	1	1	d	d

b	yz			
wx	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	d	d	d	d
10	1	1	d	d

c	yz			
wx	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	d	d	d	d
10	1	1	d	d

d	yz			
wx	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	d	d	d	d
10	1	0	d	d

e	yz			
wx	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	d	d	d	d
10	1	0	d	d

f	yz			
wx	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	d	d	d	d
10	1	1	d	d

g	yz			
wx	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	d	d	d	d
10	1	1	d	d

$a = w + xz + \bar{x}\bar{z} + yz$
$b = \bar{x} + \bar{y}\bar{z} + yz$
$c = \bar{y} + z + x$
$d = \bar{x}\bar{z} + x\bar{y}z + y\bar{z} + \bar{x}y$
$e = \bar{x}\bar{z} + y\bar{z}$
$f = w + \bar{y}\bar{z} + x\bar{y} + x\bar{z}$
$g = w + \bar{x}y + y\bar{z} + x\bar{y}$

This gives us this PLA:

input	w	xz	$\bar{x}\bar{z}$	yz	$\bar{y}\bar{z}$	$\bar{x}$	$\bar{y}$	z	x	$x\bar{y}z$	$y\bar{z}$	$\bar{x}y$	$x\bar{y}$	$x\bar{z}$	output
w	2	0	0	0	0	0	0	0	0	0	0	0	0	0	
x	0	2	3	0	0	3	0	0	2	2	0	3	2	2	
y	0	0	0	2	3	0	3	0	0	3	2	2	3	0	
z	0	2	3	2	3	0	0	2	0	2	3	0	0	3	
	1	1	1	1	0	0	0	0	0	0	0	0	0	0	a
	0	0	0	1	1	1	0	0	0	0	0	0	0	0	b
	0	0	0	0	0	0	1	1	1	0	0	0	0	0	c
	0	0	1	0	0	0	0	0	0	1	1	1	0	0	d
	0	0	1	0	0	0	0	0	0	0	1	0	0	0	e
	1	0	0	0	1	0	0	0	0	0	0	0	1	1	f
	1	0	0	0	0	0	0	0	0	0	1	1	1	0	g

We can optimize $x = x\bar{z} + xz$ so we could substitute the x column above, so we would only need 13 columns. (I am sure there is more, but for the exercise it is *optimized* now.)

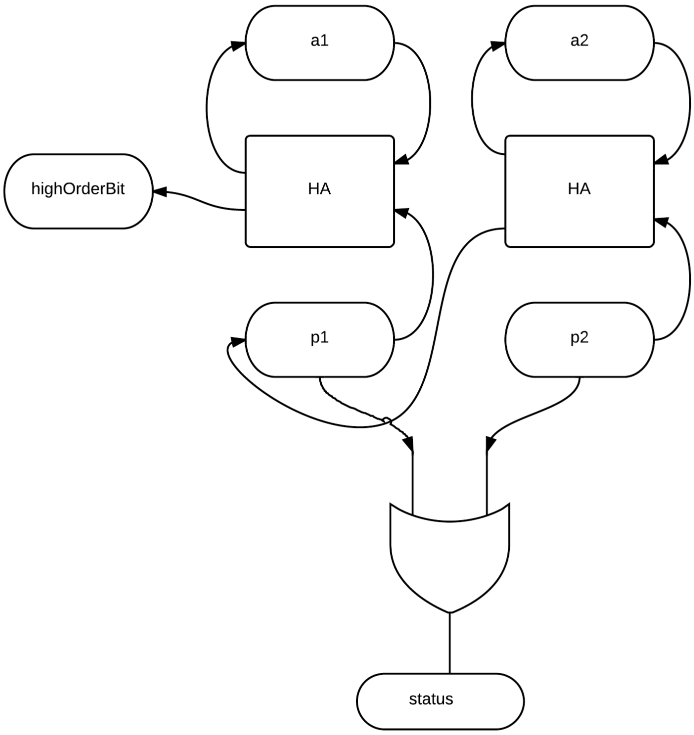
input	w	xz	$\bar{x}\bar{z}$	yz	$\bar{y}\bar{z}$	$\bar{x}$	$\bar{y}$	z	$x\bar{y}z$	$y\bar{z}$	$\bar{x}y$	$x\bar{y}$	$x\bar{z}$	output
w	2	0	0	0	0	0	0	0	0	0	0	0	0	
x	0	2	3	0	0	3	0	0	2	0	3	2	2	
y	0	0	0	2	3	0	3	0	3	2	2	3	0	
z	0	2	3	2	3	0	0	2	2	3	0	0	3	
	1	1	1	1	0	0	0	0	0	0	0	0	0	a
	0	0	0	1	1	1	0	0	0	0	0	0	0	b
	0	1	0	0	0	0	1	1	0	0	0	0	1	c
	0	0	1	0	0	0	0	0	1	1	1	0	0	d
	0	0	1	0	0	0	0	0	0	1	0	0	0	e
	1	0	0	0	1	0	0	0	0	0	0	1	1	f
	1	0	0	0	0	0	0	0	0	1	1	1	0	g

10.2 (Von-Neumann adder)

We want to implement the computation unit of a Von-Neumann adder by a PLA. The question is how many columns are required for addition of n-bit numbers.

Claim: The PLA has to have at least $3n$ columns.

Proof: We construct the Von-Neumann adder beginning on the end with the highest order bit carry, and number the delays from left to right, as in the following figure:



Numbering of the registers for the von-Neumann adder is from left to right (CLK omitted)

Recall that the function we wish to implement is defined as follows:

$$A_i = a_i\bar{p}_i + \bar{a}_i p_i$$

$$P_i = \begin{cases} a_{i+1}p_{i+1} & \text{for } i < n \\ 0 \text{ for } i = n \end{cases}$$

$$\text{highOrderBit} = a_1 p_1$$

$$\text{status} = \bigvee_{i=1}^{n-1} a_i p_i$$

Thus, for each new digit we need three new minterms: $\bar{a}_i p_i, a_i\bar{p}_i, a_i p_i$. Each of these minterms requires one new column in the PLA. The resulting PLA then looks as follows (most zeroes omitted):

a_1	2	3	2										
a_2				2	3	2							
$\vdots$							$\ddots$	$\ddots$	$\ddots$				
a_n										2	3	2	
p_1	2	2	3										
p_2				2	2	3							
$\vdots$							$\ddots$	$\ddots$	$\ddots$				
p_n										2	2	3	
	0	1	1										A_1
				0	1	1							A_2
							$\ddots$	$\ddots$	$\ddots$				$\vdots$
										0	1	1	A_n
	1	0	0										highOrderBit
				1	0	0							P_1
							$\ddots$	$\ddots$	$\ddots$				$\vdots$
										1	0	0	P_{n-1}
													P_n
				1	0	0	$\cdots$	$\cdots$	$\cdots$	1	0	0	status

Thus, the PLA consists of an AND part with $2n \times n$ blocks of size 1×3 and an OR part with $2n \times n$ blocks of size 1×3. The total number of columns required is therefore $3n$.

Remark: As always, the dot-notation ($\cdots$) could be rewritten in a more formal inductive argument, but it is questionable whether it would make the explanation shorter or clearer. In this particular case, the dot-notation seemed much more readable.