

10. Vaja: Sinteza kombinacijskega vezja v izvedbi z GAL-om

Skonstruirajte logično kombinacijsko vezje v izvedbi z GAL vezjem *PALCE16V8*. Kombinacijsko vezje naj deluje tako, da bo v okviru podanega številčnega območja*, preverjalo ali je število sodo oz. liho. Vezje ima za vhodni številčni podatek šest binarnih vhodov $A_0, A_1, A_2, A_3, A_4, A_5$ in dva izhoda Y_S in Y_L za sodo oz. liho signalizacijo

Postopek sinteze

Najprej na podlagi podanega številčnega območja*, zapišite potrebne kombinacije v logično tabelo in zapišite pripadajoče vrednosti za funkciji Y_S oz. Y_L . Nato iz tabele zapišite obe logični enačbi (brez minimiziranja) in pristopite k pisanju programa v tekstovni obliki.

Program napišite na podlagi vzorčnega programa, ki mu primerno spremenite ime datoteke, vpišete ustrezen tip GAL-a, na novo definirate oznake vhodov in izhodov, definirate številke uporabljenih priključkov in v program zapišete obe logični enačbi.

Nato z GAL prevajalnikom (*gal asm*), prevedite v tekstovni obliki napisani program, v obliko JEDEC formata, katero »razume« programator. Po uspešnem prevodu datoteko shranite na disketo in jo prenesite na računalnik s priključenim programatorjem.

Temeljito proučite tehnične podatke za PAL vezje *PALCE16V8* in še posebej vzorec programa *7SEG_DEC.gal* (posebej bodite pozorni na sintakso).

Po izvedenem programiranju, PAL vezje *PALCE16V8* vstavite v digitalno testno ploščo in ga povežite z napajanjem, vhodnimi stikali, ter led diodami na oba izhoda. Vezje in program preizkusite tako, da pri nastavljanju kombinacij na stikalih, poiščite tiste iz podanega številčnega obsega in pri tem opazujete odziva obeh izhodov.

Izhod Y_S bo v stanju »1« le tedaj, kadar bo vhodna vrednost števila $A_0, A_1, A_2, A_3, A_4, A_5$ v desetiškem pomenu soda in to le v obsegu podanega območja (npr. med 30 in 36, kjer so obe mejne vrednosti vključene).

Izhod Y_L pa bo v stanju »1« le ko bo vhodna vrednost števila $A_0, A_1, A_2, A_3, A_4, A_5$ v desetiškem pomenu liha. V vseh ostalih primerih morata izhoda Y_S in Y_L ostati neaktivna (logična »0«).

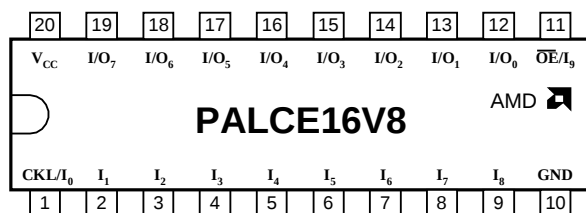
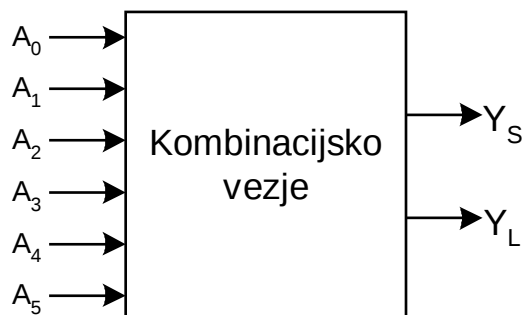
Prevajanje programa

S programom **WordPad** odprite podani vzorec programa npr.: *7SEG_DEC*, ki je na disku v mapi *C:\PALCE* in ga preuredite tako, da bo ustrezal željenim zahtevam. Shranite ga pod imenom *Kom_XY* (pri čemer črka *X* predstavlja skupino, *Y* pa številko računalnika). V isti mapi je tudi datoteka s tehničnimi podatki za integrirano vezje *PALCE16V8* pod imenom *PALCE16V8.pdf*, katerega je možno odpreti z programom **Adobe Acrobat**®.

Prepišite si program v tekstovni obliki in si oglejte prevedeno obliko programa (JEDEC).

XP windows start > zaženi > cmd > cd.. > cd.. > cd palce

Blokovna shema:



Slika 1: Razpored priključkov PALCE16V8

Pisanje in popravljanje programa ter delo s prevajalnikom:

Najprej zaženite program **WordPad** ali **BELEŽNICA**, katerega najdete pod zavihkom **PRIPOMOČKI** in odprite datoteko **7SEG_DEC**, ki je v mapi **C:\PALCE**.

Vzorčni program spremenite tako, da ustrezno spremenite naslov in v nadaljevanju pod oznako ***PINS**, definirate ustrezno število vhodov in oba izhoda.

Pod naslovom ***BOOLEAN-EQUATIONS** vpišite obe logični enačbi za izhoda Y_S in Y_L . Spremenite lahko tudi vsebino vrstic s komentarji, ki jih lahko še po želji dodate.

OPOMBA: Za prevajalnik je nujno, da ima ta programska datoteka končnico **.gal** ali **.asm** zato ji zapišite naslov na primer kot **Kom_XY.gal** (črka *X* naj predstavlja skupino, *Y* pa številko delovne mize).

Preverite pravilnost zapisa log. enačb, vejic in podpičij.

Nato program prevedite z GAL assemblerjem v **jedec** format, in sicer na sledeč način:

- poženite **MS DOS** in z ukazom **cd ..** izberite osnovni disk **C:**
- z ukazom **cd GAL** izberite mapo **GAL**.
- Startajte prevajalnik tako, da vtipkate ime prevajalnika (**gal_asm**) in ime pravkar napisanega programa po sledečem vzorcu: **gal_asm ime.txt** in pritisnite enter.

V primeru, da v programu ni napak, bo prevajanje uspešno kar potrjuje naslednje sporočilo:

Reading ime.txt ...

... writing ime.jed

Assembly successfully terminated.

V primeru napake v sintaksi, bo prevajalnik sporočil mesto napak-e v obliki:

Ime.gal XX: error: sledi opis vrste napake

(*XX* je številka vrstice v programu).

Programiranje GAL-a:

Programator **MegaMax** startate z klikom na ikono **Bližnjica do Mm** in ko se odpre osnovni meni postopajte sledeče:

- V menijski vrstici izberite ukaz Buffer/Load in naložite vsebino JEDEC programa v začasni pomnilnik,
- z izbiro Select/PLD določite ustrezn tip programirljivega vezja oz. PLD (Programmable Logic Device),
- z izbiro Device/Function/Erase zbršite »staro« vsebino GAL-a ali
- z izbiro Device/Function/Blank check preverite ali so spominske lokacije prazne,
- z izbiro Device/Function/Program prenesite vsebino programa iz začasnega pomnilnika v GAL vezje,
- z izbiro Device/Function/Verify preverite enakost programske vsebine v začasnem pomnilniku in v GAL-u (v primeru napak na spominskih lokacijah lahko pride tudi do razlike, kar pomeni napako v funkciji delovanja).

Bin 7-Segmentni-Decoder

***IDENTIFICATION**
bin_to_7;

***TYPE**
GAL16V8;

***PINS**

% dvojiški vhodi%

A = 2,
B = 3,
C = 4,
D = 5,

% Izhodi Sedem segmentnih LED-prikazovalnikov s skupno anodo %

/a = 18,
/b = 17,
/c = 16,
/d = 15,
/e = 14,
/f = 13,
/g = 12;

***BOOLEAN-EQUATIONS**

a = /A & /C + B & /D + B & C + /A & D
+ /B & /C & D + A & C & /D;

b = /A & /C + /C & /D + A & B & /D
+ A & /B & D + /A & /B & /D;

/c = /A & B & /C & /D + B & C & D + /A & C & D;

d = /B & D + /A & /C & /D + A & B & /C
+ A & /B & C + /A & B & C;

e = /A & /C + B & D + C & D + /A & B;

f = /A & /B + B & D + /C & D + /A & C + /B & C & /D;

g = /A & B + B & /C + A & D + /C & D + /B & C & /D;

***END**

Logična tabela (podano številčno območje je: _____):

Vhodna kombinacija	Število A						Izhod	
	F	E	D	C	B	A	Y _S (sodi)	Y _L (lihi)
0.	0	0	0	0	0	0		
1.	0	0	0	0	0	1		
2.	0	0	0	0	1	0		
3.	0	0	0	0	1	1		
4.	0	0	0	1	0	0		
5.	0	0	0	1	0	1		
6.	0	0	0	1	1	0		
7.	0	0	0	1	1	1		
8.	0	0	1	0	0	0		
9.	0	0	1	0	0	1		
10.	0	0	1	0	1	0		
11.	0	0	1	0	1	1		
12.	0	0	1	1	0	0		
13.	0	0	1	1	0	1		
14.	0	0	1	1	1	0		
15.	0	0	1	1	1	1		
16.	0	1	0	0	0	0		
17.	0	1	0	0	0	1		
18.	0	1	0	0	1	0		
19.	0	1	0	0	1	1		
20.	0	1	0	1	0	0		
21.	0	1	0	1	0	1		
22.	0	1	0	1	1	0		
23.	0	1	0	1	1	1		
24.	0	1	1	0	0	0		
25.	0	1	1	0	0	1		
26.	0	1	1	0	1	0		
27.	0	1	1	0	1	1		
28.	0	1	1	1	0	0		
29.	0	1	1	1	0	1		
30.	0	1	1	1	1	0		
31.	0	1	1	1	1	1		
32.	1	0	0	0	0	0		
33.	1	0	0	0	0	1		
34.	1	0	0	0	1	0		
35.	1	0	0	0	1	1		
36.	1	0	0	1	0	0		
37.	1	0	0	1	0	1		
38.	1	0	0	1	1	0		
39.	1	0	0	1	1	1		
40.	1	0	1	0	0	0		
41.	1	0	1	0	0	1		
42.	1	0	1	0	1	0		
43.	1	0	1	0	1	1		
44.	1	0	1	1	0	0		
45.	1	0	1	1	0	1		
46.	1	0	1	1	1	0		
47.	1	0	1	1	1	1		
48.	1	1	0	0	0	0		
49.	1	1	0	0	0	1		

Vhodna kombinacija	Število A						Izhod	
	F	E	D	C	B	A	Y_S (sodi)	Y_L (lihi)
50.	1	1	0	0	1	0		
51.	1	1	0	0	1	1		
52.	1	1	0	1	0	0		
53.	1	1	0	1	0	1		
54.	1	1	0	1	1	0		
55.	1	1	0	1	1	1		
56.	1	1	1	0	0	0		
57.	1	1	1	0	0	1		
58.	1	1	1	0	1	0		
59.	1	1	1	0	1	1		
60.	1	1	1	1	0	0		
61.	1	1	1	1	0	1		
62.	1	1	1	1	1	0		
63.	1	1	1	1	1	1		

Logična enačba za sode in lihe izhodne vrednosti podanega območja:

$Y_S =$ _____

$Y_L =$ _____

Program v besedilni obliki (*.txt):

Prevedena programska koda (*.jed):

GENERAL DESCRIPTION

The PALCE16V8 is an advanced PAL device built with low-power, high-speed, electricallyerasable CMOS technology. It is functionally compatible with all 20-pin GAL devices. The macrocells provide a universal device architecture. The PALCE16V8 will directly replace the PAL16R8, with the exception of the PAL16C1.

The PALCE16V8Z provides zero standby power and high speed. At 30- μ A maximum standby current, the PALCE16V8Z allows battery-powered operation for an extended period. The PALCE16V8 utilizes the familiar sum-of-products (AND/OR) architecture that allows users to implement complex logic functions easily and efficiently. Multiple levels of combinatorial logic can always be reduced to sum-of-products form, taking advantage of the very wide input gates available in PAL devices. The equations are programmed into the device through floating-gate cells in the AND logic array that can be erased electrically.

The fixed OR array allows up to eight data product terms per output for logic functions. The sum of these products feeds the output macrocell. Each macrocell can be programmed as registered or combinatorial with an active-high or active-low output. The output configuration is determined by two global bits and one local bit controlling four multiplexers in each macrocell.

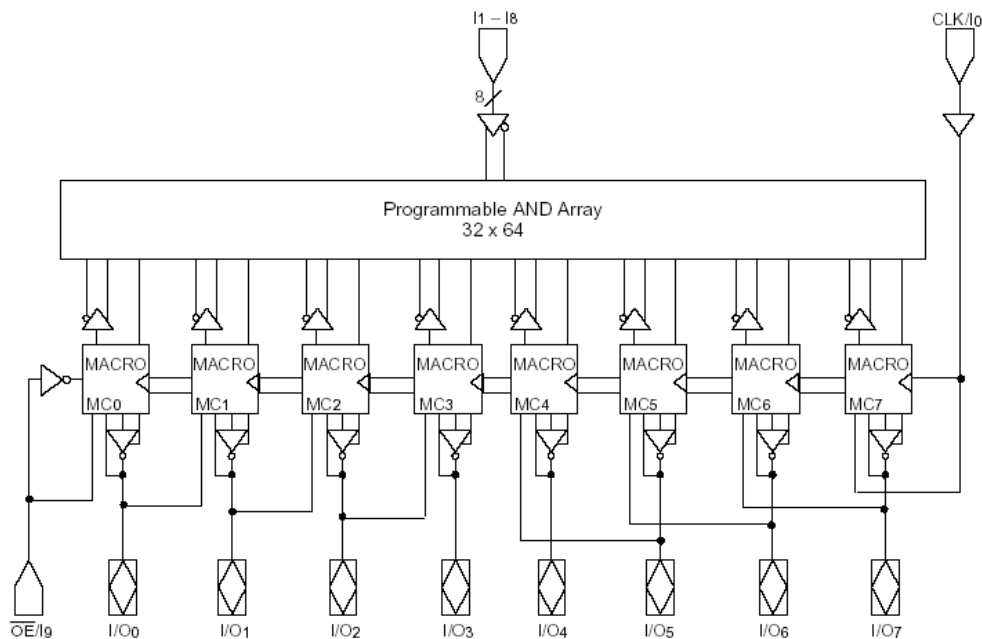


Figure 3: Block diagram

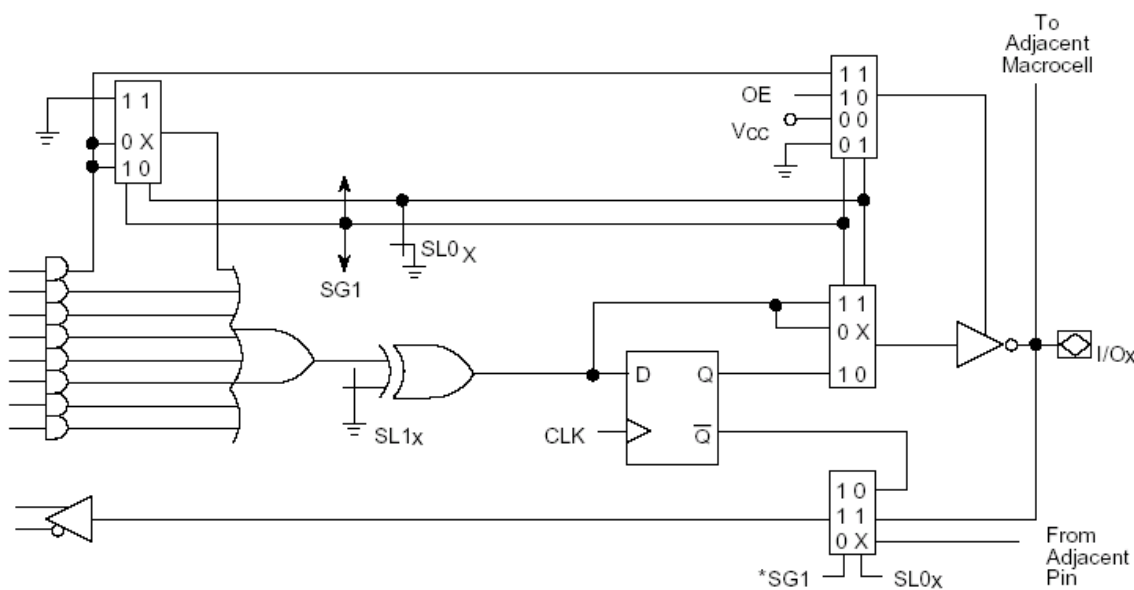


Figure 4: PALCE16V8 Macrocell

Combinatorial I/O in a Registered Device

The control bit settings are SG0 = 0, SG1 = 1 and SL0x = 1. Only seven product terms are available to the OR gate. The eighth product term is used as the output enable. The feedback signal is the corresponding I/O signal.

Dedicated Input Configuration

The control bit settings are SG0 = 1, SG1 = 0 and SL0x = 1. The output buffer is disabled. Except for MC0 and MC7 the feedback signal is an adjacent I/O. For MC0 and MC7 the feedback signals are pins 1 and 11. These configurations are summarized in Table 1 and illustrated in Figure 2.

SG0	SG1	SL0X	Cell Configuration	Devices Emulated
Device Uses Registers				
0	1	0	Registered Output	PAL16R8, 16R6, 16R4
0	1	1	Combinatorial I/O	PAL16R6, 16R4
Device Uses No Registers				
1	0	0	Combinatorial Output	PAL10H8, 12H6, 14H4, 16H2, 10L8, 12L6, 14L4, 16L2
1	0	1	Input	PAL12H6, 14H4, 16H2, 12L6, 14L4, 16L2
1	1	1	Combinatorial I/O	PAL16L8

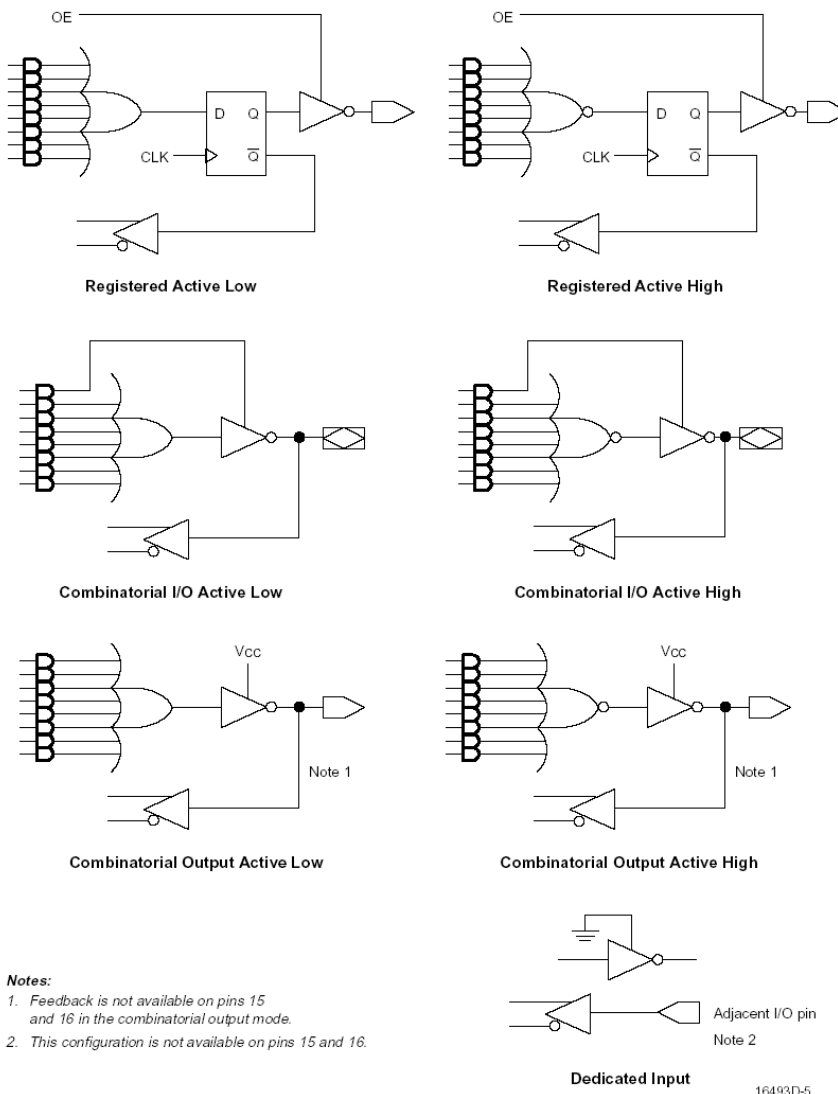


Figure 4. Macrocell configurations

