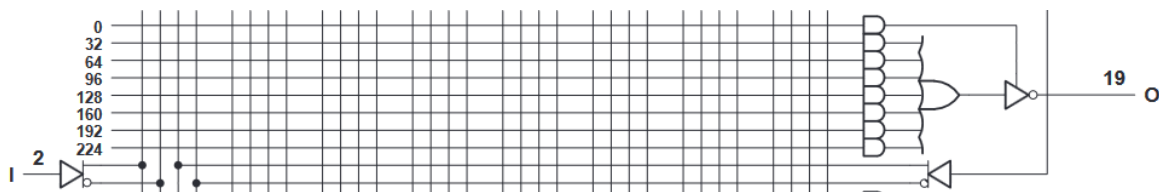


## 4.2 Programabilno vezje GAL in kombinacijska logika

Pri vaji 2.16 smo gradili kombinacijska logična vezja s pomočjo logičnih vrat. V posamezno integrirano vezje je zaradi omejenega števila nožic (14, 16 ali morda 20) mogoče spraviti le majhno število logičnih vrat, zaradi tega je za implementacijo bolj kompliciranih logičnih funkcij potrebnih veliko integriranih vezij in gnezdo medsebojnih povezav. Tako vezje je nepregledno in ga ni enostavno modificirati.

Vsako logično funkcijo lahko prevedemo tako, da jo je mogoče implementirati v standardni strukturi logičnih vrat, ki jo sestavlja najprej niz negatorjev vhodnih signalov, ki jim sledi niz AND vrat, na koncu vezja pa so OR vrata. Na sliki 1 je zgled take univerzalne strukture, pri kateri je za implementacijo poljubne logične funkcije potrebno le vzpostaviti prave povezave na križiščih horizontalnih in vertikalnih povezav na vseh v AND vrata.

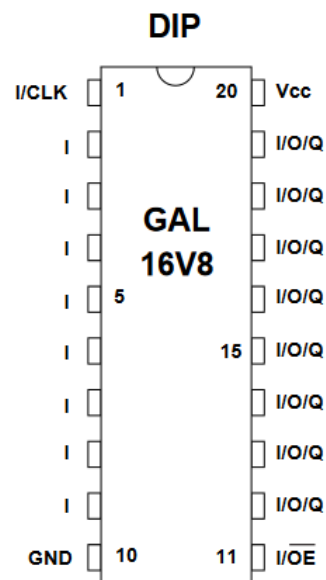


Slika 1: Zgled dela univerzalne strukture. Vertikalne linije so navezane na vhodne priključke tako, kot je to nakazano za I(2) vhod na levi strani slike. Izhodni priključek O (19) je na desni strani slike.

Vsaka od AND vrat imajo po 32 vhodov (prikazanih z le eno horizontalno črto), od tu se vzpostavljajo priključki na vertikalne linije vhodnih signalov. S takim vezjem je mogoče implementirati poljubno logično funkcijo 16 spremenljivk, ki jo sestavlja 7 AND skupkov. Za nameček je rezultat vezja vezan še skoti tri-stanjska vrata...

Industrija je pripravila niz programabilnih logičnih vezij, ki vsebujejo večje število struktur s slike 1. Povezave med križišči vezja vzpostavljamo s programiranjem, že enkrat vzpostavljene povezave pa lahko izbrišemo in vezje programiramo z novimi. Taka vezja tretiramo danes za SPLD, to je »Simple Programmable Logic Device«; obstajajo pa še kompleksnejša CPLD («Complex ...») in najkompleksnejša FPGA («Field Programmable Gate Array«).

Pri tej vaji se bomo ukvarjali z vezjem GAL16V8, ki vsebuje osem struktur s slike 1. Za nameček vsebuje vezje še osem flip-flopov, ki jih pri tej vaji še ne potrebujemo. Kot pove ime tega vezja je na razpolago 16 vhodnih in 8 izhodnih priključkov. Ker ima integrirano vezje le 20 nožic, je nekaterim nožicam dodeljena dvojna vloga: lahko služijo kot izhodni priključki, kot vhodni priključki ali pa kot mešani. V splošnem velja raspored nožic s slike 2.



Slika 2: Priključki vezja GAL16V8, VCC je +5 V

Postopek programiranja GAL vezja je sledeč:

- Zapišemo željene logične funkcije.
- Prepišemo logične funkcije v programsko okolje CUPL in dobimo tekstovno datoteko.
- V programu CUPL poženemo prevajanje, dobimo datoteko po JEDEC standardu, ki popisuje točke povezav na prej omenjenih križiščih.
- Po potrebi v programskem okolju CSIM simuliramo delovanje vpisanih logičnih funkcij.
- V programator vstavimo vezje GAL16V8.
- Poženemo program Xgpro, izberemo ustrezno datoteko JEDEC in sprožimo programiranje, ki je opravljeno v nekaj sekundah.
- Sprogramirano vezje GAL uporabimo na prototipni plošči. Pozor! Napajalna napetost za GAL vezje je +5 V!

## Zgled

- Potrebujemo vezje, ki implementira osnovne logične funkcije negacije, AND, OR in XOR vrat s po dvema vhodoma I1 in I2. Ustrezne logične funkcije so:

negacija	$O1 = \bar{I1}$
AND	$O2 = I1 \cdot I2$
OR	$O3 = I1 + I2$
XOR	$O4 = I1 \oplus I2 = I1 \cdot \bar{I2} + \bar{I1} \cdot I2$

- Odpremo program CUPL in kreiramo nov projekt. V okna obvezno vpišemo ime projekta («Name») in oznako vezja, ki ga želimo programirati («Device»): »g16v8«. V nadaljevanju odgovarjamo na vprašanja: naše vezje potrebuje dva vhodna in štiri izhodne priključke ter nič vozlišč. CUPL spiše za nas predlogo z navedenimi podatki. V to predlogo vpišemo logične enačbe iz točke a) in navedemo nožice, na katerih naj bodo vhodni in izhodni signali. Končna datoteka je na sliki 3.

```

D:\PROGRAM FILES (X86)\WINCUPL\WINCUPL\NAME.PLD
Name Name ;
PartNo 00 ;
Date 26.3.2024 ;
Revision 01 ;
Designer Engineer ;
Company sola ;
Assembly None ;
Location ;
Device g16v8 ;

/* ***** INPUT PINS ***** */
PIN 2 = I1 ; /* */
PIN 3 = I2 ; /* */

/* ***** OUTPUT PINS ***** */
PIN 12 = O1 ; /* */
PIN 13 = O2 ; /* */
PIN 14 = O3 ; /* */
PIN 15 = O4 ; /* */

O1 = !I1;
O2 = I1 & I2;
O3 = I1 # I2;
O4 = I1 $ I2;

```

Slika 3: Tekstovna datoteka z opisom vezja GAL; vhodna priključka sta na nožicah 2 in 3, izhodna pa na 12 do 15.

Simboli za posamezne logične funkcije se razlikujejo od tistih, ki jih uporabljamo pri pouku!

- Poženemo prevajanje. Uspešno prevajanje potrjuje sporočilo »Compilation successful«. Dobili smo datoteko po formatu JEDEC, ta ima podaljšek »jed« in je navadna tekstovna datoteka, ki jo je mogoče pregledovati z urejevalnikom besedila.
- Poženemo simulator, kjer kreiramo novo simulacijo, navedemo vhodne in izhodne signale ter dodamo vektorje vhodnih signalov. Nato poženemo simulacijo in preverimo rezultate.
- V programator vstavimo vezje GAL16V8. Najprej položimo ročko na programatorju v horizontalni položaj, nato vstavimo vezje tako, da je nožica številka 1 diagonalno nasproti ročke za zaklepanje integriranega vezja ter dvignemo ročko za zaklepanje v navpični položaj. Zdaj preberemo generirano JEDEC datoteko v programator in poženemo programiranje.

f) Ročko za zaklepanje postavimo v zgornji položaj in iz programatorja vzamemo vezje GAL ter ga uporabimo v prototipni plošči. Še enkrat: POZOR! Vezje GAL potrebuje napajalno napetost +5 V, vhodni signali smejo biti bodisi 0 V ali +5 V. Vhodni signali izven tega območja uničijo GAL vezje.