**12. [A1,A0]+[B1,B0]=[S1,S0], prijenos se sprema u C0. Zbog čega nije moguće realizirati šest bitno potpuno zbrajalo na GAL16V8?**

Prvo zbrojimo bitove niže težine (A0 i B0) i pogledamo ima li prijenosa ili ne.

|  |  |  |  |
| --- | --- | --- | --- |
| A0 | B0 | S0 | C0 |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Gledamo kako je realiziran S0= (/A0 \* B0) + (A0 \* /B0) => S0 = A0 :+: B0 (XOR)  
C0 = A0 \* B0  
Sad zbrajamo bitove više težine i još njima pribrajamo prijenos C0, koji će ujedno biti i izlaz zbrajanja jer je regisatrski!

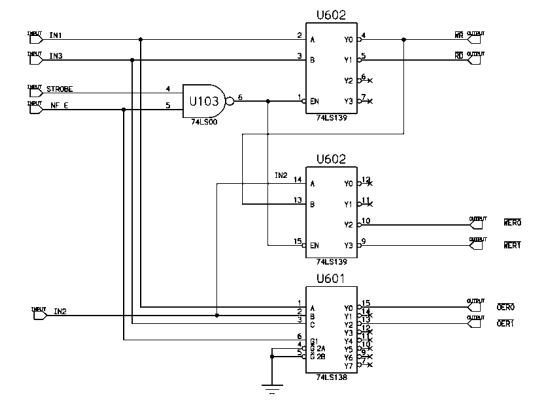
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A1 | B1 | C0 | S1 | C0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

S1=( /A1 \* B1 \* /C0) + (A1 \* /B1 \* /C0 ) + (/A1 \* /B1 \* C0) + (A1 \* B1 \* C1) = (A1 :+: B1 :+: C0)

C0 = /C0 \* ( A1 \* B1) + C0 \* ( A1 + B1 ) ; nisam sigran dal je potreban ovaj izlazni prijenos, ali ga se može napisati za svaki slučaj

Višebitno potpuno zbrajalo nije moguće izvesti jer GAL16v8 jer nema dovoljno ulaza, također sklop u "ili matrici" nema dovoljan broj produkntih članova!

**13. Jednadžba za /WER1 koji omogućava upis u registara R0. Ovo je čudno jer u biti /WER1 omogućava upis u R1, a /WER0 upis u R0, ali zanemarimo to.**

****Napišimo /WER1 = /Y3 jer gledamo di nam je spojen taj izlaz!  
Vidimo da se radi o DEKODERU U602 na ulaz B je spojen signal Y0 prvog dekodera ( WR ), a na ulaz A je spojen signal IN2. Na EN je uvijek spojen na signal NI(STROBE, NF\_E), a on je opet na ulazu negiran.

/EN = / (STROBE \* NF\_E)  
Sljedeći korak je da napišemo zablicu istinitosti za DEKODER:  
(Ak ste zaboravili EN je enable za omogućavanje deokodera, zatim unosimo dva signala, gdje je A bit više težine, a B niže, te ovisno o kojem se binarnom broju radi taj izlaz je aktivan, nas u ovom slučaju zanima Y3.)

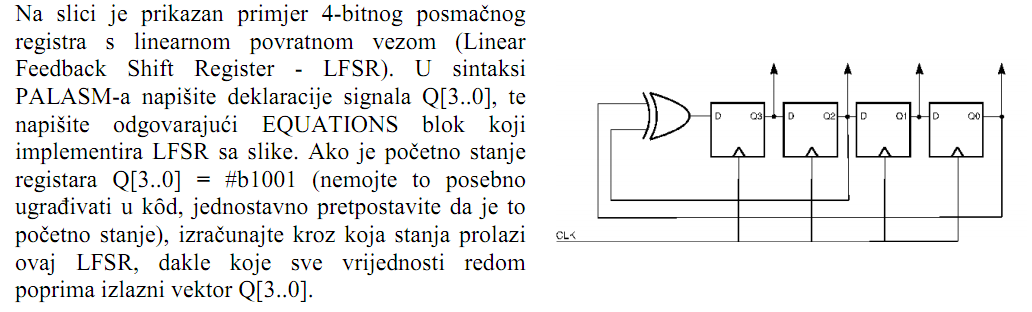
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | /EN | /Y0 | /Y1 | /Y2 | /Y3 |
| X | X | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 |

Instrukcija za Y3 bi izgledala ovako:  
Y3 = A \* B \* /(/EN) ; OVO JE BIL ZAJEB OD POČETKA GLEDA SE DVOSTRUKA OBRNUTA LOGIKA JER JE ULAZ KOMPLEMENTIRAN I U TABLICI PIŠEMO U OBRNUTOJ LOGICI!  
Sad te signale treba raspisati po pravim imenima! Signal A = IN2.

Signal B je malo kompliciraniji, on je zapravo izlaz /Y0 prvog dekodera, ponovno će nam koristiti tablica istinitosti dekodera od malo prije, pa možemo pisati:  
Y0 = /A1 \* /B1 \* //EN   
U ovom slučaju A1 = IN1, a B1 = IN3, pa pišemo:  
Y0 = /IN1 \* /IN3 \* (STROBE \* NF\_E)  
Ali nama zapravo treba komplementirani Y0, ne normalni pa ponovno isčitamo tablicu...  
/Y0 = (STROBE \* NF\_E) \* ( IN1 \* IN3 + IN1 \* /IN3 + /IN1\*IN3) = (STROBE \* NF\_E) \* (IN3 + IN1 \* /IN3)  
Primjetimo da bi bilo isto da umjesto zadnjeg člana napišemo ( IN1 + IN3 \* /IN1 ), ovisi kak odlučimo minimizirati!

Sad ćemo to vratiti u prvu jednadžbu pa dobijemo:  
  
 Y3 = (STROBE \* NF\_E) \* IN2 \* (IN3 + IN1 \* /IN3 ) i sad za /WER1 komplementiramo to:

/WER1 = / ( (STROBE \* NF\_E) \* IN2 \* (IN3 + IN1 \* /IN3 ) )

**14.** 

PIN 1 CLK COMBINATORIAL

PIN 11 /OE

PIN 20 VCC

PIN 10 GND  
PIN 2..5 Q[3..0] REGISTERED

EQUATIONS

Q[0]:= Q[1]

Q[1]:= Q[2]

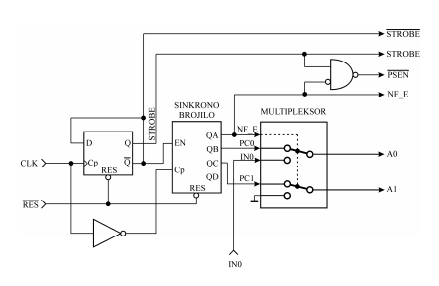
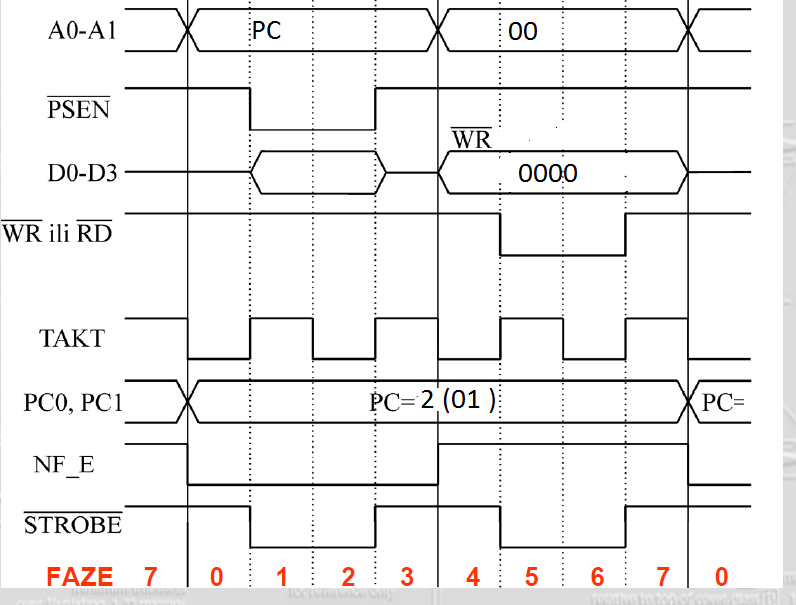
Q[2]:= Q[3]

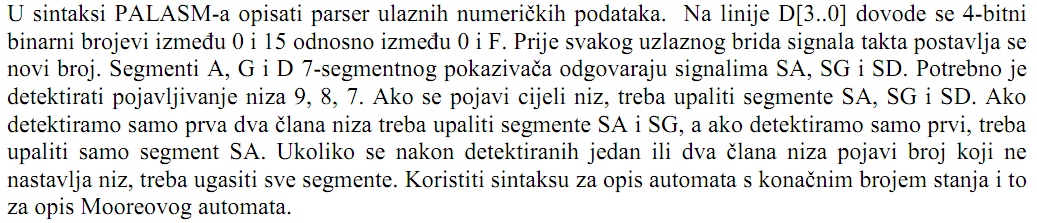
Q[3]:= Q[0] \* /Q[2] + Q[2] \* /Q[0] ; XOR gate

Mislim da ovo nije potrebno dodatno komentirati vidimo da se svi bitovi posmaknu u desno, a na Q[3] ide xor gate.

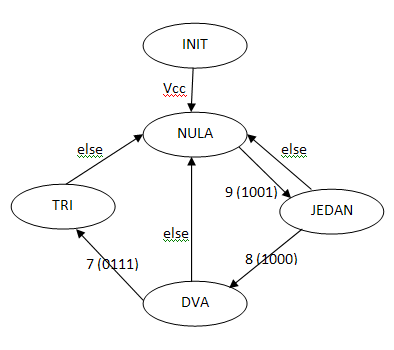
Ako je početno stanje 1001 onda se odvija sljedeće:

Q[3..0]= 1001 -> 1100 -> 1110 -> 1111 -> 0111-> 0011 -> 1001 i opet sve u krug

**15. Vremenski dijagram adresnog generatora odnosa svih ulaznih i izlaznih signala, trenutno stanje je PC = 2, pohranjena je instrukcija : MOV 0, R0 !  
**

**16.** 

D[3..0] ulaz koji porpima vrijednosti od 0 do F.  
Jedini segmenti koji nam trebaju su SA, SG i SD. Znači ujedno ćemo imati stanja kad svijetle po redu ti segmetni, njih ćemo nazvati NULA, JEDAN, DVA i TRI. U zadatku je definirano da trebamo odrediti pojavljivanje niza 9,8,7 ... mislim da se tu radi isključivo o tom redosljedu:

9 -> (A gori ) -> 8 (A i G gore) -> 7 (A, G i D gore), inače se vraćaju na NULU  


/\* deklaracija pinova, mislim da nije potrebno \*/  
PIN 1 CLK COMBINATORIAL

PIN 11 /OE

PIN 20 VCC

PIN 10 GND

PIN 17 SA REGISTERED

PIN 16 SB REGISTERED

PIN 14 SC REGISTERED

PIN 13 SD REGISTERED

PIN 12 SE REGISTERED

PIN 18 SF REGISTERED

PIN 19 SG REGISTERED  
PIN 2..5 D[3..0] COMBINATORIAL

EQUATIONS  
STATE  
MOORE\_MACHINE

; kao što vidimo na slici imamo 4 stanja koja se mijenjaju, ovisno kakav je ulaz tako ulazimo u drugo stanje, +-> znači inače

INIT := VCC -> NULA

; ako dođe devetka pređi u jedan   
NULA := (D[3] \* /D[2] \* /D[1] \* D[0]) -> JEDAN  
 +-> NULA

; ak dođe osmica pređi u dva , ali ak dođe opet devetka počinjemo novi niz i ide opet u jedan  
JEDAN := (D[3] \* /D[2] \* /D[1] \* /D[0]) -> DVA  
 +-> NULA

DVA := (/D[3] \* D[2] \* D[1] \* D[0]) -> TRI  
 +-> NULA  
; ---

TRI := (D[3] \* /D[2] \* /D[1] \* D[0]) -> JEDAN   
 +-> NULA

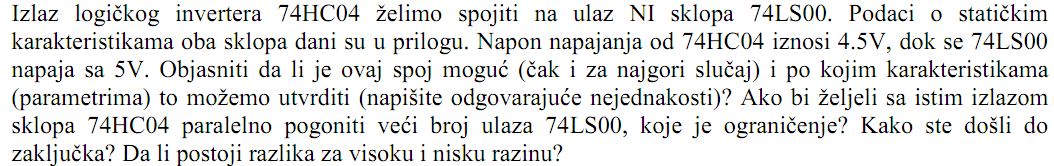
; deklariramo kaj se događa u kojem stanju, kad svijetli jedan pa dva pa sva tri...

INIT= SA\* SB\* SC\* SD\* SE\* SF\* SG

NULA= /SA\* /SB\* /SC\* /SD\* /SE\* /SF\* /SG  
JEDAN= SA\* /SB\* /SC\* /SD\* /SE\* /SF\* /SG

DVA= SA\* /SB\* /SC\* /SD\* /SE\* /SF\* SG

TRI= SA\* /SB\* /SC\* SD\* /SE\* /SF\* SG

**17.** 

Ucc (74HC04) = 4.5 V  
Ucc (74LS00) = 5 V

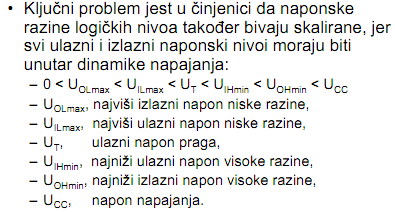
Za visoku razinu : "1" ->

Ugsv = Uohmin (74HC04) - Uihmin (74LS00) = ... tablica ... = 3.98 V - 2 V = 1.98 V

Za nisku razinu : "0" ->

Ugsn = Uilmax (74LS00) - Uolmax (74HCO4) = ... tablica ... = 0.8 V - 0.26 V = 0.54

Ujedno min(Ugsn,Ugsv) = Ugs to je granica šuma!



U našem slučaju: 0 < 0.26 < 0.8 < Ut = 1.5 < 2 < 3.98 < 4.5  
Iz toga možemo zaključiti da će sklop raditi i za najgori slučaj ! Alternativne vrijednosti su bile:   
Uohmin (74HC04) = 4.4 V i Uolmax (74HCO4) = 0.1V

Ioh = -4mA Iih = 20 mikroA

Iol = 4 mA Iil = -0.4 mA

Za maksimalne struje dobivamo maksimalan broj sklopova koje možete dobiti u paraleli:

Ioh / Iih = 200

Za minimalne struje dobivamo minimalan broj sklopova koje mozemo staviti u paralelu:

Iil / Iol = 10

Iz toga sljedi da 10 sklopova sigurno mozemo napajati sa 74HC04.