



Digitalni Elektronski Sistemi

Načrtovanje, sinteza in simulacija

Časovni potek signalov v digitalnih vezjih

1

1

Načrtovanje s programirljivimi vezji

1. Opis vezja in simulacija (Design Entry Utilities)
2. Sinteza logičnega vezja (Synthesize)
3. Prevajanje in tehnološka preslikava
 - določimo lokacije priključkov (User Constraints)
4. Izdelava prog. datotek in nalaganje vezja

Proces za CPLD



Proces za FPGA

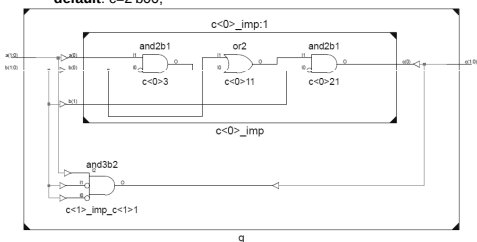


2

2

Kaj dela opisano vezje?

```
always @ (a or b)
case (b)
0: c=a;
1: c=2'b01;
default: c=2'b00;
```

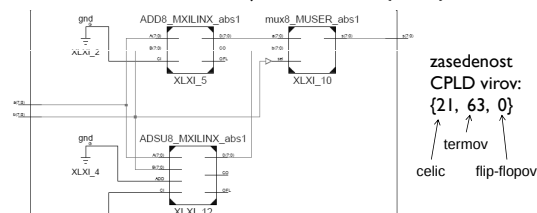


3

3

Izbirni stavek

- ▶ Primer: $c \leq a+b$ when $b > 0$ else $a-b$;
- ▶ Kaj naredi program za sintezo vezja?
 - ▶ operatorji +, - so kombinacijska vezja ADD {14}, ADSU {13}
 - ▶ izbirni stavek when ... else je izbirnik MUX {8 celic}



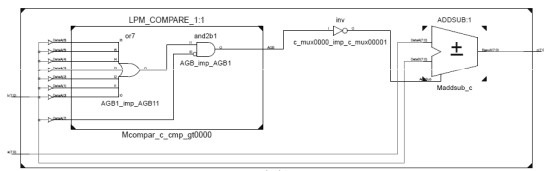
zasedenost
CPLD virov:
{21, 63, 0}
celic termov flip-flopov

4

4

Postopek sinteze vključuje optimizacijo

- ▶ Stavek: $c \leq a+b$ when $b > 0$ else $a-b$;
- ▶ primerjalnik (or7, and2) in ADDSUB blok
- ▶ zasedenost 19 / 55 / 0



- ▶ Drug zapis: $c \leq a+b$ when $b \geq 0$ else $a-b$;
- ▶ zasedenost 17 / 53 / 0

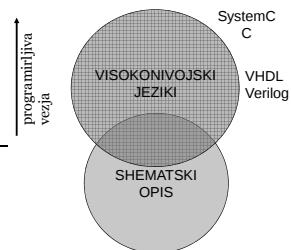
5

5

Način opisovanja digitalnih vezij

Nivoji opisa vezja:

- ▶ specifikacija
- ▶ postopkovni (behavioral)
- ▶ funkcijski (dataflow, RTL)
- ▶ logični
- ▶ nivo transistorjev
- ▶ geometrija vezja (layout)
- ▶ Standardizirani jeziki (IEEE)
 - ▶ VHDL
 - ▶ Verilog, System Verilog
 - ▶ SystemC

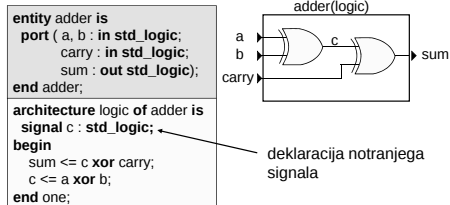


6

6

Funkcijski opis vezja v jeziku VHDL

- stavki opisujejo gradnike vezja
 - stavki za opis vezja se izvajajo paralelno
 - vrstni red stavkov ni pomemben (sočasni stavki)

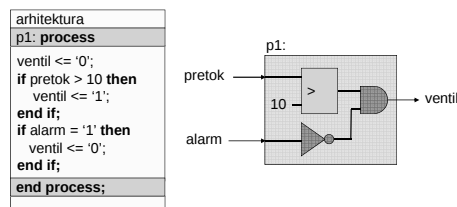


7

7

Postopkovni opis vezja v jeziku VHDL

- v procesu opišemo delovanje vezja
 - zgradbo vezja določi program za sintezo vezij
 - vrstni red stavkov je pomemben (sekvenčni stavki)

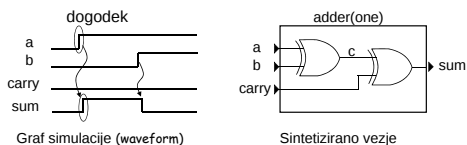


8

8

Obravnava VHDL modelov vezij

- Simulacija modela
 - na simulatorju določimo spreminjanje vhodnih signalov in opazujemo izhode
- Sinteza modela
 - program za sintezo določi zgradbo vezja, ki izhaja iz VHDL modela

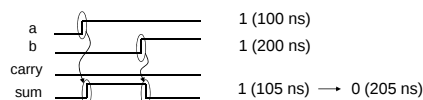


9

9

Simulator diskretnih dogodkov

- Dogodki so opisani z vrednostjo signala in časom ob katerem se zgodijo

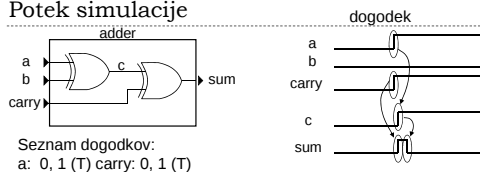


- Ciklus simulatorja
 - simulator izračunava dogodke na signalih in jih uvršča na seznam dogodkov
 - po vseh izračunih se poveča simulacijski čas in
 - izvrši dogodke, ki spremenijo vrednosti signalom

10

10

Potek simulacije



Seznam dogodkov:
a: 0, 1 (T) carry: 0, 1 (T)

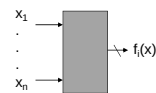
korak	čas	a	b	carry	sum	c
seznam	0	0, 1(T)	0	0, 1(T)	0	0
izvrši	T	1	0	1	0	0
izračunaj	T	1	0	1	1(T+Δ)	1(T+Δ)
izvrši	T+Δ	1	0	1	1	1
izračunaj	T+Δ	1	0	1	0(T+2Δ)	1
izvrši	T+2Δ	1	0	1	0	1

11

11

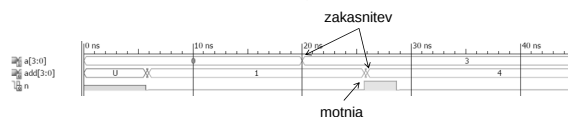
Lastnosti realnih kombinajskih vezij

- Izhod se spremeni z zakasnitvijo ($\Delta > 0$)
- Večina vezij je večnivojskih
 - zaporedno vezana logična vrata
 - v času spremembe so na izhodu motnje!



- Primer: seštevalnik in primerjalnik

add <= a + 1;
n <= '1' when add=0 else '0';

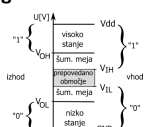


12

12

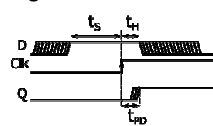
Digitalni sistemi so diskretni

Digitalne vrednosti



- ▶ Kako razlikujemo med logično 1 in 0?
- ▶ Statični red
 - ▶ med 0 in 1 uvedemo nedovoljeno stanje

Digitalni čas



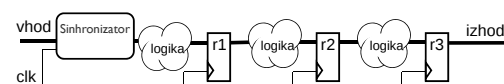
- ▶ Kateri signal se je prej spremenil?
- ▶ Dinamični red
 - ▶ izogibanje "tekmovanju" med signali

▶ 13

13

Pravila dinamičnega reda

- ▶ Ali lahko zagotovimo, da bo vezje vedno delovalo?



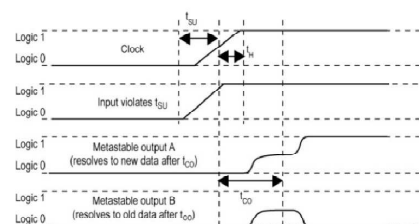
- ▶ S skrbnim načrtovanjem bo vedno izpolnjen dinamični red
 - ▶ ura mora priti do vseh pomnilnih elementov istočasno
 - ▶ sinhrono sekvenčno vezje
- ▶ Na asinhronem vhodu ne moremo upoštevati pravil!
 - ▶ vhodi se ne spreminjajo po zakonitostih naše ure in ne moremo zagotavljati dinamičnega reda (t_S in t_H)

▶ 14

14

Vezje za sinhronizacijo

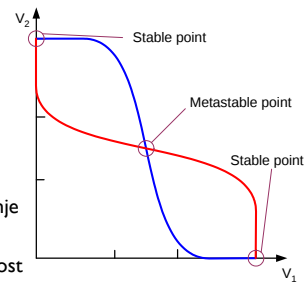
- ▶ Slaba novica: asinhroni arbiter ne obstaja!
- ▶ Poskus rešitve: uporabimo D flip-flop
 - ▶ obstaja možnost, da gre v metastabilno stanje
 - ▶ čez (nedoločen) čas gre izhod v eno ali drugo stabilno stanje



▶ 15

15

Metastabilno stanje



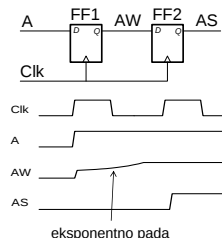
- ▶ lastnost bistabilnih vezij
- ▶ čez čas gre v stabilno stanje
- ▶ nedoločen čas okrevanja
- ▶ verjetnost za metastabilnost eksponentno pada s časom

▶ 16

16

Sinhronizacija z dvema D flip-flopoma

- ▶ metastabilno stanje na izhodu FF1
- ▶ stanje se stabilizira na izhodu FF2, če je na voljo dovolj časa
- ▶ pri višjih frekvencah ure uporabimo več zaporednih D flip-flopov



eksponentno pada
 $P(\text{napake}) = P(\text{metastab stanja}) \times P(\text{ni še stabilno po } t_w)$

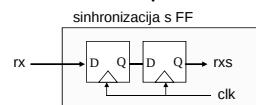
- ▶ Zakasnitev signala je cena za sinhronizacijo, ki se ji ne moremo izogniti!

▶ 17

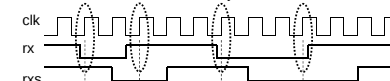
17

Načrtovanje vmesnikov

- ▶ problem komunikacijskih vmesnikov je **sinhronizacija**
 - ▶ asinhrono signale vzorčimo z višjo frekvenco ure in jih peljemo čez sinhronizacijsko vezje
- ▶ težav z metastabilnostjo ne vidimo na simulaciji!



možna metastabilna stanja

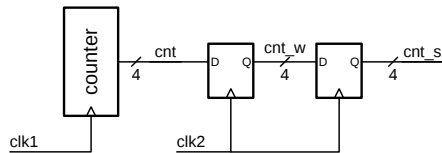


▶

18

Sinhronizacija večbitnih signalov

- ▶ Ali bo prikazano vezje delovalo ?
 - ▶ 4-bitni števec dela z uro clk1, vrednost potrebujemo v vezju, ki preklaplja z uro clk2



- ▶ Vsak bit je posebej sinhroniziran
 - ▶ Kaj se zgodi ob prehodu števca iz 0111 v 1000 ?
 - ▶ Spremenijo se vsi biti, na izhodu je lahko napačen katerikoli...

▶ 19

19

Rešitev 1: Grayev števec

- ▶ Binarni števec zamenjamo z Grayevim, ki naenkrat spreminja le en bit
 - ▶ v najslabšem primeru bomo prebrali prejšnjo vrednost števca in v naslednjem ciklu branja dobili pravo vrednost
 - ▶ **0111** => 1. cikel: **0101**, 2. cikel: **0101** (brez napak)
 - ▶ **0111** => 1. cikel: **0111**, 2. cikel: **0101** (napaka)
- ▶ Takšno vezje se uporablja na naslovne kazalce v pomnilnikih FIFO
 - ▶ kazalec za pisanje šteje z uro clk1
 - ▶ Kazalec za branje šteje z uro clk2
 - ▶ Razlika kazalcev pove koliko je pomnilnik poln

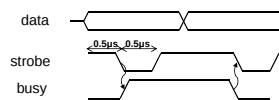
```
# 0000
# 0001
# 0011
# 0010
# 0110
# 0111
# 0101
# 0100
# 1100
# 1101
# 1111
# 1110
# 1010
# 1011
# 1001
# 1000
# 0000
# 0001
# 0011
# 0010
```

▶ 20

20

Rešitev 2: komunikacija z usklajevanjem

- ▶ Usklajevalni protokol (handshaking) določa časovno okno v katerem je mogoč prenos podatka
- ▶ Primer: paralelni vmesnik Centronics
 - ▶ 8-bitni asinhroni prenos podatkov za zunanje naprave
 - ▶ PC z negativnim impulzom (strobe) označi nov podatek
 - ▶ ko naprava zazna strobe, postavi busy in prebere podatek



- ▶ Sinhroniziramo le enobitne kontrolne signale (strobe in busy)

▶ 21

21